SIA SMB SSM API Reference

19.2.1

August 1, 2025

Akamai Technologies, Inc.
150 Broadway
Cambridge, Massachusetts
United States 02142
www.akamai.com

Akamai powers and protects life online. Leading companies worldwide choose Akamai to build, deliver, and secure their digital experiences — helping billions of people live, work, and play every day. Akamai Connected Cloud, a massively distributed edge and cloud platform, puts apps and experiences closer to users and keeps threats farther away. Learn more about Akamai's cloud computing, security, and content delivery solutions at akamai.com and akamai.com/blog, or follow Akamai Technologies on Twitter and Linkedin.

Akamai is headquartered in Cambridge, Massachusetts in the United States with operations in more than 57 offices around the world. Our services and renowned customer care are designed to enable businesses to provide an unparalleled Internet experience for their customers worldwide. Addresses, phone numbers, and contact information for all locations are listed on www.akamai.com/locations.

Contents

19.2.1 API Changes

The following list describes the SSM API changes in release 19.2.1 that are reflected in this document:

19.2.1 SSM API Changes
The SIA Remote service now supports non-expiring deep links for SIA Remote. In releases prior to Release 19.2.1, the SIA Remote service supports time-bound deep links only. The following new SSM APIs are added to support non-expiring deep links: • The new **POST /account/<*accountId*>/spson/mdm_deeplink** API to generate a non-expiring deep link for an SIA Remote service subscriber. • The **PUT /account/<*accountId*>/spson/entitlement** API rotates the entitlement code for time-bound deep links • The **PUT /account/<*accountId*>/spson/mdm_entitlement** API rotates the entitlement code for non-expiring (fixed) deep links. The following SSM APIs are updated to support non-expiring deep links: • The **POST /account/<*accountId*>/spson** API now includes the **"mdm-deeplink-enabled**" field which enables and disables non-expiring deep link support on the subscriber account. Setting the **"mdm-deeplink-enabled**" field to **true** enables non-expiring deep links. Setting the **"mdm-deeplink-enabled**" field to **false** disables non-expiring deep links. • The **PATCH /account/<accountId>/spson** API now includes the **"mdm-deeplink-enabled"** field, which enables and disables non-expiring deep link support on the subscriber account. Setting the **"mdm-deeplink-enabled**" field to **true** enables non-expiring deep links. Setting the **"mdm-deeplink-enabled**" field to **false** disables non-expiring deep links. • The **GET /account/<*accountId*>/spson** API response now contains the "mdm-deeplink-enabled" field, which indicates whether non-expiring deep links are enabled (true) or disabled (false) for a subscriber account. If non-expiring deep links are enabled, the response includes an "mdm-entitlement" field, which displays the entitlement UUID for the non-expiring deep link. **NOTE:** We recommend enabling non-expiring deep links for subscribers in MDM environments so that you do not need to regenerate the deep link and update the MDM configuration every time the link expires. See Chapter 11: SIA Remote Service APIs, for details on the using new and updated APIs for managing non-expiring deep links.

19.2.1 SSM API Changes

The SSM API no longer supports the following API, which retrieves the top queries sent from a subscriber account during a specified time period:

GET /series/dns/top-requests-by-device?view=<*accountId*>&device_id=<*identifier*>&start_inclusive=<*startTime*>&end_exclusive=<*endTime*>

The swagger-api-definition-sb.yaml file replaces the JSON SSM API Swagger file. To see additional details about the endpoints and operations of all SSM APIs in Release 19.2.1, load the swagger-api-definition-sb.yaml file into Swagger.

Chapter 1: Introduction to the SSM RESTful API

This document describes the Representational State Transfer (RESTful) SSM APIs, which are available as part of the SSM. The SSM APIs are remotely accessible service APIs for configuring and managing the SSM and subscribers using one or more of the SSM services that follow:

- Secure Internet Access (SIA) Small and Medium Business (SMB)—The SIA SMB service leverages underlying Domain Name System (DNS) Internet infrastructure so that a carrier can protect subscribers and customize subscribers' internet experiences.
- Subscriber Safety—A toggle which applies malware and botnet protection to subscribers. Setting the toggle to the ON position (in the Subscriber Portal) applies the Subscriber Safety service to all devices in a subscriber's network.
- SIA Remote—Provides DNS-based security to protect data assets and reputations, block security threats, and filter inappropriate content. To see the APIs specific to the SIA Remote service, see the *Common SIA Service Use Cases* manual.

SSM RESTful API Calling Conventions

This section describes the following SSM API call conventions:

Format Differences for nom-http and Curl Clients

How you format SSM APIs depends on whether you are using curl or a nom-http client. The nom-http client interacts with the SSM and allows a user to send API requests in batches which is useful for retrieving configuration information quickly. Using the nom-http client is typically more convenient than performing the same task using curl. The nom-http client is installed in the following location:

/usr/local/nom/sbin

By default, the SSM listens on port **9090** for REST API calls over HTTP. The examples in this document use the nom-http client; however the same information is conveyed regardless of the client you are using.

The examples that follow demonstrate the differences in formatting an API for the curl and nom-http clients. These examples show how to perform the same task (creating a new SSM subscriber account using the **POST /control** API) on each client.

NOTE: In the examples that follow and throughout this document, the SSM API prompt is SSM>; note that your prompt will be the IP address of your SSM host.

Curl Command Example

You enter curl commands on the UNIX shell. The following example shows how to enter the **POST /control** API formatted for curl:

```
# curl -i --header "Content-Type:\ application/json"  --request POST \
--data '{"id":"joe.coffee", "time-zone":"America/Los_Angeles"}' \
http://localhost:9090/control
```

nom-http Command Example

There are two ways to access the nom-http client:

- Using the AMC-X jump host tool (as described in the "Access the SSM API nom-http Client From Your AMC-X Jump Host Tool" section).
- Invoking the following command from the UNIX shell:

```
# /usr/local/nom/sbin/nom-http -s localhost#9090 -u <username> -p <password>
```

For details on using the **nom-http** command, see Appendix D, "Using the nom-http Client."

The example that follows shows how to enter the **POST /control** API formatted for the nom-http client:

```
SSM> POST /control
{"id":"joe.coffee",
"time-zone":"America/Los_Angeles"}
```

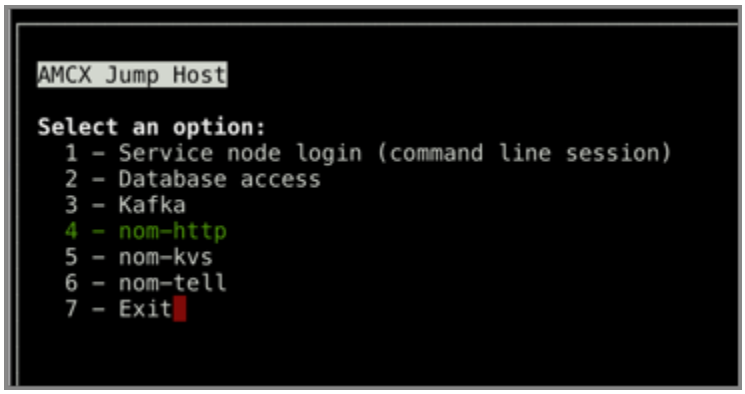
How to Access the SSM API nom-http Client From Your AMC-X Jump Host Tool

Use the following steps to access the SSM API nom-http client from your AMC-X jump host tool:

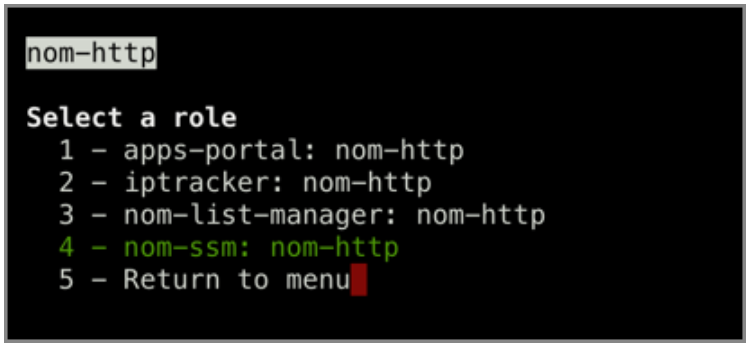
1. To open the AMC-X jump host tool, enter the following command on the node running config-hub:

```
# /usr/local/nom/sbin/amcx-jump-host
```

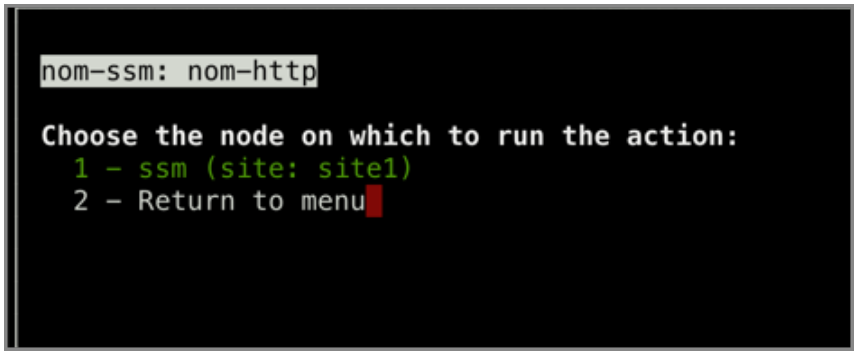
2. In your AMC-X jump host tool, select the **nom-http** option as demonstrated in the following example:



3. Select the **nom-ssm: nom-http** option as demonstrated in the following example:



4. Select the **ssm (site: site1)** option as demonstrated in the following example:



5. Enter your username and password when prompted to access your SSM prompt, where you can start entering APIs. In the following example and throughout this document, the prompt is **SSM>**; note that your prompt will be the IP address of your SSM host:

Supported HTTP Methods

The SSM API uses the following HTTP methods:

- **GET**—Retrieves and displays information about a resource object. The GET method does not perform state change operations on the SSM.
- **POST**—Creates a new resource object or modifies one or more subordinate fields of a resource. You cannot update objects using the POST method.
- **PUT**—Modifies the entire content (all subordinate fields) of an existing resource object. Because this method replaces the entire object, you need to submit a complete representation of that object in the API call; in other words, you need to include all of the object's modifiable fields in the API call. Consider the following rules when entering a **PUT** call:
    - If you omit a modifiable field from the **PUT** request body, that field reverts back to the default setting; if no default setting exists for that field, the API clears any existing configuration from the field.
    - If an unmodifiable field is using the default setting, you do not need to include that field in the **PUT** request body.
- **PATCH**—Modifies only the specified fields for a resource object.
- **DELETE**—Removes a resource object from the system.

Limiting Retrieved Results for GET Requests

A **GET** request can have pervasive effects on your system if the API needs to retrieve extensive amounts of data. To limit the impact on your system's performance, the SSM API supports configurable pagination for all top-level object **GET** requests. You can limit the number of results returned by appending a question mark (**?**), followed by one or both of the following parameters, to the end of any **GET** request string:

- **offset=*number-of objects-omitted-from-response***
- **total=*number-of objects-included-in-response***
- **limit=*maximum-number-of-objects-allowed-in-response***

By default, if you do not append the offset and limit query parameters to a **GET** request, the SSM returns all of the requested objects that exist in the solution. For example, if you enter the **GET /line-list/mobile/nodes** request without appending the **offset** or **limit** parameters, the API returns all line list nodes that exist in the **mobile** line list. The following example shows how to append the **limit** parameter to the **GET /line-list/mobile/nodes** API request so that only the first ten objects (nodes) are returned:

```
GET /line-list/mobile/nodes?limit=10
```

You can append both the **limit** and **offset** parameters to one request, as shown in the example that follows. This example requests a list of policies that are bound to a specified subscriber, with the policies being listed in the order in which the policies were bound to the subscriber. In this example, the limit is set to **10**, which means the API response will include up to ten objects only. The **offset** is set to **2**, which means the first two policies that were bound to the subscriber will be omitted from the API response:

```
GET /line-list/mobile/nodes?limit=10&offset=2
```

If the number of objects in the database is less than the number of objects specified by the limit parameter, the API returns all of the objects in the database. For example, if you set the limit to **15** when the database contains ten objects only, the SSM API returns all ten objects. If you set the limit to **10** and the offset to **2** for that same database (which contains ten objects), the API returns eight objects only.

Understanding Network Convergence

Use the SSM RESTful APIs to configure network convergence on your SSM. Network convergence enables Internet Service Providers (ISPs) to manage multiple mobile and fixed devices in an individual subscriber account. With network convergence, an ISP uses Subscriber Services Manager (SSM) to provision subscribers with fixed and mobile network identifiers (known as lines). After the ISP has provisioned these lines for a subscriber, that subscriber can log into the SIA SMB application to manage those lines.

Using the lines for device identification subscribers can merge two identifiers into a single ID (known as a logical device), simplifying policy enforcement so that you can simultaneously apply one policy to a group of devices, regardless whether those devices use wireless or fixed connections.

Within each subscriber account, logical devices can have different policies applied which dictate the types of websites those devices can access at specific times.

Accessing the SSM API Definitions YAML File for Swagger

For additional details about the endpoints and operations of all SSM APIs in Release 19.2.1, load the swagger-api-definition-sb.yaml file into Swagger.

NOTE: Akamai does not natively support Swagger; however, the curated swagger-api-definition-sb.yaml file is tested to be compatible with third-party tools and integration into an SIA solution.

Chapter 2: Subscriber Account APIs

This chapter describes the API calls that perform the following SSM subscriber account management tasks:

- Create a subscriber account
- Activate a service on a subscriber account
- Configure a subscriber whitelist (allow list)
- Configure a subscriber blacklist (block list)
- Clear the historical data for a specific account
- Update service activation status for a subscriber
- Update the service and time zone configuration for a subscriber account
- Retrieve subscriber account service details
- Retrieve address and timezone details for a subscriber account
- Retrieve the subscriber identifier associated with an IP address
- Retrieve the timestamps at which historical data was last cleared for a subscriber account
- Retrieve the contents of a subscriber blacklist
- Retrieve the contents of a subscriber whitelist
- Deactivate a service for a subscriber
- Delete a subscriber account

Before activating subscriber services and managing SSM subscriber accounts, you need to understand the differences between activating and enabling a service as described in the "Differences Between Activating and Enabling a Service" section that follows.

Differences Between Activating and Enabling a Service

Activating a service provisions the subscriber with the specified service. After a service is activated for a subscriber, an administrator or the subscriber can configure (enable or disable) the enforcement of that service on the subscriber's account.

Consider the following differences between activating and enabling services for a subscriber account:

- Only administrators can activate and deactivate services on subscriber accounts.
- Administrators and subscribers can enable and disable activated services on the subscriber's account.
- You cannot enable or disable a service on an account that does not have that service activated. When you activate a service, the SSM provisions the subscriber with the specified service. The SSM preserves the subscriber's service configuration, even when a subscriber or administrator disables the service on that account. When an administrator deactivates the service, the SSM removes all traces of that service (including the pertinent lists and device group references) from all assigned devices.

- When you activate a service for a subscriber account, the service is enforced (enabled) for the subscriber by default; an administrator and the subscriber can enable and disable an activated service as described in the "Update the Service and Time Zone Configuration for a Subscriber Account" section.
- In the event of an emergency, a subscriber can disable the enforcement of an active service as a troubleshooting step.

Create a Subscriber Account

Use the following **POST** API to create a new subscriber account:

```
SSM> POST /control
{
    "id": "<accountId>",
    "time-zone": "<zone-id>"
}
```

Fields

Field	Description
"id"	Required. Specifies a name for the subscriber account. Replace the ***<accountId>*** argument with a subscriber account name; the name is a unique alphanumeric string that is a maximum length of 36 characters and can include the following special characters: • A dash (-), for example **joe-coffee**. • An underscore (_); for example **joe_coffee**. • A period (.); for example **joe.coffee**. **NOTE:** Ensure the ID you specify is unique in the SSM.
"time-zone"	Optional. Specifies the time zone for the subscriber account. Replace the ***<zone-id>*** argument with a string that represents a time zone belonging to the Arthur David Olson tz database. For a list of possible time zone identifiers, see *https://en.wikipedia.org/wiki/List_of_tz_database_time_zones*. **NOTE:** By default, the time-zone inherits the time-zone setting from the SSM.

Expected Response

The SSM creates the new subscriber account.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows creates a subscriber account called **joe.coffee** and maps that account to the **America/Los_ Angeles** time zone:

```
SSM> POST /control {"id": "joe.coffee","time-zone": "America/Los_Angeles"}
```

Activate a Service on a Subscriber Account

After creating a subscriber account, use the following **POST** APIs to activate services on a subscriber account:

- Use the following **POST** API to activate the SIA SMB (**secure-business**) service on a subscriber account:

```
SSM> POST /control/<accountId>/service
{
    "service":"secure-business",
    "service-profile":"<profileName>",
    "time-zone": "<zoneId>"
}
```

- Use the following **POST** API to activate the Subscriber Safety and SIA Remote services on subscriber accounts that are licensed for those services:

```
SSM> POST /control/<accountId>/service
{
    "service":"<serviceId>",
    "time-zone": "<zoneId>"
}
```

Activating the SIA SMB service on subscriber account provisions the subscriber with the SIA SMB service. After activating the SIA SMB service, an administrator or subscriber can configure (enable or disable) the enforcement of that service on the subscriber account. See the "Differences Between Activating and Enabling a Service" section for more information about activating and enabling services on a subscriber account.

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account for which to activate the specified service.

“service”	Required. Keyword that identifies the service to activate for the specified subscriber account. Replace the ***<serviceId>*** argument with one of the following keywords to activate the desired service: ● **secure-business**—Specifies the SIA SMB service. When activating the SIA SMB service, you need to include the **“service-profile”** construct in the API call. ● **subscriber-safety**—Specifies the Subscriber Safety service. ● **spson**—Specifies the SIA Remote service. **NOTE:** The spson object needs to exist in your system before you enable the **spson** (SIA Remote) service.
“service-profile”	Required for the SIA SMB service. Indicates whether the subscriber account supports a single profile or multiple profiles. Replace the ***<profileName>*** argument with **single-profile** (to specify the single profile SIA SMB service) or **multiple-profile** (to support multiple profiles for an SIA SMB subscriber account). Be aware the **“service-profile”** construct is valid for the SIA SMB (**secure-business**) service only. Do not include this construct if you are activating the Subscriber Safety (**subscriber-safety**) or SIA Remote (**spson**) services. **NOTE:** In the SIA SMB subscriber portal application, profiles are called “device groups.”
“time-zone”	Optional. Specifies the time zone for the subscriber account. Replace the ***zone-id*** argument with a string that represents a time zone belonging to the Arthur David Olson tz database. **Caution!** To avoid confusion and the possibility of conflicting time zone settings, we recommend omitting the optional **"time-zone"** field from this API. Instead, we recommend setting the time zone when you first create the subscriber account with the **POST /control** API as described in the “Create a Subscriber Account” section.

NOTE: If you are supporting the SIA Remote service in your deployment, be aware the spson (SIA Remote) object needs to exist in your system before you enable the **spson** service. If you try to enable the **spson** service before the spson object exists, the API returns the following error:

```
localhost> POST /control/paul.perdevice/service {"service":"spson"}
Error: {
    "error": {
    "status": 404,
    "reason": "SPSON_OBJECT_DOES_NOT_EXIST",
    "message": "spson object does not exist for account"
    }
}
```

Expected Response

The SSM activates the requested service for the subscriber.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Examples

The example that follows enables multiple-profile SIA SMB (**secure-business**) service for a subscriber called **joe.coffee**:

```
SSM> POST /control/joe.coffee/service {"service":"secure-business",
"service-profile":"multiple-profile"}
```

The example that follows enables single-profile SIA SMB (**secure-business**) service for a subscriber called **art.gallery**:

```
SSM> POST /control/art.gallery/service {"service":"secure-business",
"service-profile":"single-profile"}
```

The example that follows enables the **subscriber-safety** service for the subscriber called **art.gallery**:

```
SSM> POST /control/art.gallery/service {"service":"subscriber-safety"}
```

The example that follows enables the SIA Remote (**spson**) service for the subscriber called **joe.coffee**:

```
SSM> POST /control/joe.coffee/service {"service":"spson"}
```

Configure a Subscriber Whitelist (Allow List)

Use the following **POST** API to add nodes (domains) to and remove nodes from a subscriber's global whitelist:

```
SSM> POST /sb/<accountId>/whitelist
{
    "<action>":
        ["<domain>", …],
    …
}
```

NOTE: In the Akamai DNSi/SIA and subscriber portal SIA SMB applications, a subscriber whitelist is called an "allow list."

All SIA SMB subscribers have a global whitelist containing a list of domains (called "nodes") that devices can always access, regardless of any assigned protection level. Whitelists ensure the devices using the SIA SMB service can always access certain websites that might otherwise fall under a restricted content category. The only time whitelist domains are not reachable is during a scheduled internet-off timeframe, during which all internet access is blocked on a subscriber device. See the "Add Internet-off or Content Restriction Schedules to an Existing Profile" section in Chapter 7, "Schedule APIs," for details about internet-off schedules.

Keep the following information in mind when configuring a whitelist:

- Global whitelists take precedence over global blacklists.
- When you add a domain to a whitelist, the SIA SMB service allows all subdomains of the core domain. For example, if you add the domain *example.news.com* to a whitelist, the service always allows access to *news.com* and so forth.
- The sites-limit value in the `business.yaml` file determines the maximum number of domains you can add to a whitelist. See the *SIA 19.2.1 Products: Special Configuration Files* document for details on using the `business.yaml` file.

NOTE: Use the **GET /sb/<*accountId*>/whitelist** command as described in the "Retrieve the Contents of a Subscriber Whitelist" section to retrieve the contents of the global blacklist for a subscriber account.

Fields

Field	Description
<*accountId*>	Required. String that identifies the subscriber account for which to update a whitelist.
<action>	Required. Indicates whether to **add** or **remove** the specified domains to or from a whitelist. Replace the argument with one of the following actions: • **add**—Adds the specified domains to the whitelist. • **remove**—Removes the specified domains from the whitelist. **NOTE:** You need to specify at least one action for the API to be valid. If desired, you can include the **add** and **remove** actions in a single API call.

<*domain*>	Required. Specifies a domain to add to or remove from a whitelist. Replace the **<*domain*>** argument with the desired domain name in the following format:
	example.com
	To add or remove multiple domains, separate each domain name with a comma as demonstrated in the examples that follow:
	• To add multiple domains:
	`{"add": ["domain-1.com", "domain-2.com", "domain-3.com"]`
	• To remove multiple domains:
	`{"remove": ["domain-.com", "domain-2.com", "domain-3.com"]`

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows configuration details for the specified subscriber's whitelist:

```
{
   "adds": [
       {
           "type": "core-domain",
           "node": "<domain>"
       },
       ...
   ],
   "removes": [
       {
           "type": "core-domain",
           "node": "<domain>
       },
```

```
        ...
    ],
    "failures": {
        "adds": [

        ],
        "removes": [

        ]
    },
    "status": "success"
}
```

The table that follows describes the fields displayed in the response body:

Field	Description
"adds"	Displays the following information for each domain that was successfully added to the whitelist: ● "type": Identifies the domain type as a core domain. ● "node": Identifies the domain that was added to the whitelist.
"removes"	Displays the following information for each domain that was successfully removed from the whitelist: ● "type": Identifies the domain type as a core domain. ● "node": Identifies the domain that was removed from the whitelist.
"failures"	Displays the following information for each domain that could not be added to or removed from the whitelist: ● "type": Identifies the domain type as a core domain. ● "node": Identifies the domain that could not be added to or removed from the whitelist.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The examples in this section demonstrate how to perform the following tasks:

- Add a domain to a whitelist
- Remove a domain from a whitelist
- Add and remove multiple domains to and from a whitelist in a single request

Add a Domain to a Whitelist

The example that follows adds a single domain (**domain-1.com**) to a whitelist that belongs to a subscriber account called **joe.coffee**:

```
SSM> POST /sb/joe.coffee/whitelist {"add": ["domain-1.com"]}
{
    "adds": [
        {
            "type": "core-domain",
            "node": "domain-1.com"
        }
    ],
    "failures": {
        "adds": [

        ],
        "removes": [

        ]
    },
    "status": "success"
}
```

Remove a Domain from a Whitelist

The example that follows removes a single domain (**domain-1.com**) from a whitelist that belongs to a subscriber account called **joe.coffee**:

```
SSM> POST /sb/joe.coffee/whitelist {"remove": ["domain-1.com"]}
{
   "removes": [
       {
           "type": "core-domain",
           "node": "domain-1.com"
       }
   ],
   "failures": {
       "adds": [

       ],
       "removes": [

       ]
```

```
    },
    "status": "success"
}
```

Add and Remove Multiple Domains to and from a Whitelist in a Single Request

The example that follows demonstrates how to perform multiple whitelist configuration tasks in a single API call. This example performs the following configuration tasks on a whitelist that belongs to a subscriber account called **joe.coffee**:

- Adds two domains (**domain-1.com** and **domain-2.com**) to the whitelist.
- Removes two domains (**domain-3.com** and **domain-4.com**) from the whitelist.

```
SSM> POST /sb/joe.coffee/whitelist {"add": ["domain-1.com",
"domain-2.com"], "remove": ["domain-3.com", "domain-4.com"]}
{
    "adds": [
        {
            "type": "core-domain",
            "node": "domain-1.com"
        },
        {
            "type": "core-domain",
            "node": "domain-2.com"
        }
    ],
    "removes": [
        {
            "type": "core-domain",
            "node": "domain-3.com"
        },
        {
            "type": "core-domain",
            "node": "domain-4.com"
        }
    ],
    "failures": {
        "adds": [

        ],
        "removes": [

        ]
    },
    "status": "success"
}
```

Configure a Subscriber Blacklist (Block List)

Use the following **POST** API to add nodes to and remove nodes from a global blacklist:

```
SSM> POST /sb/<accountId>/blacklist
{
    "<action>":
        ["<domain>", …],
    …
}
```

NOTE: In the Akamai DNSi/SIA and subscriber portal SIA SMB applications, the subscriber blacklist is called a "block list."

All SIA SMB subscribers have a global blacklist containing a list of domains (called "nodes") that cannot be accessed, regardless of any assigned protection level. Blacklists make it easy to restrict websites that do not belong to any restricted content categories.

Keep the following information in mind when configuring a blacklist:

- Global whitelists take precedence over global blacklists.
- When you add a domain to a blacklist, the SIA SMB service blocks all subdomains of the core domain. For example, if you add the domain *example.news.com* to a block list, the service blocks access to *news.com* and so forth.
- The sites-limit value in the `business.yaml` file determines the maximum number of domains you can add to a blacklist. See the *SIA 19.2.1 Products: Special Configuration Files* document for details on using the `business.yaml` file.

NOTE: Use the **GET /sb/<*accountId*>/blacklist** command as described in the "Retrieve the Contents of a Subscriber Blacklist" section to retrieve the contents of the global blacklist for a subscriber account.

Fields

Field	Description
<*accountId*>	Required. String that identifies the subscriber account for which to update a blacklist.
<action>	Required. Indicates whether to **add** or **remove** the specified domains to or from a blacklist. Replace the argument with one of the following actions: • **add**—Adds the specified domains to the blacklist. • **remove**—Removes the specified domains from the blacklist **NOTE:** You need to specify at least one action for the API to be valid. If desired, you can include the **add** and **remove** actions in a single API call.

Field	Description
<domain>	Required. Specifies a domain or URL to add to or remove from a blacklist. Replace the ***<domain>*** argument with the desired domain name as follows: ***<domain-name>*.com** To add or remove multiple domains, separate each domain name with a comma as demonstrated in the examples that follow: • To add multiple domains to a blacklist: **{"add": ["domain-1.com", "domain-2.com", "domain-3.com"]** • To remove multiple domains from a blacklist: **{"remove": ["domain-4.com", "domain-5.com", "domain-6.com"]**

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows configuration details for the specified subscriber's blacklist:

```
{
   "adds": [
       {
           "type": "core-domain",
           "node": "<domain>"
       },
       ...
   ],
   "removes": [
       {
           "type": "core-domain",
           "node": "<domain>
       },
       ...
   ],
   "failures": {
       "adds": [
      {
           "type": "core-domain",
           "node": "<domain>"
      },

       ],
       "removes": [
       {
           "type": "core-domain",
```

```
            "node": "<domain>"
        },
        ]
    },
    "status": "success"
}
```

The table that follows describes the fields displayed in the response body:

Field	Description
"adds"	Displays the following information for each domain that was successfully added to the blacklist: • "type": Identifies the domain type as a core domain. • "node": Identifies the domain that was added to the blacklist.
"removes"	Displays the following information for each domain that was successfully removed from the blacklist: • "type": Identifies the domain type as a core domain. • "node": Identifies the domain that was removed from the blacklist.
"failures"	Displays the following information for each domain that could not be added to or removed from the blacklist: • "type": Identifies the domain type as a core domain. • "node": Identifies the domain that could not be added to or removed from the blacklist.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The examples in this section demonstrate how to perform the following tasks:

- Add a domain to a blacklist
- Remove a domain from a blacklist
- Add and remove multiple domains to and from a blacklist in a single request

Add a Domain to a Blacklist

The example that follows adds a single domain (**domain-1.com**) to a blacklist that belongs to a subscriber account called **joe.coffee**:

```
SSM> POST /sb/joe.coffee/blacklist {"add": ["domain-1.com"]}
{
    "adds": [
        {
```

```
            "type": "core-domain",
            "node": "domain-1.com"
        }
    ],
    "failures": {
        "adds": [

        ],
        "removes": [

        ]
    },
    "status": "success"
}
```

Remove a Domain from a Blacklist

The example that follows removes a single domain (**domain-1.com**) from a blacklist that belongs to a subscriber account called **joe.coffee**:

```
SSM> POST /sb/joe.coffee/blacklist {"remove": ["domain-1.com"]}
{
   "removes": [
       {
           "type": "core-domain",
           "node": "domain-1.com"
       }
   ],
   "failures": {
       "adds": [

       ],
       "removes": [

       ]
   },
   "status": "success"
}
```

Add and Remove Multiple Domains to and from a Blacklist in a Single Request

The example that follows demonstrates how to perform multiple blacklist configuration tasks in a single API call. This example performs the following configuration tasks on a blacklist that belongs to a subscriber account called **joe.coffee**:

- Adds two domains (**domain-1.com** and **domain-2.com**) to the blacklist.
- Removes two domains (**domain-3.com** and **domain-4.com**) from the blacklist.

```
SSM> POST /sb/joe.coffee/blacklist {"add": ["domain-1.com",
"domain-2.com"], "remove": ["domain-3.com", "domain-4.com"]}
{
    "adds": [
        {
            "type": "core-domain",
            "node": "domain-1.com"
        },
        {
            "type": "core-domain",
            "node": "domain-2.com"
        }
    ],
    "removes": [
        {
            "type": "core-domain",
            "node": "domain-3.com"
        },
        {
            "type": "core-domain",
            "node": "domain-4.com"
        }
    ],
    "failures": {
        "adds": [

        ],
        "removes": [

        ]
    },
    "status": "success"
}
```

Clear the Historical Data for a Specific Account

Use the following **POST** API to clear specific types of historical data for a subscriber account:

```
SSM> POST /account/<accountId>/history:clear
{
    "metrics":[
        "<data-type>"...
    ]
}
```

When you clear historical data for a subscriber account, the subscriber does not see the cleared data in reports and upon device discovery; in addition, SIA SMB subscribers do not see the cleared data in emailed scheduled reports.

NOTE: When clearing historical data, we recommend using the **POST /account/<*accountId*>/history:clear** API for web applications.

Fields

Field	Description
<*accountId*>	Required. String that identifies the subscriber account for which to clear historical data.
"metrics"	Required. Specifies the historical data you want to clear; Replace the **<*data-type*>** argument with one of the following keywords: • **dns** -- Clears Internet activity. • **content** -- Clears content blocks. • **safety** -- Clears safety blocks. • **botnet** -- Clears botnet stops. • **device-discovery** -- Clears device discovery. To clear multiple historical data types, separate each data type with a comma as shown in the following example: `"metrics": ["content", "safety", "botnet"]`

Disable Automatic History Clearance for New Subscriber Accounts

By default, the SSM automatically clears the history for all new subscriber accounts. When a new subscriber activates the SIA SMB or Subscriber Safety service, the SSM clears all prior historical data related to that service from the specified subscriber account. This feature is useful in systems that reuse subscriber IDs.

To disable the automatic history clearance feature in your system, use the following command in the SSM nom-tell Command Channel (CC):

```
nom-ssm> server.update service-activation-clears-history=false
```

To re-enable the automatic history clearance feature, use the following command in the SSM nom-tell CC:

```
nom-ssm> server.update service-activation-clears-history=true
```

Expected Response

A successful response includes HTTP code 200 with a JavaScript Object Notation (JSON) based response body confirming the specified data types are cleared as shown in the example that follows:

```
{
   "successful":[
      {
         "value":"content"
      },
      {
         "value":"safety"
      }
   ],
   "status":"success"
}
```

To display timestamps indicating when historical data was last cleared for a subscriber account, send the following **GET** request to the SSM as described in the "Retrieve the timestamps at which historical data was last cleared for a subscriber account" section:

```
GET /account/<accountId>/history
```

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Examples

The example that follows clears **DNS** (internet activity) data for a subscriber account called **joe.coffee**:

```
SSM> POST /account/joe.coffee/history:clear {"metrics":["dns"]}
{
    "successful": [
        {
            "value": "dns"
        }
    ],
    "status": "success"
}
```

The example that follows shows how to clear multiple historical data types; this example clears data related to **content** blocks, **safety** blocks, and **botnet** stops for a subscriber called **art.gallery**:

```
SSM> POST /account/art.gallery/history:clear
{"metrics":["content", "safety", "botnet"]}
{
   "successful": [
       {
           "value": "content"
       },
       {
           "value": "safety"
       },
       {
           "value": "botnet"
       }
   ],
   "status": "success"
}
```

Update Service Activation Status for a Subscriber

Use the following **PATCH** API to toggle the service activation status for a specific subscriber account:

```
SSM> PATCH /control/<accountId>/service/<serviceName>
{
    "enabled": boolean
}
```

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account for which to activate the specified service.
<serviceName>	Required. String that identifies the service to activate or deactivate. Replace the ***<serviceName>*** argument with one the following keywords: • **secure-business**—Specifies the SIA SMB service) • **subscriber-safety**—Specifies the Subscriber Safety service. • **spson**—Specifies the SIA Remote service.
"enabled"	Required. Boolean value to toggle the activation status of the specified service. Replace the *boolean* argument with **true** to activate the service or **false** to deactivate the service. See the "Differences Between Activating and Enabling a Service" section for details on activating and enabling services for a subscriber account.

Expected Response

The SSM activates or deactivates the specified service for the subscriber as requested.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Examples

The example that follows deactivates the **Subscriber Safety** service for a subscriber called **joe.coffee**:

```
SSM> PATCH /control/joe.coffee/service/subscriber-safety
{"enabled":false}
```

The example that follows disables the SIA Remote (**spson**) service for a subscriber called **art.gallery**:

```
SSM> PATCH /control/art.gallery/service/spson {"enabled":false}
```

To reenable a disabled service for a subscriber, set the **"enabled"** field to **true** as demonstrated in the following example; this example reenables the SIA Remote service for a subscriber called **art.gallery**:

```
SSM> PATCH /control/art.gallery/service/spson {"enabled":true}
```

Update the Service and Time Zone Configuration for a Subscriber Account

Use the following **PATCH** API call to update the timezone and service activation status for a subscriber account:

```
SSM> PATCH /account/<accountId>
  {
    "secure-business": "<boolean>",
    "subscriber-safety": "<boolean>",
    "time-zone": "<zone-id>"
  }
```

Activating a subscriber account provisions the subscriber with the specified service. After a service is activated, an administrator or subscriber can configure (enable or disable) the enforcement of that service on the subscriber's account. See the "Differences Between Activating and Enabling a Service" section for more information about activating and enabling services on a subscriber account.

Fields

Field	Description
<*accountId*>	Required. String that identifies the subscriber account you want to retrieve.
"secure-business"	Optional. Indicates whether the SIA SMB service evaluates traffic for the subscriber account. Replace the <*boolean*> argument with **true** to enable the SIA SMB service and **false** to disable the service.
"subscriber-safety"	Optional. Indicates whether the Subscriber Safety service evaluates traffic for the subscriber account. Replace the <*boolean*> argument with **true** to enable the Subscriber Safety service and **false** to disable the service.
"time-zone"	Optional. Updates the time zone associated with the subscriber account. Replace the <*zone-id*> argument with a string that represents a time zone belonging to the Arthur David Olson tz database. For a list of possible time zone identifiers, see the following URL: *https://en.wikipedia.org/wiki/List_of_tz_database_time_zones*. **NOTE:** By default, the time-zone inherits the time-zone setting from the SSM.

Expected Response

The SSM updates the subscriber account as requested.

Consider the following information when using this **PATCH** API to update a subscriber account:

- If you try to enable a service that is not activated for a subscriber account, the API returns a 403 error indicating the service is not activated.
- If any of the fields you specify contain errors, the SSM does not make any of the requested updates, regardless of whether other fields do not have errors.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Examples

The example that follows disables the SIA SMB service settings for a subscriber account called **joe.coffee**:

```
SSM> PATCH /account/joe.coffee  {"secure-business":"false"}
{
    "successful": [
        {
            "value": false
            "field": "secure-business",
        }
    ],
    "status": "success"
}
```

The example that follows enables the SIA SMB service settings for a subscriber account called **art.gallery** and sets the time zone to **America/Los_Angeles**:

```
SSM> PATCH /account/art.gallery {"secure-business":"true",
"time-zone":"America/Los_Angeles"}
{
    "successful": [
        {
            "field": "time-zone",
            "value": "America/Los_Angeles"
        },
        {
            "field": "secure-business",
            "value": true
        }
```

```
    ],
    "status": "success"
}
```

The example that follows shows what happens when you try to enable a service that is not activated for the specified subscriber account. In this example, the administrator tries to enable the SIA SMB service and update the time zone for the subscriber called **joe.coffee**; the API returns an error and does not make the requested updates because the SIA SMB service is not active for the subscriber:

```
SSM> PATCH /account/joe.coffee {"secure-business":"true",
"time-zone":"America/Los_Angeles"}
Error: {
    "error": {
        "status": 403,
        "reason": "ACCOUNT_SERVICE_NOT_ACTIVATED",
        "message": "service not activated: secure-business"
    }
}
```

Retrieve Subscriber Account Service Details

Use the following **GET** API to retrieve details about a specific subscriber account:

```
SSM> GET /account/<accountId>
```

Fields

Field	Description
<accountId>	Required. Identifies the subscriber account for which to retrieve configuration information.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows information about each service that has been provisioned:

```
{
    "id": "<accountId>",
    "time-zone": "<zoneId>",
    "subscriber-safety": {
        "enabled": <boolean>,
```

```
        "activated": <boolean>,
        "time": "<year-month-day>T<time>",
    },
    "secure-business": {
        "enabled": <boolean>,
        "activated": <boolean>,
        "time": "<year-month-day>T<time>",
        "type": "<multiple-or-single-profile>",
        "bypass-pin-enabled": <boolean>,
        "profile-count": <integer>,
        "profile-limit": <integer>,
        "spson-enabled": <boolean>,
        "bypass-private-dns": {
            "apple-private-relay": <boolean>
        }
    },
    "https-termination": {
        "enabled": <boolean>,
        "activated": <boolean>
    }
}
```

The table that follows describes the possible fields in the API response:

Field	Description
"Id"	String that identifies the subscriber account for which service information is displayed.
"time-zone"	String that identifies the time zone for the subscriber account (where the string represents a time zone belonging to the Arthur David Olson tz database). For a list of possible time zone identifiers, see the following URL: *https://en.wikipedia.org/wiki/List_of_tz_database_time_zones*.
Subscriber Safety	
"subscriber-safety"	String indicating the fields that follow pertain to the Subscriber Safety service.
"enabled"	Boolean value indicating whether the SSM evaluates subscriber traffic for the Subscriber Safety service. A value of **true** indicates the Subscriber Safety service is enabled (enforced), while a value of **false** indicates the service is disabled (not enforced). In the event of an emergency, a subscriber can disable Subscriber Safety service enforcement as a troubleshooting step. See the "Differences Between Activating and Enabling a Service" section for details on activating and enabling services for a subscriber account.

Field	Description
"activated"	Boolean value indicating whether a subscriber is entitled to use the Subscriber Safety service. A value of **true** indicates the Subscriber Safety service is activated for the subscriber account; a value of **false** indicates the service is deactivated. When the Subscriber Safety service is activated for a subscriber account, the subscriber can enable and disable that service as desired. The "activated" field is useful for development teams who want to drive the user experience.
"time"	String indicating the time the Subscriber Safety service was activated. The time is expressed in the following format, with the ***.<microseconds>*** and ***+<timezoneId>*** arguments being optional: ***<year>-<month>-<day>*** **T*<hours>*:<minutes>:*<seconds>*** [***.<microseconds>***][***+<timezoneId>***] In the following example, the “time” value indicates the service was activated on June 4, 2024 at 5:51 PM and 19 seconds and 212 microseconds: `2024-06-04T17:51:19.212` **NOTE:** This field is valid only for services in the **activated** state.
SIA SMB	
"secure-business"	Indicates the fields that follow pertain to the SIA SMB service.
"enabled"	Boolean value indicating whether the SSM evaluates subscriber traffic for the SIA SMB service. A value of **true** indicates the SIA SMB service is enabled (enforced), while a **false** value indicates the service is disabled (not enforced). In the event of an emergency, a subscriber can disable SIA SMB service enforcement as a troubleshooting step.
"activated"	Boolean value indicating whether a subscriber is entitled to use the SIA SMB service. A value of **true** indicates the SIA SMB service is activated for the subscriber account; a value of **false** indicates the service is deactivated. When the SIA SMB service is activated for a subscriber account, the subscriber can enable and disable that service as desired. This field is useful for development teams who want to drive the user experience.
"time"	String indicating the time the SIA SMB service was activated. The time is expressed in the following format: *year-month-day***T***hours* **:** *minutes* **:** *seconds*. This field is valid only for services in the **activated** state.

Field	Description
"type"	String that indicates whether the SIA SMB subscriber account has single or multiple profile service enabled; a **single-profile** value indicates the service supports a single device group; a **multi-profile** value indicates the service can support multiple device groups.
bypass-pin-enabled	This field is not supported for SIA SMB and will be removed in a future release.
"profile-count"	Integer indicating the number of profiles (device groups) the SIA SMB service supports. **NOTE:** In the SIA SMB subscriber portal application, profiles are called “device groups.”
"profile-limit"	Integer indicating the maximum number of profiles (device groups) The SIA SMB service allows. **NOTE:** In the SIA SMB subscriber portal application, profiles are called “device groups.”
"spson-enabled"	Boolean value indicating whether the subscriber has the SIA Remote service is enabled (**true**) or disabled (**false**) with their SIA SMB service.
"bypass-private-dns": {"apple-private-relay": <*boolean*>}	Boolean value indicating whether iCloud Private Relay blocking is enabled (**true**) or disabled (**false**) for the SIA SMB service.
https-termination	
"enabled"	Boolean value indicating whether the Proxy server evaluates subscriber traffic for the secure socket layer (SSL) termination service (which is also known as HTTPS termination service). A value of **true** indicates HTTPS termination is enabled (enforced), while a value of **false** indicates HTTPS termination is disabled (not enforced). In the event of an emergency, a subscriber can disable HTTPS termination enforcement as a troubleshooting step. **Caution!** SSL (HTTPS) termination service evaluation is a limited feature that is not commonly configured; set the "https-termination" field to **false** unless Akamai has granted your company explicit contracted permissions to enable that feature.
"activated"	Boolean value indicating whether a subscriber is entitled to use the HTTPS termination service. A value of **true** indicates the HTTPS termination service is activated for the subscriber account; a value of **false** indicates the HTTPS termination service is deactivated. When a service is activated for a subscriber account, the subscriber can enable and disable that service as desired. This field is useful for development teams who want to drive the user experience.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows retrieves configuration details for a subscriber account called **joe.coffee**:

```
SSM> GET /account/joe.coffee
{
    "id": "joe.coffee",
    "time-zone": "America/New_York",
    "subscriber-safety": {
        "enabled": true,
        "activated": true,
        "time": "2024-04-27T00:27:15.704+00:00",
    },
    "secure-business": {
        "enabled": true,
        "activated": true,
        "time": "2024-04-27T00:27:15.187+00:00",
        "type": "multiple-profile",
        "bypass-pin-enabled": false,
        "profile-count": 3,
        "profile-limit": 8,
        "spson-enabled": false,
        "bypass-private-dns": {
            "apple-private-relay": false
        }
    },
    "https-termination": {
        "enabled": false,
        "activated": false
    }
}
```

Retrieve Address and Timezone Details for a Subscriber Account

Use the following **GET** API to retrieve time zone and address information for a specific subscriber account:

```
SSM> GET /control/<accountId>
```

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account you want to retrieve.

NOTE: The IP Tracker needs to be configured to report the IP addresses to Levant. If this call does not return IP addresses in the response body, check the IP Tracker configuration to ensure IP Tracker is configured properly.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows information about each service that has been provisioned:

```
{
    "id":"<accountId>",
    "timeZone":"<zoneId>",
    "addresses":["<ipAddress…>"]
}
```

The table that follows describes the fields in the API response:

Output Field	Description
"id"	String that identifies the subscriber account for which information is displayed.
“TimeZone”	String that identifies the timezone for the subscriber account, where the zone-id is a string representing a time zone from the Arthur David Olson tz database
"addresses"	Specifies a list of addresses associated with the subscriber. Each "ipAddress" argument is formatted as a canonical IPv4 or IPv6 address or in CIDR notation. If the "addresses" field is empty, all address bindings are removed for the specified subscriber.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows retrieves time zone and address information for a subscriber account called **joe.coffee**:

```
SSM> GET /control/joe.coffee
{
    "id":"joe.coffee",
    "timeZone":"America/Cancun",
    "addresses":["192.0.2.10/8"]
}
```

Retrieve the Subscriber Identifier Associated with an IP Address

Use the following **GET** API to retrieve the name of the subscriber account associated with a specific IP address:

```
SSM> GET /check/address/<ipAddress>
```

Fields

Field	Description
<ipAddress>	Required. Specifies the IP address for which to display the associated subscriber ID. Replace the ***<ipAddress>*** argument with an IP address in the format *AAA.BBB.CCC.DDD*.

Expected Response

The API returns the subscriber ID associated with the specified IP address as demonstrated in the following example:

```
{
    "address": "<ipAddress>",
    "subscriber-id": "<accountId>"
}
```

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows retrieves the subscriber ID that is associated with the IP address **192.0.2.10**:

```
SSM> GET /check/address/192.0.2.10
{
    "address": "192.0.2.10",
    "subscriber-id": "art.gallery"
}
```

Retrieve the Timestamps at which Historical Data was Last Cleared for a Subscriber Account

Use the following **GET** API to retrieve the UNIX-formatted timestamps at which historical data was last cleared for a specific subscriber account:

```
SSM> GET /account/<accountId>/history
```

Fields

Field	Description
<accountId>	Required. Identifies the subscriber account for which to retrieve historical data.

Expected Response

A successful response includes an HTTP code 200 with a JSON based response body that displays the number of seconds that have passed since the associated data type was last cleared (in other words, the <*seconds*> variable uses the UNIX timestamp format):

```
{
   "metrics": {
       "device-discovery": <seconds>,
       "dns": <seconds>,
       "safety": <seconds>,
       "botnet": <seconds>,
       "content": <seconds>,
   }
}
```

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows retrieves the times (as UNIX-formatted timestamps) at which different historical data types were last cleared for a subscriber called **joe.coffee**:

```
SSM> GET /account/joe.coffee/history
{
    "metrics": {
        "device-discovery": 1714177635,
        "dns": 1714177416,
        "safety": 1715152453,
        "botnet": 1714177635,
        "content": 1714177113
    }
}
```

Retrieve the Contents of a Subscriber Blacklist

Use the following **GET** API to retrieve a list of the nodes (domains) in a subscriber's personal blacklist:

```
SSM> GET /sb/<accountId>/blacklist
```

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account for which to retrieve blacklist details.

NOTE: Use the **POST /sb/*<accountId>*/blacklist** API as described in the "Configure a Subscriber Blacklist (Block List)" section to add domains (nodes) to a blacklist.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows configuration details for the specified subscriber's blacklist:

```
{
    "total": <integer>,
    "limit": <integer>,
    "content":[
        "<node>",
        "<node>",
        ...
    ]
}
```

The table that follows describes the fields displayed in the response body:

Field	Description
"total"	Total number of domains included in the list.
"limit"	Integer indicating the maximum number of domains that can be displayed in the response body.
"content"	New-line-delimited list of domains (nodes) in the subscriber's blacklist. Each node is formatted as a text string.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Examples

The example that follows shows sample blacklist content for an SIA SMB subscriber called **joe.coffee**. In this example, the subscriber has five nodes in their blacklist:

```
SSM> GET /sb/joe.coffee/blacklist
{
    "total": 5,
    "limit": 10,
    "content": [
        "www.blocked-url-example.com",
        "blocked-domain-example.com",
        "www.blocked-url-example.org",
        "www.blocked-url-2-example.com",
        "blocked-domain-example.edu"
    ]
}
```

Retrieve the Contents of a Subscriber Whitelist

Use the following **GET** API to retrieve a list of the nodes (domains) in a subscriber's personal whitelist (known as an allow list in the SIA SMB application):

```
SSM> GET /sb/<accountId>/whitelist
```

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account for which to retrieve **whitelist** (allow list) details.

NOTE: Use the **POST /sb/*<accountId>*/whitelist** API as described in the "Configure a Subscriber Whitelist (Allow List)" section to add domains (nodes) to a whitelist.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows configuration details for the specified subscriber's whitelist (allow list):

```
{
    "total": <integer>,
    "limit": <integer>,
    "content": [
        "<node>",
        "<node>",
        ...
    ]
}
```

The table that follows describes the fields displayed in the response body:

Field	Description
"total"	Integer indicating the number of nodes (domains) in the whitelist.
"limit"	Integer indicating the maximum number of domains that can be displayed in the response body.
"content"	New-line-delimited list of domains (nodes) in the subscriber's whitelist. Each node is formatted as a text string.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows shows sample allow list (**whitelist**) content for an SIA SMB subscriber called **joe.coffee**. In this example, the subscriber has three nodes in an allow list:

```
SSM> GET /sb/joe.coffee/whitelist
{
    "total": 3,
    "limit": 10,
    "content": [
        "allowed-domain1-example.net",
        "allowed-domain2-example.com",
        "allowed-domain3-example.edu"
    ]
}
```

Deactivate a Service for a Subscriber

Use the following **DELETE** API to deactivate service support for a specific subscriber account:

```
SSM> DELETE /control/<accountId>/service/<service-name>
```

Fields

Field	Description
<accountId>	Required. Identifies the subscriber account for which to deactivate the specified service.
<service-name>	Required. Identifies the service to deactivate for the specified subscriber account. Possible services are: • **subscriber-safety** • **secure-business** • **spson**

When you deactivate a service for a subscriber account, the service is removed from a subscriber along with all traces of that service's association with the subscriber's account. Deactivation removes the pertinent personal lists and device group references from all assigned devices. See the "Differences Between Activating and Enabling a Service" section for details on activating and enabling services for a subscriber account.

NOTE: To reactivate a disabled service, use the **POST /control/*<accountId>*/service** API as described in the "Activate a Service on a Subscriber Account" section.

Expected Response

The SSM deactivates the specified service for the subscriber as requested.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows deactivates the SIA SMB service for the subscriber called **joe.coffee**:

```
SSM> DELETE /control/joe.coffee/service/secure-business
```

Delete a Subscriber Account

Use the following **DELETE** API to remove a subscriber account from the SSM:

```
SSM> DELETE /account/<accountId>
```

Fields

Field	Description
<accountId>	Required. Identifies the subscriber account you want to delete.

Expected Response

The SSM removes the specified subscriber account from the system.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows deletes a subscriber account called **joe.coffee**:

```
SSM> DELETE /account/joe.coffee
```

Chapter 3: Subscriber Line APIs

The SSM API uses "lines" to identify network activity that is associated with a subscriber. The following identifiers are examples of lines:

- The DHCP client-id.
- The MSSISDN (a device phone number).
- Any other identifier that occurs in network traffic and can be used to identify a subscriber.

A line can be used to identify network traffic (through RADIUS or DHCP) and relate the traffic to a subscriber in the network. Typically, you use the SSM subscriber line APIs to provision lines associated with specific subscribers so that network traffic can be associated with the appropriate subscriber.

For converged networks, you use the SSM subscriber line APIs to provision the lines associated with a phone and associate those lines with device identifiers; this allows subscribers to merge the two identifiers into a single ID (known as a logical device). Using a single logical device ID simplifies policy enforcement so that you can simultaneously apply one policy to a group of devices, regardless whether those devices use wireless or fixed connections. For details about subscriber lines, see the *Network Convergence Configuration Manual*. This chapter describes the SSM API calls you can use to manage SSM subscriber lines and line lists. Use the line APIs to perform the following tasks:

- Create a line list
- Add and remove lines to and from a line list
- Update the contents of a specific line list
- Retrieve the contents of all line lists associated with a subscriber account
- Retrieve all line lists currently in your SSM
- Retrieve the “has-devices” value for a specific line list
- Retrieve general information about a line
- Display general information about all lines in a line list
- Remove a line list from the SSM

Create a Line List

Use the following **POST** API to add an existing line list to your SSM:

```
SSM> POST /line-list
{
  "has-devices": <boolean>,
  "name":"<listId>"
}
```

After adding a line list to your SSM, add subscriber lines to that line list as described in the "Add and Remove Lines to and from a Line List" section.

Fields

Field	Description
"has-devices"	Required. Boolean value indicating whether the specified list contains line nodes that need to be treated as devices. A value of **true** indicates that the line node for this list will be treated as devices. A value of **false** indicates that the line node for this list will not be treated as devices. **NOTE:** For fixed line lists, the **"has-devices"** field is always **false**. Only mobile networks can have lines that are devices.
"name"	Required. String that identifies a line list that exists in your Levant subscriber mapping configuration; replace the <*listId*> argument with the name of the line list.

Expected Response

The SSM adds the line list to your SSM as requested and displays a "201 Created" response.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows adds a line list called **mobile-network** to the SSM:

```
SSM> POST /line-list {"has-devices":true, "name":"mobile-network"}
```

Add and Remove Lines to and from a Line List

Use the following **POST** API to add subscriber lines to and remove subscriber lines from an existing line list:

```
SSM> POST /account/<accountId>/line
{
    "add":[
        {
            "name":"<lineId>",
            "list":"<listId>"
        },
            …
    ]
    "remove":[
        {
            "name":"<lineId>",
            "list":"<listId>"
        },
        …
    ]
}
```

NOTE: A line represents a device requiring access to the SIA SMB service. Each line has a line ID, which is the network identifier for the associated device.

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account for which to add or remove subscriber lines.
"name"	Required. String that identifies the subscriber line you want to add to or remove from the specified line list. For a mobile line, use the MSISDN associated with the subscriber device. For a fixed line, use a name which clearly identifies the subscriber, such as the login ID for the subscriber.
"list"	Required. String that identifies the line list for which you are adding or deleting subscriber lines. The specified list needs to exist before you can add lines.

Expected Response

The SSM creates the new line list as requested and returns a response body which includes the configuration of the lines you added to or removed from SSM:

```
{
    "adds": [
        {
            "name": "<lineId>",
            "list": "<listId>",
            "subscriber": "<accountId>",
            "is-device": <boolean>
        },
        ...
    ],
    "removes": [
        {
            "name": "<lineId>",
            "list": "<listId>",
            "subscriber": "<accountId>",
            "is-device": <boolean>
        },
        ...
    ],
    "failures": {
        {
            "name": "<lineId>",
            "list": "<listId>",
            "subscriber": "<accountId>",
            "is-device": <boolean>
        },
        ...
    },
    "status": "<status>"
}
```

The table that follows describes the API response body fields:

Field	Description
"adds"	Provides the following information about the lines that are successfully added to the specified line list: • "name"— Network identifier associated with the line that is successfully added to the line list. • "list"—Identifies the line list to which the specified line is added. • "subscriber"—Identifies the subscriber account to which the updates apply. • "is-device"—Boolean value indicating whether the specified line is a device. A value of true indicates the line node is treated as a device. A value of false indicates the line node is not treated as a device. **NOTE:** For fixed line lists, the "is-device" field is always false. Only mobile networks can have lines that are devices. **NOTE:** The response does not include the “adds” fields the API does not include an **“add”** request.
"removes"	Provides the following information about the lines that are successfully removed from the specified line list: • "name"— Network identifier associated with the line that is successfully removed from the line list. • "list"—Identifies the line list from which the specified line is removed. • "subscriber"—Identifies the subscriber account to which the update applies. • "is-device"—Boolean value that indicates whether the specified line node needs to be treated as a device. A value of true indicates that the line node is a device. A value of false indicates that the line node is not a device. **NOTE:** For fixed line lists, the "is-device" field is always false. Only mobile networks can have lines that are devices. **NOTE:** The response does not include the “removes” fields the API does not include a **“remove”** request.
"failures"	Provides the following information about operation attempts that failed: • “adds” or “removes”—Indicates whether the failure pertains to an **“add”** or **"remove"** request. • "name"— Network identifier associated with the line that failed to be added to or removed from the line list. • "list"—Identifies the line list for which the failed operation attempt occurred. • "subscriber"—Identifies the subscriber account to which the failed updates apply.
"status"	Indicates whether the entire operation succeeded, failed, or was partially successful.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Examples

This section provides examples that demonstrate how to perform the following tasks:

- Add a line to the fixed-network line list
- Add a line to the mobile-network line list
- Add lines to the fixed-network and mobile-network line lists in a single API request
- Remove a line from a line list
- Add lines to and remove lines from line lists in a single API request

Example: Add a Line to the fixed-network Line List

The example that follows adds a subscriber line (with the line identifier **11223445514)** to the **fixed-network** line list that belongs to a subscriber account called **joe.coffee**:

```
SSM> POST /account/joe.coffee/line {"add":[{"name":"11223445514",
"list":"fixed-network"}]}
{
    "adds": [
        {
            "name": "11223445514",
            "list": "fixed-network",
            "subscriber": "joe.coffee",
            "is-device": false
        }
    ],
    "failures": {

    },
    "status": "success"
}
```

Example: Add a Line to the mobile-network Line List

The example that follows adds a subscriber line (with the line identifier **11255544193**) to the **mobile-network** line list that belongs to a subscriber account called **art.gallery**:

```
SSM> POST /account/art.gallery/line {"add":[{"name":"11255544193",
"list": "mobile-network"}]}
{
    "adds": [
        {
            "name": "11255544193",
            "list": "mobile-network",
            "subscriber": "art.gallery",
            "is-device": true
        }
```

```
    ],
    "failures": {

    },
    "status": "success"
}
10.42.83.196> POST /line-list {"has-devices":false,
"name":"fixed-network"}

10.42.83.196> POST /account/joe.coffee/line
{"add":[{"name":"11223445514", "list":"fixed-network"}]}
{
   "adds": [
       {
           "name": "11223445514",
           "list": "fixed-network",
           "subscriber": "joe.coffee",
           "is-device": false
       }
   ],
   "failures": {

   },
   "status": "success"
}
```

Example: Add Lines to the fixed-network and mobile-network Line Lists in a Single API Request

The example that follows performs the following tasks for a subscriber account called **joe.coffee**:

- Adds the subscriber line **12345498914** to the **fixed-network** line list.
- Adds the subscriber line **16504914193** to the **mobile-network** line list.

```
SSM> POST /account/joe.coffee/line {"add":[{"name":"12345498914",
"list":"fixed-network"},{"name":"16504914193", "list":
"mobile-network"}]}
{
    "adds": [
        {
            "name": "12345498914",
            "list": "fixed-network",
            "subscriber": "joe.coffee",
            "is-device": false
        },
        {
```

```
            "name": "16504914193",
            "list": "mobile-network",
            "subscriber": "joe.coffee",
            "is-device": true
        }
    ],
    "failures": {

    },
    "status": "success"
}
```

Example: Remove a Line from a Line List

The example that follows removes a subscriber line (with the line identifier **14414470187**) from the **mobile-network** line list for a subscriber account called **joe.coffee**:

```
SSM> POST /account/joe.coffee/line {"remove":[{"name":"14414470187",
"list": "mobile-network"}]}
{
    "removes": [
        {
            "name": "14414470187",
            "list": "mobile-network",
            "subscriber": "joe.coffee",
            "is-device": true
        },
    ],
    "failures": {

    },
    "status": "success"
}
```

Example: Add Lines to and Remove Lines from Line Lists in a Single API Request

The example that follows demonstrates how to add and remove subscriber lines in a single API request. This example performs the following line changes in line lists belonging to a subscriber account called **joe.coffee**:

- Adds a new line (called **12345498920**) to the **fixed-network** line list.
- Removes an existing line (called **16504914193**) from the **mobile-network** line list.

```
SSM> POST /account/joe.coffee/line {"add":[{"name":"12345498920",
"list":"fixed-network"}], "remove":[{"name":"16504914193", "list":
"mobile-network"}]}
```

```
{
    "adds": [
        {
            "name": "12345498920",
            "list": "fixed-network",
            "subscriber": "joe.coffee",
            "is-device": false
        }
    ],
    "removes": [
        {
            "name": "16504914193",
            "list": "mobile-network",
            "subscriber": "joe.coffee",
            "is-device": true
        }
    ],
    "failures": {

    },
    "status": "success"
}
```

Update the Contents of a Specific Line List

Use the following **POST** API to add and remove lines to and from a specified line list:

```
SSM> POST /line-list/<listId>/nodes
{
    "add": [
        {
            "name": "<lineId>",
            "subscriber": "<accountId>"
        }
   ],
   "remove": [
        {
            "name": "<lineId>",
            "subscriber": "<accountId>"
        }
    ]
}
```

NOTE: The term “line” refers to the Network identifier. Each line in a line list is referred to as a “node.”

Fields

Field	Description
<*listId*>	Required. String that identifies the line list to update.
"name"	Required. Network identifier associated with the line you are adding to or removing from a line list. Replace the <*lineId*> argument with an address that acts as a network identifier for the device.
"subscriber"	Required. Identifies the subscriber account to which the updates apply.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows information related to the lines that were added to and removed from the specified line list. If any part of the operation fails, the output provides information about those failures:

```
{
    "adds": [
        {
            "name": "<lineId>",
            "list": "<listId>",
            "subscriber": "<accountId>"
            "is-device": <boolean>
        }
    ],
    "removes": [
        {
            "name": "<lineId>",
            "list": "<listId>",
            "subscriber": "<accountId>"
            "is-device": <boolean>
        }
    ],
    "failures": {
        "adds": [
            {
                "name": "<lineId>",
                "list": "<listId>",
                "subscriber": "<accountId>"
            }
        ],
        "removes": [
            {
                "name": "<lineId>",
                "list": "<listId>",
                "subscriber": "<accountId>"
            }
```

```
            ]
        },
        "status": "string"
}
```

The table that follows describes the fields in the API response:

Field	Description
"adds"	Provides the following information about the lines that are successfully added to the specified line list: ● "name"— Network identifier associated with the line that is successfully added to the line list. ● "list"—Identifies the line list to which the specified line is added. ● "subscriber"—Identifies the subscriber account to which the updates apply. ● "is-device"—Boolean value indicating whether the specified line is a device. A value of true indicates the line node is treated as a device. A value of false indicates the line node is not treated as a device. **NOTE:** For fixed line lists, the "is-device" field is always false. Only mobile networks can have lines that are devices. **NOTE:** The response does not include the “adds” fields the API does not include an **“add”** request.
"removes"	Provides the following information about the lines that are successfully removed from the specified line list: ● "name"— Network identifier associated with the line that is successfully removed from the line list. ● "list"—Identifies the line list from which the specified line is removed. ● "subscriber"—Identifies the subscriber account to which the update applies. ● "is-device"—Boolean value indicating whether the specified line node needs to be treated as a device. A value of true indicates that the line node is a device. A value of false indicates that the line node is not a device. **NOTE:** For fixed line lists, the "is-device" field is always false. Only mobile networks can have lines that are devices. **NOTE:** The response does not include the “removes” fields the API does not include a **“remove”** request.

Field	Description
"failures"	Provides the following information about operation attempts that failed: • “adds” or “removes”—Indicates whether the failure pertains to an **“add”** or **"remove"** request. • "name"— Network identifier associated with the line that failed to be added to or removed from the line list. • "list"—Identifies the line list for which the failed operation attempt occurred. • "subscriber"—Identifies the subscriber account to which the failed updates apply.
"status"	Indicates whether the entire operation succeeded, failed, or was partially successful.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Examples

The example that follows updates the **mobile-network** line list as follows:

- Adds a line node with the network ID **11223334444** to the line list called **mobile-network**.
- Removes a line node with the network ID **67778889999** from the list called **mobile-network**.

```
SSM> POST /line-list/mobile-network/nodes {"add": [{ "name":
"11223334444", "subscriber": "joe.coffee"}], "remove": [{ "name":
"67778889999", "subscriber": "joe.coffee"}]}
{
    "adds": [
        {
            "name": "11223334444",
            "list": "mobile-network",
            "subscriber": "joe.coffee",
            "is-device": true
        }
    ],
    "removes": [
        {
            "name": "67778889999",
            "list": "mobile-network",
            "subscriber": "joe.coffee",
            "is-device": true
        }
    ],
```

```
    "failures": {

    },
"status": "success"
}
```

The example that follows adds a line node to the **mobile-network** list for a subscriber called **joe.coffee**; this API adds a device with the network ID **MSISDN1** to the **mobile-network** list:

```
SSM> POST /line-list/mobile-network/nodes {"add": [{"name":
"123456789", "subscriber": "joe.coffee"}]}
{
    "adds": [
        {
            "name": "MSISDN1",
            "list": "mobile-network",
            "subscriber": "joe.coffee",
            "is-device": true
        }
    ],
    "failures": {
    },
    "status": "success"
}
```

Retrieve the Contents of all Line Lists Associated with a Subscriber Account

Use the following **GET** API to retrieve general information about the lines that are associated with a particular SIA SMB subscriber account:

```
SSM> GET /account/<accountId>/line
```

NOTE: The term “line” refers to the Network identifier.

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber for which to display line list information.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows general information about the lines that are associated with the specified subscriber account:

```
{
   "total":<integer>,
   "limit":<integer>,
    "content": [
        {
            "name":"<lineId>",
            "list":"<listId>",
            "subscriber": "<accountId>"
            "Is-device": <boolean>
        },
        ...
   ]
}
```

The table that follows describes the fields in the API response:

Field	Description
"total"	Integer indicating the number of lines associated with the subscriber account.
"limit"	Integer indicating the maximum number of lines that can be included in the retrieved output. **NOTE:** See the "Limiting Retrieved Results for GET Requests" section in Chapter 1, "Introduction to the SSM Restful API," for information about how to set the limit field for **GET** requests.
"content"	Displays the following information for each line: • "name"— Network identifier associated with the subscriber line for which list information is displayed. • "list"— String that identifies the line list to which the specified subscriber line belongs. • "subscriber"— Identifies the subscriber account to which the list belongs. • is-device—Boolean value indicating whether the specified line is a device. A value of true indicates the line node is treated as a device. A value of false indicates the line node is not treated as a device. **NOTE:** For fixed line lists, the "is-device" field is always false. Only mobile networks can have lines that are devices.

Example

The example that follows displays general information about all the lines configured for a subscriber account called **joe.coffee**:

```
SSM> GET /account/joe.coffee/line
{
    "total": 2,
    "limit": 2,
    "content": [
        {

            "name":"12223334444",
            "list":"fixed-network",
            "subscriber": "joe.coffee",
            "is-device":false
        },
        {
            "name": "87902409974",
            "list": "mobile-network",
            "subscriber": "joe.coffee",
            "is-device": true
        }

    ]
}
```

Retrieve all Line Lists Currently in your SSM

Use the following **GET** API to retrieve information pertaining to all line lists configured in your SSM:

```
SSM> GET /line-list
```

Fields

None

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output displays all line lists currently in the SSM and includes whether the specified list contain line nodes that are treated as devices:

```
{
    "total": <integer>,
    "limit": <integer>,
    "content": [
        {
            "name": "<lineId>",
            "has-devices": <boolean>
        },
        ...
    ]
}
```

The table that follows describes the fields in the API response:

Field	Description
“total”	Integer indicating the number of line lists in the SSM.
“limit”	Integer indicating the maximum number of lines that can be included in the retrieved output. **NOTE:** See the "Limiting Retrieved Results for GET Requests" section in Chapter 1, “Introduction to the SSM Restful API," for information about how to set the limit field for **GET** requests.
“content”	Displays the following information for each line list: ● “name”—String that identifies the line list. ● “has-devices”—Boolean value indicating whether the specified list contains line nodes that are treated as devices. A value of **true** indicates that the line nodes for this list are treated as devices. A value of **false** indicates that the line nodes for this list are not treated as devices. **NOTE:** For fixed line lists, the "has-devices" field is always false. Only mobile networks can have lines that are devices.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows retrieves all line lists for an SSM:

```
SSM> GET /line-list
{
    "total": 2,
    "limit": 2,
    "content": [
        {
            "name": "mobile-network",
            "has-devices": true
        },
        {
            "name": "fixed-network",
            "has-devices": false
        }
    ]
}
```

Retrieve the “has-devices” Value for a Specific Line List

Use the following **GET** API to retrieve the “has-devices” value for a specific line list (where the “has-devices” value indicates whether the line nodes in a list are treated as devices):

```
SSM> GET /line-list/<listId>
```

Fields

Field	Description
<listId>	Required. String that identifies the line list for which to display the “has-devices” value.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows the line list name and whether the line nodes for that list are treated as devices:

```
{
  "name":"<listId>",
  "has-devices":<boolean>
}
```

The table that follows describes the fields in the API response:

Field	Description
“name”	String that identifies the line list. This field matches the string you specified for the ***<listId>*** argument in the API call.
“has-devices”	Boolean value indicating whether the specified list contains line nodes that are treated as devices: • true—Indicates that the line nodes for this list are treated as devices. • false—Indicates that the line nodes for this list are not treated as devices. **NOTE:** For fixed line lists, the "has-devices" field is always false. Only mobile networks can have lines that are devices.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows displays the "has-devices" field setting for a line-list called **mobile-network**; in this example, the "has-devices" field is set to **true**, indicating the **mobile-network** line-list contains lines that are treated as devices:

```
SSM> GET /line-list/mobile-network
{
  "name":"mobile-network",
  "has-devices":true
}
```

Retrieve General Information About a Line

Use the following **GET** API to retrieve general information for a particular line (for example, the associated subscriber account identifier):

```
SSM> GET /line-list/<listId>/line/<lineId>
```

NOTE: A line represents a device requiring access to the SIA SMB service. Each line has an associated line ID.

Fields

Field	Description
<listId>	Required. String that identifies the line list for which to display configuration information.
<lineId>	Required. String that identifies the line for which to display configuration information. Replace the ***<lineId>*** argument with the network identifier that is associated with a line node. **NOTE:** Use the ***GET /account/<accountId>/line*** API to see a list of all line identifiers associated with a particular subscriber account; see the "Retrieve the Contents of All Line Lists Associated with a Subscriber Account" section for more information.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output contains general configuration information for the specified line:

```
{
   "name": "<lineId>",
   "list": "<listId>",
   "subscriber": "<accountId>",
   "is-device": <boolean>
}
```

The table that follows describes the fields in the API response:

Field	Description
"name"	Identifies the line for which information is being displayed; this field matches the string you specify for the **<lineId>** argument in the API call.
"list"	Identifies the line list that contains the line for which information is displayed.
"subscriber"	Identifies the subscriber account that is associated with the line for which information is displayed.
"is-device"	Boolean value indicating whether the specified line is a device; the following values are possible: • true—Indicates the line node is treated as a device. • false—Indicates the line node is not treated as a device. **NOTE:** For fixed line lists, the "is-device" field is always false. Only mobile networks can have lines that are devices.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows displays sample output from the API; this example shows configuration details for a line that has the line identifier **36738876126**, which belongs to the **mobile-network** line list:

```
SSM> GET /line-list/mobile-network/line/36738876126
{
    "name": "36738876126",
    "list": "mobile-network",
    "subscriber": "joe.coffee",
    "is-device": true
}
```

Display General Information About all Lines in a Line List

Use the following **GET** API to retrieve general information about all the lines in a line list:

```
SSM> GET /line-list/<listId>/nodes
```

NOTE: The term “line” refers to the Network identifier. Each line in a line list is referred to as a “node.”

Fields

Field	Description
<listId>	Required. String that identifies the line list for which to display line configuration information.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows configuration information for all the lines in a line list:

```
{
   "total": <integer>,
   "limit": <integer>,
   "offset": <integer>,
   "content": [
       {
           "name": "<lineId>",
           "list": "<listId>",
           "subscriber": "<accountId>",
```

```
            "is-device": <boolean>
        },
        ...
    ]
}
```

The table that follows describes the fields in the API response:

Field	Description
"total"	Total number of lines the specified line list contains.
"limit"	Indicates the maximum number of lines (nodes) that can be included in the API response. **NOTE:** See the "Limiting Retrieved Results for GET Requests" section in Chapter 1, "Introduction to the SSM Restful API," for information about how to set the limit field for **GET** requests.
"offset"	Specifies the number of lines (nodes) omitted from the API response. For example, if you set the offset to be **2**, the API response excludes the first two lines in the specified line list. **NOTE:** By default, if you do not append the offset and limit query parameters to a **GET** request, the API returns all of the requested lines that exist in the specified line-list. See the "Limiting Retrieved Results for GET Requests" section in Chapter 1, "Introduction to the SSM Restful API," for information about how to set the limit field for **GET** requests.
"name"	Identifies the line for which information is displayed; this field matches the string you specify for the **<lineId>** argument in the API call.
"list"	Identifies the line list that contains the line identified in the "name" field.
"subscriber"	Identifies the subscriber account that is associated with the line identified in the "name" field.
"is-device"	Boolean value indicating whether the specified line is a device; the following values are possible: • true—Indicates that the line node is treated as a device. • false—Indicates that the line node is not treated as a device. **NOTE:** For fixed line lists, the "is-device" field is always false. Only mobile networks can have lines that are devices.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows displays the devices that are assigned to the **mobile-network** line list:

```
SSM> GET /line-list/mobile-network/nodes
{
    "total": 60,
    "limit": 5,
    "offset": 0,
    "content": [
        {
            "name": "40990621157",
            "list": "mobile-network",
            "subscriber": "joe.coffee",
            "is-device": true
        },
        {
            "name": "57519913676",
            "list": "mobile-network",
            "subscriber": "joe.coffee",
            "is-device": true
        },
        {
            "name": "91535757856",
            "list": "mobile-network",
            "subscriber": "joe.coffee",
            "is-device": true
        },
        {
            "name": "41843358703",
            "list": "mobile-network",
            "subscriber": "joe.coffee",
            "is-device": true
        },
        {
            "name": "50227473836",
            "list": "mobile-network",
            "subscriber": "joe.coffee",
            "is-device": true
        }
    ]
}
```

Remove a Line List from the SSM

Use the following **DELETE** API to remove a specified line list from the SSM:

```
SSM> DELETE /line-list/<listId>
```

Fields

Field	Description
<listId>	Required. String that identifies the line list to remove from the SSM.

Expected Response

The SSM deletes the specified line list as requested.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows removes a line list called **mobile-network** from the SSM:

```
SSM> DELETE /line-list/mobile-network
```

Chapter 4: Subscriber Account Attribute APIs

This chapter describes the SSM API calls you can use to manage subscriber account attributes. These APIs perform the following tasks:

- Update subscriber account attributes
- Modify a specific subscriber account attribute
- Retrieve subscriber account attribute values
- Retrieve the current value of a subscriber account attribute
- Return all subscriber account attribute values to the default settings
- Return a specific subscriber account attribute to the default value

The table that follows describes attributes you can set on an SIA SMB subscriber account; all attributes are set to null by default which means the attributes use the default settings:

SSM Attributes

Attribute	Description
nom.smb.businessname	String specifying a business name to use on the SIA SMB portal application. The default is **null**, where no name is displayed.
nom.smb.email	String that specifies an email address to which the SSM sends SIA SMB scheduled reports. The email is formatted as <*local-part*>@<*domain.top-level-domain*>.
nom.smb.email.frequency	String that indicates the frequency at which the SIA SMB application sends an Internet usage report for the specified subscriber account to the specified email address. The following values are possible: • **P1D**—Sends an Internet usage report email daily. • **P1W**—Sends an Internet usage report weekly • **P1M**—Sends an Internet usage report monthly. • **off**—The SIA SMB does not send a report. • **null**—This is the default setting and is the same as if you set the **nom.smb.email.frequency** field to **off**; in other words, the SIA SMB does not send a report.

Attribute	Description
nom.smb.format24h	Boolean value indicating whether the SMB portal application uses a 12- or 24-hour format. The following values are possible: • **true**—Indicates the SMB portal application time values display in the 24-hour format. • **false**—Indicates the SMB portal application time values display in the 12-hour format. • **null**—This is the default setting, indicating the SIA SMB time values display in the 12-hour format.
nom.smb.language	String that specifies the language the SIA SMB portal application is configured to use. The default setting is **null**, which means the SIA SMB subscriber portal application uses the language configured in the appConfig.json file. Language values which are not set to **null** use the format <*language-code*>-<*region-code*>, (for example **ru-RU** or **en-US**). See the document *ISO 639-1:2002 Codes For The Representation Of Names Of Languages - Part 1: Alpha-2 Code* for a list of possible language codes.

Update Subscriber Account Attributes

Use the following **PUT** API to set or update all subscriber account attribute values:

```
SSM> PUT /account/<accountId>/attribute
[
    {
        "name": "<attributeName>",
        "value": "<attributeValue>"
    },
    ...
]
```

NOTE: Use the **GET /account/<*accountId*>/attribute** API to retrieve all attribute values for a subscriber account; see the "Retrieve Subscriber Account Attribute Values" section for details on the **GET /account/<*accountId*>/attribute** API.

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account for which to update account attributes.
"name"	At least one attribute "name" field is required for the API to be valid. The **"name"** field identifies the attribute you want to set or update. Replace the ***<attributeName>*** argument with one of the following attribute names: • **nom.smb.email** • **nom.smb.language** • **nom.smb.businessname** • **nom.smb.email.frequency** • **nom.smb.format24h** See the "SSM Attributes" table for a detailed description of all attributes and possible attribute values.
"value"	Required for each attribute you are setting or updating. The **"value"** field specifies a value for the specified account attribute. Replace the ***<attributeValue>*** argument with an appropriate value for the attribute. The "SSM Attributes" table describes the values you can specify for each attribute.

Expected Response

The SSM updates the specified subscriber account attributes as requested.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows demonstrates how to update an individual attribute for a subscriber account. This example configures the language attribute value for a subscriber account called **joe.coffee** to be English (**en-US**):

```
SSM> PUT /account/joe.coffee/attribute [{"name": "nom.smb.language",
"value": "en-US"}]
```

The example that follows demonstrates how to update multiple attributes for a subscriber account. This example performs the following configuration for a subscriber account called **joe.coffee**:

- Sets the **nom.smb.businessname** attribute value to **Joe's coffee, inc**.
- Sets the **nom.smb.email** attribute value to **joe.coffee@example.com**.

```
SSM> PUT /account/joe.coffee/attribute [{"name": "nom.smb.businessname",
"value": "Joe's coffee"}, {"name": "nom.smb.email", "value":
"joe.coffee@example.com"}]
```

Modify a Specific Subscriber Account Attribute

Use the following **PUT** API to set or update a specific subscriber account attribute:

```
SSM> PUT /account/<accountId>/attribute/<attributeName>
{
    "name": "<attributeName>",
    "value":"<attributeValue>"
}
```

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account for which to update the specified attributes.
<attributeId>	Required. Identifies the attribute you want to set or update. Replace the ***<attributeName>*** argument with one of the following attribute names: • **nom.smb.businessname** • **nom.smb.email** • **nom.smb.email.frequency** • **nom.smb.format24h** • **nom.smb.language** See the "SSM Attributes" table for a detailed description of all attributes and possible attribute values.
"name"	Required. Identifies the attribute you want to set or update. Replace the ***<attributeName>*** argument with the same attribute name (a string) you specified for the ***<attributeName>*** argument in the API call.

Field	Description
"value"	Required. Sets or updates the value for the specified account attribute. Replace the ***<attributeValue>*** argument with an appropriate value for the attribute. The "SSM Attributes" table describes the values you can specify for each attribute.

Expected Response

The SSM sets or updates the specified subscriber account attribute as requested.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows sets the **nom.smb.email** attribute to the email address **joe.coffee@example.com**:

```
SSM> PUT /account/joe.coffee/attribute/nom.smb.email
{"name":"nom.smb.email", "value":"joe.coffee@example.com"}
```

Retrieve Subscriber Account Attribute Values

Use the following **GET** API to see all current attribute values for a subscriber account:

```
SSM> GET /account/<accountId>/attribute
```

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account for which to retrieve current attribute information.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows information related to the subscriber account attributes, including how each attribute is currently set as shown in the example that follows:

```
{
    "total": <integer>,
    "limit": <integer>,
    "content": [
        {
            "name": "nom.smb.businessname",
            "value": "<attributeValue>"
        },
        {
            "name": "nom.smb.email",
            "value": "<attributeValue>"
        },
        {
            "name": "nom.smb.email.frequency",
            "value": "<attributeValue>"
        },
        {
            "name": "nom.smb.format24h",
            "value": "<attributeValue>"
        },
        {
            "name": "nom.smb.language",
            "value": "<attributeValue>"
        }
    ]
}
```

The table that follows describes the fields in the API response:

Field	Description
"total"	Integer indicating the number of attributes set for the subscriber account.
"limit"	Integer indicating the maximum number of attributes that can be included in the API response. **NOTE:** See the "Limiting Retrieved Results for GET Requests" section in Chapter 1, "Introduction to the SSM Restful API," for information about how to set the limit field for **GET** requests.
"content"	Displays the following information for each subscriber account attribute: • "name"—String that identifies the attribute being displayed. • "value"—String indicating how the specified attribute is currently set. In the API response, the unset attributes have a "value" field setting of **null**.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows retrieves the current attribute values for a subscriber account called **joe.coffee**. In this example the "limit" field indicates there are seven (**7**) possible attributes; however, the "total" field indicates two (**2**) of those attributes are set for the subscriber:

```
SSM> GET /account/joe.coffee/attribute
{
    "total": 2,
    "limit": 7,
    "content": [
        {
            "name": "nom.smb.email",
            "value": "joe.coffee@example.com"
        },
        {
            "name": "nom.smb.language",
            "value": null
        },
        {
            "name": "nom.smb.businessname",
            "value": "Joe's coffee, inc"
        },
        {
            "name": "nom.smb.email.frequency",
            "value": null
        },
        {
            "name": "nom.smb.format24h",
            "value": null
```

```
            }
        ]
}
```

Retrieve the Current Value of a Subscriber Account Attribute

Use the following **GET** API to see how an attribute is currently set for a particular subscriber account:

```
SSM> GET /account/<accountId>/attribute/<attributeName>
```

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account for which to retrieve the specified attribute's value.
<attributeName>	Required. Identifies the attribute you want to display. Replace the ***<attributeName>*** argument with one of the following attributes: • **nom.smb.businessname** • **nom.smb.email** • **nom.smb.email.frequency** • **nom.smb.format24h** • **nom.smb.language** See the "SSM Attributes" table for a detailed description of all attributes and possible attribute values.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows how a specific attribute is currently set for the specified subscriber account:

```
{
    "name": "<attributeName>",
    "value": "<attributeValue>"
}
```

The table that follows describes the fields in the API response:

Field	Description
"name"	Identifies the attribute for which the current value is displayed. The response displays the value for the attribute you specify in the API request. See the "SSM Attributes" table for a detailed description of all attributes and possible attribute values.
"value"	Indicates the current setting for the specified attribute; the *<attributeValue>* is formatted as a string.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows retrieves the current setting for the **nom.smb.email** attribute (which specifies the email address to which the SSM sends SIA SMB scheduled reports) for a subscriber account called **joe.coffee**:

```
SSM> GET /account/joe.coffee/attribute/nom.smb.email
{
    "name": "nom.smb.email",
    "value": "joe.coffee@example.com"
}
```

Return all Subscriber Account Attribute Values to the Default Settings

Use the following **DELETE** API to return all attributes to the default settings for a specific subscriber account:

```
SSM> DELETE /account/<accountId>/attribute
```

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account for which to clear current attribute settings, returning all attributes to their default values.

See the "SSM Attributes" table for a detailed description of SSM all attributes and possible attribute values.

Expected Response

The SSM API returns all attributes to the default settings for the subscriber account.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows clears all attributes for a subscriber account called **joe.coffee**, returning the attributes to a default value of null:

```
SSM> DELETE /account/joe.coffee/attribute
```

Return a Specific Subscriber Account Attribute to the Default Value

Use the following **DELETE** API to return a specific subscriber account attribute to the default value of null:

```
SSM> DELETE /account/<accountId>/attribute/<attributeName>
```

Fields

Field	Description
<accountId>	Required. Identifies the subscriber account for which to clear the specified attribute's current setting; this API returns the specified attribute to the default value.
<*attributeName*>	Required. String that identifies the attribute to return to its default setting.

Expected Response

The SSM API returns the specified attribute to the default null setting for the subscriber account.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows clears the **nom.smb.format24h** attribute for a subscriber account called **joe.coffee**, returning the **nom.smb.format24h** attribute to its default value of null (where the SIA SMB application uses the 12-hour time format):

```
SSM> DELETE /account/joe.coffee/attribute/nom.smb.format24h
```

.

Chapter 5: Profile (Device Group) APIs

The SIA SMB service uses profiles to configure different protection settings for disparate groups of devices.

NOTE: In the SIA SMB subscriber portal application, profiles are called “device groups.”

There are two SIA SMB service types:

- Single profile service, where an SIA SMB subscriber configures one profile and applies that profile to any devices accessing the network.
- Multiple profile service, where an SIA SMB subscriber can configure multiple profiles that apply different levels of protection to disparate groups of devices. Each profile administers a unique set of protections to the associated devices. For example, an SIA SMB subscriber can have one profile for employee devices, another more secure profile that applies to guest devices temporarily accessing the network, and so forth.

This chapter describes how to perform the following tasks:

This chapter describes the APIs for configuring SIA SMB profiles, which are known as “device groups” in the SIA SMB subscriber portal application.

Configurable Profile Fields

Each profile has a set of configurable parameters, or fields, that get set during profile creation or, with some fields, later on after a profile is created. To create a profile, you only need to specify a profile name and protection level; all other parameters are optional and, if not specified, use the default settings.

NOTE: In the SIA SMB subscriber portal application, profiles are called “device groups.”

The table that follows describes configurable profile fields the SSM API supports. In addition, you can use the following APIs to see how these fields are currently set for a subscriber:

- **GET /sb/<*accountId*>/profile**
- **GET /sb/<*accountId*>/profile/<*profileName*>**

Table 5-1: Configurable Profile Fields

Field	Description
"id"	Numerical identifier (ID) assigned to the profile. The SSM automatically assigned this numerical ID to the profile when creating the profile.
"name"	Text string name that identifies the profile. **NOTE:** This field is required for profile creation.
"protection"	Keyword that identifies the profile's protection level. Replace the ***<protectionLevel>*** argument with the name of the protection level the profile enforces. The service provider determines the name of each protection level in an SIA SMB service. For example, common protection level names are **strict**, **medium**, and **light**. **NOTE:** This field is required for profile creation.
"protection-enabled"	Boolean value that indicates whether protection is enabled (**true**) or disables (**false**) protection at the profile-level.
"internet-access"	Boolean value that indicates whether internet access is enabled (true) or disabled (false) for all devices associated with the specified subscriber account.
"inspect-stream"	Boolean value that indicates whether an inspect-stream policy is bound to a profile. A value of **true** indicates an inspect-stream policy is bound to the profile; a value of **false** indicates the inspect-stream policy is not bound to the profile. The default value is false (the inspect-stream policy is not bound to the profile).
"safe-search"	Boolean value that indicates whether one or more safe search services (Youtube, google, and bing) are enabled. A value of **true** indicates at least one safe search service is enabled; a value of **false** indicates no safe search services are enabled.
"https-termination"	Boolean value that indicates whether the Proxy server evaluates subscriber traffic for the secure socket layer (SSL) termination service (which is also known as the HTTPS termination service). A **true** value indicates SSL termination service evaluation is enabled for devices attached to the profile; a **false** value indicates SSL termination service evaluation is disabled for devices that are attached to the profile. **Caution!** SSL (HTTPS) termination service evaluation is a limited feature that is not commonly configured; set the "https-termination" field to **false** unless Akamai has granted you explicit contracted permissions to enable that feature.
"safe-search-services"	Indicates whether the Youtube, Google, and Bing safe search services are enabled (**true**) or disabled (**false**) on devices which have the specified profile attached.

Field	Description
"default"	Boolean value that indicates whether the profile is the default profile attached to devices until a non-default profile is manually specified. A value of **true** indicates the profile is the default profile; a value of **false** indicates the profile is not the default profile.

Create Profiles for a Subscriber Account

Use the following **POST** API to create one or more profiles for a subscriber account:

```
SSM> POST /sb/<accountId>/profile
[
    {
        "name": "<profileName>",
        "protection": "<protectionLevel>",
        "protection-enabled": <boolean>,
        "internet-access": <boolean>,
        "inspect-stream": <boolean>,
        "safe-search": <boolean>,
        "https-termination": <boolean>,
        "safe-search-services": {
            "youtube": <boolean>,
            "google": <boolean>,
            "bing": <boolean>
        },
        "default": <boolean>,
    }
  ...
]
```

NOTE: In the SIA SMB subscriber portal application, profiles are called “device groups.”

Caution! If one part of the configuration fails, the entire API fails.

Fields

Field	Description
General profile fields	
<accountId>	Required. Identifies the subscriber account for which to perform profile configuration.
"name"	Required. Specifies a name for the profile. Replace the ***<profileName>*** argument with a text string that is unique in your service and helps you easily identify the profile. **NOTE** Profile (device group) names are case sensitive.

Field	Description
"protection"	Required. Sets a profile's protection level. Replace the ***<protectionLevel>*** argument with the name of the protection level the profile enforces. The service provider determines the name of each protection level in an SIA SMB service. For example, common protection level names are **strict**, **medium**, and **light**.
"protection-enabled"	Optional. Specifies a boolean value that enables (**true**) or disables (**false**) protection at the profile-level.
Additional general profile fields	
"age-group"	Optional. Assigns an age group to the profile.
"internet-access"	Optional. Specifies a boolean value that enables (**true**) or disables (**false**) internet access for all devices attached to the profile.
"inspect-stream"	Optional. Specifies a boolean value that dictates whether an inspect-stream policy is bound to a profile; specify **true** to bind an inspect-stream policy to the profile or **false** to leave the policy unbound to the profile. The default value is **false** (the inspect-stream policy is not bound to the profile).
"safe-search"	Optional. Specifies a boolean value that indicates that one or more safe search services (youtube, google, and bing) are enabled. A value of **true** indicates at least one safe search service is enabled; a value of **false** indicates no safe search services are enabled.
"https-termination"	Optional. Specifies a boolean value that enables (**true**) or disables (**false**) SSL (HTTPS) termination service evaluation on the devices that have the profile attached. When enabled, the Proxy server evaluates device traffic for the SSL (HTTPS) termination service. **Caution!** SSL (HTTPS) termination service evaluation is a limited feature that is not commonly configured; set the **"https-termination"** field to **false** unless Akamai has granted your company explicit contracted permissions to enable that feature.
"safe-search-services"	Optional. Specifies a boolean value that enables (**true**) or disables (**false**) the Youtube, Google, and Bing safe search services on devices which have this profile attached. You need to set a separate boolean value for each service as shown in the following example: `"safe-search-services": {"youtube": false, "google": false, "bing": false}`

Field	Description
"default"	Optional. Sets the profile as the default profile. Replace the ***<boolean>*** argument with a boolean value where **true** designates the profile as the default profile, and **false** designates the profile is not the default profile.

NOTE: When configuring multiple profiles in a single API, keep in mind that if one part of the configuration fails, the entire API fails.

Expected Response

The SSM creates the profile as requested and returns HTTP code 200 with a JSON based response body displaying the new profile's configuration fields as shown in the example that follows:

```
{
    "id": "<idNumber>",
    "name": "<profileName>",
    "protection": "<protectionLevel>",
    "protection-enabled": <boolean>,
    "devices": [
    ],
    "schedules": [
    ],
     "age-group": "<groupName>",
     "internet-access": <boolean>,
     "inspect-stream": <boolean>,
     "safe-search": <boolean>,
     "https-termination": <boolean>,
      "safe-search-services": {
         "google": <boolean>,
         "youtube": <boolean>,
          "bing": <boolean>
          },
    "default": <boolean>
    }
]
```

See the "Configurable Profile Fields" section for a description of the fields displayed in the API response body.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Examples

This section provides examples that demonstrate how to configure the following types of profiles:

- A profile with minimal profile configuration.
- A default profile that enables SafeSearch services and other optional settings.

Example: Create a Profile with Minimal Profile Configuration

The example that follows creates and configures a simple profile for a subscriber account called **joe.coffee**; this profile provides a **medium** level of protection to the attached devices:

```
SSM> POST /sb/joe.coffee/profile [{"name": "new-sb-profile",
"protection": "medium"}]
[
    {
        "id": "05c0fafb-2282-4bc4-b9ac-3c6719911449",
        "name": "new-sb-profile",
        "protection": "medium",
        "protection-enabled": true,
        "devices": [

        ],
        "schedules": [

        ],
        "internet-access": true,
        "inspect-stream": false,
        "safe-search": false,
        "https-termination": false,
        "safe-search-services": {
                "bing": false,
                "google": false,
                "youtube": false
        }
    }
]
```

Example: Create a Default Profile that Enables SafeSearch Services and Specifies Other Optional Settings

The example that follows demonstrates how to create a default profile with profile protections. In other words, the profile does not provide the configured protections to the attached devices until an administrator enables protection at the profile-level (by setting the "protection-enabled" field value to **true**). When protections are enabled, the profile provides the following protections to all attached devices:

- Disables internet access.
- Enables the Bing and YouTube SafeSearch services.
- Binds an inspect-stream policy to the profile.

```
SSM> POST /sb/joe.coffee/profile [{"name": "sb-profile",
"protection": "medium", "default": false, "protection-enabled":
false,  "internet-access": false, "inspect-stream": true,
"safe-search": true, "safe-search-services": {"bing": false,
"youtube": true}, "default": true}]
```

```
[
    {
        "id": "9ee4c3ed-c97d-4f9e-91f2-01f32169c912",
        "name": "sb-profile",
        "protection": "medium",
        "protection-enabled": false,
        "devices": [

        ],
        "schedules": [

        ],
        "internet-access": false,
        "inspect-stream": true,
        "safe-search": true,
        "https-termination": false,
        "safe-search-services": {
            "bing": false,
            "youtube": true,
            "google": true
        },
        "default": true
    }
]
```

Update a Profile

Use the following **PUT** API to update an existing SIA SMB profile:

```
SSM> PUT /sb/<accountId>/profile/<profileName>
{
    "name": "<profileName>",
    "protection": "<protectionLevel>",
    "protection-enabled": <boolean>,
    "age-group": <groupId>,
    "internet-access": <boolean>,
    "inspect-stream": <boolean>,
    "safe-search": <boolean>,
    "https-termination": <boolean>,
    "safe-search-services": {
        "youtube": <boolean>,
        "google": <boolean>,
        "bing": <boolean>
    },
    "default": <boolean>,
}
```

NOTE: In the SIA SMB subscriber portal application, profiles are called “device groups.”

Caution! When configuring multiple profiles in a single API, keep in mind that if one part of the configuration fails, the entire API fails.

Fields

Field	Description
General profile fields	
<*accountId*>	Required. Identifies the subscriber account for which to perform profile configuration.
"name"	Required. Specifies a name for the profile. Replace the **<*profileName*>** argument with a text string that is unique in your service and helps you easily identify the profile. **NOTE:** In the SIA SMB subscriber portal application, profiles are called “device groups.” **NOTE:** The profile name is case-sensitive.
"protection"	Required. Sets a profile's protection level. Replace the **<*protectionLevel*>** argument with the name of the protection level the profile enforces. The service provider determines the name of each protection level in an SIA SMB service. For example, common protection level names are **strict**, **medium**, and **light**.
"protection-enabled"	Optional. Specifies a boolean value that enables (true) or disables (false) protection at the profile-level.
Additional general profile fields	
"age-group"	Optional. Assigns an age group to the profile.
"internet-access"	Optional. Specifies a boolean value that enables (**true**) or disables (**false**) internet access for all devices associated with the subscriber account.
"inspect-stream"	Optional. Specifies a boolean value that dictates whether an inspect-stream policy is bound to a profile; specify **true** to bind an inspect-stream the policy to the profile or **false** to leave the policy unbound to the profile. The default value is **false** (the inspect-stream policy is not bound to the profile).
"safe-search"	Optional. Specifies a boolean value that indicates that one or more safe search services (youtube, google, and bing) are enabled. A value of **true** indicates at least one safe search service is enabled; a value of **false** indicates no safe search services are enabled.

Field	Description
"https-termination"	Optional. Specifies a boolean value that enables (**true**) or disables (**false**) SSL (HTTPS) termination service evaluation on the devices that have the profile attached. When enabled, the Proxy server evaluates device traffic for the SSL (HTTPS) termination service. **Caution!** SSL (HTTPS) termination service evaluation is a limited feature that is not commonly configured; set the **"https-termination"** field to **false** unless Akamai has granted your company explicit contracted permissions to enable that feature.
"safe-search-services"	Optional. Specifies a boolean value that enables (**true**) or disables (**false**) the Youtube, Google, and Bing safe search services on devices which have this profile attached. You need to set a separate boolean value for each service as shown in the following example: `"safe-search-services": {"youtube": false, "google": false, "bing": false}`
"default"	Optional. Sets the profile as the default profile. Replace the ***<boolean>*** argument with a boolean value where **true** designates the profile as the default profile, and **false** designates the profile is not the default profile.

Expected Response

The SSM updates the profile as requested and returns HTTP code 200 with a JSON based response body displaying all profile configuration fields (including those fields updated by the API). The following example shows what the response looks like:

```
{
    "id": "<idNumber>",
    "name": "<profileName>",
    "protection": "<protectionLevel>",
    "protection-enabled": <boolean>,
    "devices": [
    ],
    "schedules": [
    ],
    "age-group": "<groupName>",
    "internet-access": <boolean>,
    "inspect-stream": <boolean>,
    "safe-search": <boolean>,
    "https-termination": <boolean>,
    "safe-search-services": {
        "google": <boolean>,
        "youtube": <boolean>,
```

```
        "bing": <boolean>
    },
    "default": <boolean>
}
```

See the "Configurable Profile Fields" section for a description of the fields displayed in the API response body.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Examples

This section provides examples that demonstrate how to perform the following updates to an existing profile:

- Perform a simple profile update
- Change the name of a profile
- Enforce SafeSearch services on a profile

Example: Simple Profile Update

The example that follows shows a simple update to a profile (called **sb_group_visitor**) for a subscriber account called **joe.coffee**. This example sets the profile to be a non-default profile (by setting the "default" field to **false**) and sets protection level for the profile:.

NOTE: The **"protection"** field is required and you can use this field to specify the current protection level for the profile or update profile to enforce a different protection level.

```
SSM> PUT /sb/joe.coffee/profile/sb_group_visitor {"name":"sb_group_visitor",
"protection":"strict", "default": false}
{
    "id": "74f33fba-90a2-494e-83b9-b61b9df05429",
    "name": "sb_group_visitor",
    "protection": "strict",
    "protection-enabled": true,
    "devices": [

    ],
    "schedules": [

    ],
    "age-group": "pi-ag1",
    "internet-access": true,
    "inspect-stream": false,
    "safe-search": false,
    "https-termination": false,
    "safe-search-services": {
        "google": false,
        "youtube": false,
        "bing": false
    },
```

```
    "default": false
}
```

Example: Change the Name of a Profile

The example that follows demonstrates how to change the name of an existing profile. This example changes the name of a profile (originally called **visitors**) to **guests**; in the API response, the “name” field (which is highlighted in the example) reflects the name change:

```
SSM> PUT /sb/joe.coffee/profile/visitors {"name":"guests",
"protection":"strict"}
{
    "id": "817c14d7-a6bb-4592-9ae4-0aaa9e9e5122",
    "name": "guests",
    "protection": "strict",
    "protection-enabled": true,
    "devices": [

    ],
    "schedules": [

    ],
    "internet-access": true,
    "inspect-stream": false,
    "safe-search": false,
    "https-termination": false,
    "safe-search-services": {
        "bing": false,
        "youtube": false,
        "google": false
    }
}
```

Example: Enforce SafeSearch Services on a Profile

The example that follows demonstrates how to update SafeSearch service settings in an existing profile. In this example, the **PUT** API updates an existing profile (called **sb_group_employee)** that belongs to a subscriber account called **joe.coffee.** The API enables the Bing, YouTube, and Google SafeSearch services for all devices attached to the **sb_group_employee** profile.

```
SSM> PUT /sb/joe.coffee/profile/sb_group_employee
{"name":"sb_group_employee", "default": false,"protection":"medium",
"internet-access": true, "inspect-stream": false,
"safe-search-services":{"bing": true, "youtube": true, "google": true}}
{
    "id": "64cffef5-8bf1-427b-aacc-9c475098a6e8",
    "name": "sb_group_employee",
```

```
    "protection": "medium",
    "protection-enabled": true,
    "devices": [

    ],
    "schedules": [

    ],
    "internet-access": true,
    "inspect-stream": false,
    "safe-search": true,
    "https-termination": false,
    "safe-search-services": {
        "bing": true,
        "youtube": true,
        "google": true
    },
    "default": false
}
```

Add, Remove, and Partially Update Profiles for a Subscriber Account

Use the following **PATCH** API to update parts of a specific profile that is attached to a particular subscriber account; you can perform multiple configuration tasks in a single API:

```
SSM> PATCH /sb/<accountId>/profile
{
    "update": [
        {
            "name": "<profileName>",
            "protection": "<protectionLevel>",
            "protection-enabled": <boolean>,
            "age-group": <groupId>,
            "internet-access": <boolean>,
            "inspect-stream": <boolean>,
            "safe-search": <boolean>,
            "https-termination": <boolean>,
            "safe-search-services": {
                "youtube": <boolean>,
                "google": <boolean>,
                "bing": <boolean>
            },
            "default": <boolean>,
        }
    ...

    ],
  ...
```

```
}
```

NOTE: In the SIA SMB subscriber portal application, profiles are called “device groups.”

Fields

Field	Description
General Profile Fields	
<*accountId*>	Required. Identifies the subscriber account for which to perform profile configuration.
"update"	Required. Indicates you want to update a device.
"name"	Required. Identifies the profile you want to add, update, or delete. Replace the **<*profileName*>** argument with the text string name of the desired profile. **NOTE:** The profile name is case-sensitive.
"protection"	Required. Sets a profile's protection level. Replace the **<*protectionLevel*>** argument with the name of the protection level the profile enforces. The service provider determines the name of each protection level in an SIA SMB service. For example, common protection level names are **strict**, **medium**, and **light**.
"protection-enabled"	Optional. Specifies a boolean value that enables (**true**) or disables (**false**) protection at the profile-level.
Additional General Profile Fields	
"age-group"	Optional. Assigns an age group to the profile.
"internet-access"	Optional. Specifies a boolean value that enables (**true**) or disables (**false**) internet access for all devices associated with the subscriber account.
"inspect-stream"	Optional. Specifies a boolean value that dictates whether an inspect-stream policy is bound to a profile; specify **true** to bind an inspect-stream the policy to the profile or **false** to leave the policy unbound to the profile. The default value is **false** (the inspect-stream policy is not bound to the profile).
"safe-search"	Optional. Specifies a boolean value that indicates that one or more safe search services (youtube, google, and bing) are enabled. A value of **true** indicates at least one safe search service is enabled; a value of **false** indicates no safe search services are enabled.

Field	Description
"https-termination"	Optional. Specifies a boolean value that enables (**true**) or disables (**false**) SSL (HTTPS) termination service evaluation on the devices that have the profile attached. When enabled, the Proxy server evaluates device traffic for the SSL (HTTPS) termination service. **Caution!** SSL (HTTPS) termination service evaluation is a limited feature that is not commonly configured; set the **"https-termination"** field to **false** unless Akamai has granted your company explicit contracted permissions to enable that feature.
"safe-search-services"	Optional. Specifies a boolean value that enables (**true**) or disables (**false**) the Youtube, Google, and Bing safe search services on devices which have this profile attached. You need to set a separate boolean value for each service as shown in the following example: `"safe-search-services": {"youtube": false, "google": false, "bing": false}`
"default"	Optional. Sets the profile as the default profile. Replace the ***<boolean>*** argument with a boolean value where **true** designates the profile as the default profile, and **false** designates the profile is not the default profile.

Expected Response

A successful response includes HTTP code 200 with a response body confirming the API is successful. The example that follows shows the JSON output returned in the API response body for requests that affect multiple profiles:

```
{
    "processed": [
        {
            "id": "<profileId>",
            "name": "<profileName>",
            "protection": "<protectionLevel>",
            "protection-enabled": <boolean>,
            "devices": [

            ],
            "schedules": [

            ],
            "internet-access": <boolean>,
            "inspect-stream": <boolean>,
            "safe-search": <boolean>,
            "https-termination": <boolean>
```

```
        }
    ],
"status": "success"
}
```

See the "Configurable Profile Fields" section for a description of the fields displayed in the API response body.

For additions and updates or a single profile, the response body includes configuration details for the new or updated profile as demonstrated in the following example:

```
{
    "status": "success"
}
```

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The examples in this section demonstrate how to use the **PATCH /sb/<*accountId*>/profile** API to perform the following tasks:

- Perform a simple profile update
- Update multiple profiles in a single API call

Perform a Simple Profile Update

The example that follows demonstrates how to use the **PATCH** API to perform a simple task. In this example, the API updates the protection level a profile (called **sb_group_visitor**) provides to all devices attached to the profile. When the API request is successful, the response shows the updated profile configuration (in this example, the highlighted "protection" field reflects the update):

```
SSM> PATCH /sb/joe.coffee/profile {"update": [{"name":"sb_group_visitor",
"protection": "strict"}]}
{
    "processed": [
        {
            "id": "96be5792-3199-4399-85b0-45b0a1eb06d6",
            "name": "child",
            "protection": "strict",
            "protection-enabled": true,
            "devices": [
            ],
            "schedules": [
            ],
            "internet-access": true,
            "inspect-stream": false,
            "safe-search": false,
            "https-termination": false
        }
```

```
    ],
"status": "success"
}
```

Update Multiple Profiles in a Single API Call

The example that follows demonstrates how to perform multiple profile configuration tasks in a single **PATCH** API request. This example updates two profiles (called **employees** and **visitors**):

```
SSM> PATCH /sb/joe.coffee/profile {"update": [{"name": "employees",
"default": true,  "https-termination": true, "id": "sample-id",
"inspect-stream": true, "internet-access": true, "protection":
"medium", "protection-enabled": true, "safe-search": false,
"subscriber-safety": true},{"name":"visitors", "protection":
"strict"}]}

{
    "processed": [
        {
            "id": "225ef85c-2f79-462b-9a3d-52829aacbfe6",
            "name": "employees",
            "protection": "medium",
            "protection-enabled": true,
            "devices": [

            ],
            "schedules": [

            ],
            "internet-access": true,
            "inspect-stream": true,
            "safe-search": false,
            "https-termination": true,
            "default": true
        },
        {
            "id": "ea3cce87-3bb9-454f-8dce-67fcb7a6b481",
            "name": "visitors",
            "protection": "strict",
            "protection-enabled": true,
            "devices": [

            ],
            "schedules": [

            ],
            "internet-access": true,
            "inspect-stream": false,
            "safe-search": false,
```

```
            "https-termination": false
        }
    ],
    "status": "success"
}
```

Partially Update a Specific Profile for a Subscriber Account

Use the following **PATCH** API to update individual fields for profiles attached to a specific subscriber account; this API is useful for performing multiple profile configuration tasks in a single API:

```
SSM> PATCH /sb/<accountId>/profile/<profileName>
{
    "name": "<profileName>",
    "protection": "<protectionLevel>",
    "protection-enabled": <boolean>,
    "internet-access": <boolean>,
    "inspect-stream": <boolean>,
    "safe-search": <boolean>,
    "https-termination": <boolean>,
    "safe-search-services": {
        "youtube": <boolean>,
        "google": <boolean>,
        "bing": <boolean>,
    "subscriber-safety": <boolean>,
    "default": <boolean>
}
```

NOTE: In the SIA SMB subscriber portal application, profiles are called “device groups.”

Fields

Field	Description
<*accountId*>	Required. Identifies the subscriber account for which to perform profile configuration.
<*profileName*>	Required. Identifies the profile you want to update. **NOTE:** The profile name is case-sensitive.
"name"	Optional. Renames the profile. Replace the <*profileName*> argument with the desired name. **NOTE:** Profile names are case-sensitive.

Field	Description
"protection"	Optional. Sets a profile's protection level. Replace the ***<protectionLevel>*** argument with the name of the protection level the profile enforces. The service provider determines the name of each protection level in an SIA SMB service. For example, common protection level names are **strict**, **medium**, and **light**.
"protection-enabled"	Optional. Specifies a boolean value that enables (**true**) or disables (**false**) protection at the profile-level.
"internet-access"	Optional. Specifies a boolean value that enables (**true**) or disables (**false**) internet access for all devices associated with the subscriber account.
"inspect-stream"	Optional. Specifies a boolean value that dictates whether an inspect-stream policy is bound to a profile; specify **true** to bind an inspect-stream the policy to the profile or **false** to leave the policy unbound to the profile. The default value is **false** (the inspect-stream policy is not bound to the profile).
"safe-search"	Optional. Specifies a boolean value that indicates that one or more safe search services (youtube, google, and bing) are enabled. A value of **true** indicates at least one safe search service is enabled; a value of **false** indicates no safe search services are enabled.
"https-termination"	Optional. Specifies a boolean value that enables (**true**) or disables (**false**) SSL (HTTPS) termination service evaluation on the devices that have the profile attached. When enabled, the Proxy server evaluates device traffic for the SSL (HTTPS) termination service. **Caution!** SSL (HTTPS) termination service evaluation is a limited feature that is not commonly configured; set the **"https-termination"** field to **false** unless Akamai has granted your company explicit contracted permissions to enable that feature.
"safe-search-services"	Optional. Specifies a boolean value that enables (**true**) or disables (**false**) the Youtube, Google, and Bing safe search services on devices which have this profile attached. You need to set a separate boolean value for each service as shown in the following example: `"safe-search-services": {"youtube": false, "google": false, "bing": false}`

Field	Description
"subscriber-safety"	Optional. Boolean value that enables (**true**) or disables (**false**) the Subscriber Safety application for the specified subscriber account.
"default"	Optional. Sets the profile as the default profile. Replace the ***<boolean>*** argument with a boolean value where **true** designates the profile as the default profile, and **false** designates the profile is not the default profile.

Expected Response

The SSM updates the profile as requested and returns HTTP code 200 with a JSON based response body indicating the updates are successful and displaying the updated profile fields. The example that follows shows the response body for a successful request:

```
{
    "successful": [
        {
            "field": "<fieldName>",
            "value": <value>
        },
        ...
    ],
    "status": "success"
}
```

See the "Configurable Profile Fields" section for a description of the fields displayed in the API response body.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows updates profile (called **new-sb-profile**) for a subscriber account called **joe.coffee**:

```
SSM> PATCH /sb/joe.coffee/profile/new-sb-profile {"name": "visitors",
"protection": "medium", "protection-enabled": true, "internet-access":
true, "inspect-stream": true, "safe-search": false, "https-termination":
true, "default": true, "subscriber-safety": true}
{
    "successful": [
        {
            "field": "protection",
            "value": "medium"
        },
        {
            "field": "subscriber-safety",
```

```
                "value": true
            },
            {
                "field": "protection-enabled",
                "value": true
            },
            {
                "field": "internet-access",
                "value": true
            },
            {
                "field": "safe-search",
                "value": false
            },
            {
                "field": "https-termination",
                "value": true
            },
            {
                "field": "name",
                "value": "visitors"
            },
            {
                "field": "inspect-stream",
                "value": true
            },
            {
                "field": "default",
                "value": true
            }
        ],
        "status": "success"
}
```


Retrieve General Information for all Profiles Associated with a Specific SIA SMB Subscriber

Use the following **GET** API to retrieve a subscriber's profile details:

```
SSM> GET /sb/<accountId>/profile
```

NOTE: In the SIA SMB subscriber portal application, profiles are called “device groups.”

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account for which to retrieve profiles.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows configuration details for the specified subscriber:

```
{
    "total": "<integer>",
    "limit": "<integer>",
    "content": [
       {
            "id": "<uuid>",
            "name": "<profileName>",
            "protection": "protectionLevel",
            "protection-enabled": <boolean>,
            "devices": [
                {
                    "name": "<deviceName>",
                    "address": "<deviceAddress>",
                    "profile": "<profileName>",
                   "last-seen": "<date>T<time>Z",
                    "type": "<deviceType>"
                },
                ...
             ],
            "schedules": [
              {
              "categories": [
                  "<contentCategory>",
                  ...
                  ],
              "day": [
                  "<scheduleDay>",
                  ...
              ],
              "enabled": <boolean>,
              "name": "<scheduleName>",
              "timeEnd": "<startTime>",
```

```
                "timeStart": "<endTime>",
                "timeZone": "<zoneId>",
                "type": "<scheduleType>"
                }
                ...
            ],
            "internet-access": <boolean>,
            "inspect-stream": <boolean>,
            "safe-search": <boolean>,
            "http-termination": <boolean>,
            "safe-search-services": {
                "youtube": <boolean>,
                "google": <boolean>,
                "bing": <boolean>
        }
    },
    ...
  ]
}
```

NOTE: The API response includes the “devices” fields only for profiles that are attached to one or more devices. See Chapter 9, “Device APIs,” for details about the APIs that add devices to a profile. The response contains the “schedules” fields only for profiles that have schedules configured. See Chapter 7, “Schedule APIs,” for details about the APIs that add schedules to a profile.

The table that follows describes the possible fields in the API response:

Possible Profile Fields

Field	Description
General Profile Fields	
"name"	Identifies the profile for which configuration details are displayed.
"protection"	Identifies the protection level the profile enforces. **NOTE:** The service provider determines the name of each protection level in an SIA SMB service. For example, common protection level names are strict, medium, and light.
"protection-enabled"	Boolean value indicating whether protection is enabled (true) or disabled (false) at the profile-level.

Field	Description
"devices" Fields **NOTE:** The “devices” fields show only if devices are added to the profile using device-related APIs.	
"name"	Identifies a device that has the profile attached.
"address"	Displays the MAC, MSISDN (Mobile Station International Subscriber Directory Number), or UUID address that is assigned to the device.
"profile"	Identifies the profile that is attached to the specified devices.
"last-seen"	Indicates the last time the specified device accessed the network, where the date and time is expressed in the following format: <*year-month-day*>T<*hour*:*min*:*seconds*><*z-time*>Z
"type"	Identifies the type of address that is associated with the device for which configuration information is displayed. The following address types are possible: ● hardware-address—Indicates a MAC address. ● line—Indicates a mobile line. ● roaming—Indicates an SIA Remote device.
"address-type"	Indicates whether the address assigned to the device is a MAC or MSISDN address.
"device-model"	Identifies the device model.
"device-os"	Identifies the OS that is installed on the device.
"device-user-name"	Displays the user name that is associated with the device.
"secondary-device-name"	Displays the secondary name assigned to the device automatically (if the device is renamed).
"software-version"	Identifies the software version that is installed on the device.
"status"	Displays the current device status.

Field	Description
"schedules" Fields **NOTE:** The response contains the “schedules” fields only for profiles that have schedules configured (which are added to the profile using the schedule-related APIs.)	
"categories"	Displays the content categories the schedule restricts during the specified dates and times. **NOTE:** This field is applicable to content restriction schedules only.
"day"	Identifies the days of the week on which the profile is enforced.
"enabled"	Boolean value indicating whether the schedule is enabled (true) or disabled (false).
"name"	Identifies the schedule the profile is enforcing.
"timeEnd	Indicates the time at which the engines (Cacheserve and the Proxy server) stop enforcing the protection profile; the time is expressed in the following format: <*year-month-day*>T<*hours*> : <*minutes*> : <*seconds*>
"timeStart"	Indicates the time at which the engines (Cacheserve and the Proxy server) start enforcing the protection profile; the time is expressed in the following format: <*year-month-day*>T<*hours*> : <*minutes*> : <*seconds*>
"timeZone"	Displays the time zone for the schedule. The timezone is a text string representing a time zone from the Arthur David Olson tz database. For a list of possible time zone identifiers, see the following URL: *https://en.wikipedia.org/wiki/List_of_tz_database_time_zones*.
"type"	Indicates whether the schedule is a homework (content restriction) schedule or an internet-off schedule.
Additional General Profile Fields	
"age-group"	Identifies the age group that is assigned to the profile.
"internet-access"	Boolean value indicating whether internet access is enabled (true) or disabled (false) for all devices associated with the subscriber account.

Field	Description
"inspect-stream"	Boolean value indicating whether an inspect-stream policy is bound to the profile. A value of true indicates an inspect-stream policy is bound to the profile. A value of false indicates the inspect-stream policy is not bound to the profile. The default value is false.
"safe-search"	Boolean value indicating that one or more safe search services (YouTube, Google, and Bing) are enabled. A value of true indicates at least one safe search service is enabled; a value of false indicates no safe search services are enabled.
"https-termination"	Boolean value indicating whether SSL (HTTPS) termination service evaluation is enabled (true) or disabled (false) on the devices that have the profile attached. When enabled, the Proxy server evaluates device traffic for the SSL (HTTPS) termination service. **Caution!** SSL (HTTPS) termination service evaluation is a limited feature that is not commonly configured; do not enable SSL (HTTPS) termination service evaluation on devices unless Akamai has granted your company explicit contracted permissions to enable that feature.

"safe-search-services"	Boolean value indicating whether the Youtube, Google, and Bing safe search services are enabled (true) or disabled (false) on devices that have the profile attached. Each service has its own boolean value as demonstrated in the following example:
	`"safe-search-services": {`
	`"google": true,`
	`"bing": true,`
	`"youtube": true`
	`},`

"default"	Boolean value indicating whether the profile is the default profile. A value of true indicates the profile is the default profile, while a false value indicates the profile is not the default profile.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows shows sample profile details for a subscriber account called **joe.coffee**. In this example, the subscriber account has three (**3**) associated profiles (**sb_group_visitor**, **sb_group_employee**, and **sb_group_blocked_all_hidden**):

```
SSM> GET /sb/joe.coffee/profile
{
    "total": 3,
    "limit": 8,
    "content": [
        {
            "id": "8a32c413-d48f-48e4-807f-d1b3de0c9eff",
            "name": "sb_group_visitor",
            "protection": "medium",
            "protection-enabled": true,
            "devices": [
                {
                    "name": "laptop-20",
                    "address": "00:16:3b:7a:81:69",
                    "profile": "sb_group_visitor",
                    "last-seen": null,
                    "type": "hardware-address"
                },
                {
                    "name": "tablet-35",
                    "address": "41732345953",
                    "profile": "sb_group_visitor",
                    "last-seen": null,
                    "type": "line"
                },
                {
                    "name": "laptop-45",
                    "address": "67494472656",
                    "profile": "sb_group_visitor",
                    "last-seen": null,
                    "type": "line"
                }
            ],
            "schedules": [
                {
                    "name": "internet-off-weekday",
                    "type": "internet-off",
                    "enabled": true,
                    "day": [
                        "MO",
                        "TU",
                        "WE",
                        "TH",
                        "FR"
                    ],
```

```
                "timeStart": "21:00",
                "timeEnd": "08:00"
            },
            {
                "name": "homework",
                "type": "homework",
                "enabled": true,
                "categories": [
                    "sb-violence",
                    "sb-offensive"
                ],
                "day": [
                    "MO",
                    "TU",
                    "WE",
                    "TH",
                    "FR"
                ],
                "timeStart": "07:00",
                "timeEnd": "08:00"
            }
        ],
        "internet-access": true,
        "inspect-stream": false,
        "safe-search": true,
        "https-termination": false,
        "safe-search-services": {
            "google": true,
            "bing": true,
            "youtube": true
        },
        "default": true
    },
    {
        "id": "a592e0b7-e8c3-4ae9-aa01-7a6a9c2f7dbe",
        "name": "sb_group_employee",
        "protection": "medium",
        "protection-enabled": true,
        "devices": [
            {
                "name": "laptop-30",
                "address": "00:22:3a:7b:16:28",
                "profile": "sb_group_employee",
                "last-seen": null,
                "type": "hardware-address"
            }
        ],
        "schedules": [
            {
                "name": "internet-off-weekday",
                "type": "internet-off",
```

```
                    "enabled": true,
                    "day": [
                        "MO",
                        "TU",
                        "WE",
                        "TH",
                        "FR"
                    ],
                    "timeStart": "21:00",
                    "timeEnd": "08:00"
                }
            ],
            "internet-access": true,
            "inspect-stream": false,
            "safe-search": true,
            "https-termination": false,
            "safe-search-services": {
                "google": true,
                "bing": true,
                "youtube": true
            },
            "inspect-stream": false,
            "default": false
        },
        {
            "id": "ff8523f6-3f6b-499b-a992-cc35e1ccca6d",
            "name": "sb_group_blocked_all_hidden",
            "protection": "none",
            "protection-enabled": true,
            "devices": [
                {
                    "name": "tablet-35",
                    "address": "51642845953",
                    "profile": "sb_group_blocked_all_hidden",
                    "last-seen": null,
                    "type": "line"
                }

            ],
            "schedules": [

            ],
            "internet-access": false,
            "inspect-stream": false,
            "safe-search": false,
            "https-termination": false,
            "safe-search-services": {
                "google": false,
                "bing": false,
                "youtube": false
            },
```

```
            "default": false
        }
    ]
}
```

Retrieve Configuration Details for a Specific Profile

Use the following **GET** API to retrieve the contents of a profile that is attached to a specific subscriber account:

```
SSM> GET /sb/<accountId>/profile/<profileName>
```

NOTE: In the SIA SMB subscriber portal application, profiles are called “device groups.”

Fields

Field	Description
<accountId>	Required. Specifies the subscriber account for which to retrieve information about the devices that have the specified profile attached.
<profileName>	Required. Specifies the name of the profile for which to retrieve device information. **NOTE:** The profile name is case-sensitive.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output displays a list of devices using the specified profile and associated with a specified subscriber account; the response includes the address associated with each device:

```
{
    "id": "<uuid>",
    "name": "<profileName>"
    "protection": "<protectionLevel>",
    "protection-enabled": <boolean>,
    "devices": [
        {
            "address": "<deviceAddress>",
            "address-type": "<addressType>",
            "device-model": "<modelId>",
            "device-os": "<osId>",
            "device-user-name": "<userName>",
            "last-seen": "<date>T<time>Z",
            "name": "<deviceName>",
            "profile": "<profileName>",
            "secondary-device-name": "<secondaryName>",
```

```
            "software-version": "<versionNumber>",
            "status": "<deviceStatus>",
            "type": "<deviceType>"
        }
    ],
    "schedules": [{
        "categories": [
            "<contentCategory>"
            ...
        ],
        "day": [
            "<scheduleDay>"
           ...
        ],
        "enabled": <boolean>,
        "name": "<scheduleName>",
        "timeEnd": "<startTime>",
        "timeStart": "<endTime>",
        "timeZone": "<zoneId>",
        "type": "<scheduleType>"
        }
    ],
    "internet-access": <boolean>,
    "inspect-stream": <boolean>,
    "safe-search": <boolean>,
    "https-termination": <boolean>,
    "safe-search-services": {
        "youtube": <boolean>,
        "google": <boolean>,
        "bing": <boolean>
      },
    "default": <boolean>
}
```

See the "Configurable Profile Fields" section for a description of the fields displayed in the API response body.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows retrieves the contents of a profile called **sb_group_employee** for the subscriber account called **joe.coffee**:

```
SSM> GET /sb/joe.coffee/profile/sb_group_employee
{
   "id": "a592e0b7-e8c3-4ae9-aa01-7a6a9c2f7dbe",
   "name": "sb_group_employee",
   "protection": "medium",
   "protection-enabled": true,
   "devices": [
```

```
    ],
    "schedules": [

    ],
    "internet-access": true,
    "inspect-stream": false,
    "safe-search": true,
    "https-termination": false,
    "safe-search-services": {
        "google": true,
        "bing": true,
        "youtube": true
    },
    "default": false
}
```

Retrieve Information About the Devices Assigned to a Particular Profile

Use the following **GET** API to retrieve information about the devices that are assigned to a specific subscriber profile:

```
SSM> GET /sb/<accountId>/profile/<profileName>/device
```

NOTE: In the SIA SMB subscriber portal application, profiles are called “device groups.”

Fields

Field	Description
<accountId>	Required. Specifies the name of the subscriber account for which to retrieve information about the devices that have the specified profile attached.
<profieName>	Required. Specifies the name of the profile for which to retrieve device information. **NOTE:** The profile name is case-sensitive.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output displays a list of devices using the specified profile and associated with a specified subscriber account; the response includes the address associated with each device:

```
{
    "total": <integer>,
    "limit": <integer>,
    "content": [
        {
            "name": "<deviceName>",
            "address": "<deviceAddress>"
            "address-type": "<addressType>",
            "device-model": "<modelId>",
            "device-os": "<osId>",
            "device-user-name": "<userName>",
            "profile": "<profileName>",
            "secondary-device-name": "<secondaryName>",
            "software-version": "<versionNumber>",
            "status": "<deviceStatus>",
            "last-seen": "<date>T<time>Z",
            "type": "<deviceType>"
        },
        ...
    ]
}
```

The table that follows describes the API output fields:

Field	Description
"total"	Integer indicating the number of devices that have the specified profile attached.
"limit"	Integer indicating the maximum number of devices displayed in the output.
"name"	Text string that identifies a device that has the specified profile attached.
"address"	Displays the MAC, MSISDN (Mobile Station International Subscriber Directory Number), or UUID address assigned to the specified device.
"address-type"	Identifies the type of address that is associated with the device.

Field	Description
“device-model”	Identifies the model type for device instance. **NOTE:** The response includes this field only for roaming devices that have the SIA Remote service enabled. This field is for information purposes.
“device-os”	Identifies the OS installed on the device. **NOTE:** The response includes this field only for roaming devices that have the SIA Remote service enabled. This field is for information purposes.
“device-user-name”	Identifies the user name associated with the device. **NOTE:** The response includes this field only for roaming devices that have the SIA Remote service enabled. This field is for information purposes.
"profile"	Identifies the profile for which information is displayed; this is the profile attached to the specified device.
“secondary-device--name”	Displays the original name assigned to a device that is renamed. **NOTE:** The response includes this field only for devices that have been renamed.
“software--version”	Identifies the software version that is installed on the device instance. **NOTE:** The response includes this field only for roaming devices that have the SIA Remote service enabled. This field is for information purposes.
“status”	Indicates whether the device instance is currently enabled or disabled. **NOTE:** The response includes this field only for roaming devices that have the SIA Remote service enabled. This field is for information purposes.
"last-seen"	Indicates the last time the specified device accessed the network, where the date and time is expressed in the following format: <*year-month-day*>**T**<*hour*:*min*:*seconds*><*z-time*>**Z**

Field	Description
"type"	Identifies the type of address that is associated with the device for which configuration information is displayed. The following address types are possible: • hardware-address—Indicates a MAC address. • line—Indicates a mobile line. • roaming—Indicates an SIA Remote device. **NOTE:** For the SPS Remote service, the "type" is always "roaming".

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows retrieves a list of devices associated with a subscriber account called **joe.coffee** and using a profile called **sb_group_employee**:

```
SSM> GET /sb/joe.coffee/profile/sb_group_employee/device
{
    "total": 4,
    "limit": 4,
    "content": [
        {
            "name": "tablet-52",
            "address": "00:16:89:73:02:c9",
            "profile": "sb_group_employee",
            "last-seen": "2024-05-21T06:18:34Z",
            "type": "hardware-address"
        },
        {
            "name": "Mac Book Pro-93",
            "address": "00:16:d0:64:bd:65",
            "profile": "sb_group_employee",
            "last-seen": "2024-05-21T06:33:34Z",
            "type": "hardware-address"
        },
        {
            "name": "tablet-13",
            "address": "74049349999",
            "profile": "sb_group_employee",
            "last-seen": "2024-05-21T06:33:34Z",
            "type": "line"
        },
        {
            "name": "iphone-os13",
            "address": "8005552345"
            "profile": "sb_group_employee",
            "address-type": "roaming",
```

```
            "last-seen": "2024-06-11T17:30:01Z",
            "secondary-device-name": "jane doe - 1718127001",
            "device-model": "Phone9,3",
            "device-os": "iOS Version 13.5.1 (Build 17F80)",
            "device-user-name": "jane doe",
            "software-version": "ios-1.0.7",
            "status": "enabled",
            "type": "roaming"
        }

    ]
}
```


Retrieve the Content Categories Included in a Subscriber Profile

Use the following **GET** API to retrieve a list of the content categories included in a specific profile:

```
SSM> GET /sb/<accountId>/profile/<profileName>/protection
```

NOTE: In the SIA SMB subscriber portal application, profiles are called “device groups.”

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account for which to retrieve a profile's content categories.
<profileName>	Required. String that identifies the profile for which to retrieve included content categories. **NOTE:** The profile name is case-sensitive.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows configuration details for the specified subscriber profile's content categories:

```
SSM> GET /sb/<accountId>/profile/<profileName>/protection
{
    "total": <integer>,
    "limit": <integer>,
    "content": [
        {
            "name": "<contentCategory>",
            "description": "<textString>"
        },
        …
    ]
}
```

The table that follows describes the fields displayed in the response body:

Field	Description
"total"	Integer indicating the number of nodes (domains) in the black list.
"limit"	Integer indicating the maximum number of domains that can be displayed in the response body.
"content"	Includes the following information about the content categories included in the specified protection profile: • "name"—Identifies the content categories included in the specified profile. • "description"—Text string describing the content category.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows shows sample output for the **sb_group_visitor** protection profile for the subscriber called **joe.coffee**:

```
SSM> GET /sb/joe.coffee/profile/sb_group_visitor/protection
{
    "total": 28,
    "limit": 28,
    "content": [
        {
            "name": "sb-crime",
            "description": "Criminal Activity"
        },
        {
           "name": "sb-self-harm",
           "description": "Self-Harm"
        },
        {
           "name": "sb-child-abuse",
           "description": "Child Abuse Images"
        },
        {
           "name": "sb-porn",
           "description": "Adult content"
        },
        {
           "name": "sb-cults",
           "description": "Cults"
        },
        {
           "name": "sb-nudity",
           "description": "Nudity"
        },
        {
           "name": "sb-lingerie",
           "description": "Lingerie & Swimsuits"
        },
        {
           "name": "sb-sex-education",
           "description": "Sex Education"
        },
        {
           "name": "sb-offensive",
           "description": "Tasteless"
        },
        {
           "name": "sb-hacking",
           "description": "Sites that promote hacking and other
illegal online activities"
        },
```

```
        {
            "name": "sb-cheating",
            "description": "School Cheating"
        },
        {
            "name": "sb-anonymizers",
            "description": "Anonymizer"
        },
        {
            "name": "sb-translate",
            "description": "Translators"
        },
        {
            "name": "sb-dating",
            "description": "Dating sites"
        },
        {
            "name": "sb-drugs",
            "description": "Illegal drugs"
        },
        {
            "name": "sb-alc-tob",
            "description": "Alcohol and Tobacco"
        },
        {
            "name": "sb-software",
            "description": "Download Sites"
        },
        {
            "name": "sb-p2p",
            "description": "Peer-to-Peer"
        },
        {
            "name": "sb-warez",
            "description": "Illegal software"
        },
        {
            "name": "sb-gambling",
            "description": "Gambling sites"
        },
        {
            "name": "sb-shopping",
            "description": "Online shopping and auction sites"
        },
        {
            "name": "sb-weapons",
            "description": "Weapons"
        },
        {
            "name": "sb-violence",
            "description": "Violence"
```

```
            },
        ]
}
```

Remove a Profile from a Subscriber Account

Use the following **DELETE** API to remove a profile from the specified subscriber account; subscriber devices which had that profile applied return to using the default profile:

```
SSM> DELETE /sb/account/<accountId>/profile/<profileName>
```

NOTE: In the SIA SMB subscriber portal application, profiles are called “device groups.”

Be aware that you cannot remove the default profile from a subscriber account. By default, all devices have the default profile applied until you assign a non-default profile to the device. When you delete a non-default profile from a subscriber account, the SSM performs the following tasks:

- Removes that non-default profile from any devices which have that non-default profile attached.
- Reapplies the default profile to those devices.

Fields

Field	Description
<accountId>	Required. Specifies the subscriber account from which to remove the specified profile. Replace the ***<accountId>*** argument with the desired account identifier.
<profileName>	Required. Specifies the profile to remove from the specified subscriber account (along with any devices using that profile). Replace the ***<profileName>*** argument with the name of the profile you want to remove. **Caution!** The profile name is case-sensitive. **NOTE:** You cannot delete the default profile.

Expected Response

The SSM removes the specified profile from the specified subscriber account as requested. In addition, the SSM applies the default profile to any subscriber account devices that were using the deleted profile.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows removes a profile called **old_sb_profile** from a subscriber account called **joe.coffee**; after deleting the profile, the SSM applies the default profile to the affected devices:

```
SSM> DELETE /sb/account/joe.coffee/profile/old_sb_profile
```

Chapter 6: Content Category APIs

This chapter describes the APIs for performing the following tasks:

Retrieve the Content Categories that Contain a Specific URL

Use the following **POST** API to retrieve the content categories that contains a particular URL:

```
SSM> POST /service/sb/protection
{
  "node": "<url>"
}
```

Fields

Field	Description
<url>	Required. Identifies a URL for which to display the associated content categories. Replace the ***<url>*** argument with the desired URL in the format ***<textString>*.com.** For example, **sample-url.com**.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows the content categories that contain the specified URL:

```
{
  "total": <integer>,
  "limit": <integer>,
  "content": [
        {
            "name": "<contentCategory>",
            "description": "<textString>"
```

```
        },
    ]
    ...
}
```

The table that follows describes the fields in the response body:

Field	Description
"total"	Integer indicating the number of SIA SMB content categories that contain the specified URL. **NOTE:** See the "Limiting Retrieved Results for GET Requests" section in Chapter 1, "Introduction to the SSM Restful API," for information about how to set the limit field for **GET** requests.
"limit"	Integer indicating the maximum number of results (content categories) the API response body includes. **NOTE:** See the "Limiting Retrieved Results for GET Requests" section in Chapter 1, "Introduction to the SSM Restful API," for information about how to set the limit field for **GET** requests.
"content"	Includes the following information about the content categories that contain the specified URL: • "name"—Identifies the content category • "description"—Text string describing the content category.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows retrieves the content categories associated with the SIA SMB service and containing the URL **sample-tasteless-url.com**:

```
SSM> POST /service/sb/protection {"node":"sample-tasteless-url.com"}
{
  "total": 1,
  "limit": 1,
  "content": [
        {
            "name": "sb-offensive",
            "description": "Tasteless"
        }
    ]
}
```

Retrieve Content Category Mapping Information for Web Content Filtering Lists

Use the following **POST** API to retrieve SIA SMB content category mapping information for one or more web content filtering lists:

```
SSM> POST /service/sb/category/search
{
    "terms": [
        "<listName>"
        ]
}
```

NOTE: The information the API retrieves is valuable for reporting on blocks-by-category.

Fields

Field	Description
<listName>	Required. Identifies a web content filtering list; the API returns content category mapping information for the specified list. Replace the ***<listName>*** argument with the desired list name. To return content category mapping information for multiple lists, separate each list name with a comma as shown in the following example: **"Terms": ["*<listName-1>*", "*<listName-2>*", "*<listName-3>*"]**

Expected Response

A successful response includes HTTP code 200 and a JSON based response body that shows content category mapping information for the specified web content filtering lists. The example that follows shows what the response body looks like:

```
{
    "total": <integer>,
    "limit": <integer>,
    "content": [
        {
            "term": "<listName>",
            "item": {
                "name": "<contentCategoryName>",
                "description": "<textString>"
            }

        },
```

```
        ....
    ]
}
```

The table that follows describes the fields in the API response:

Field	Description
"total"	Integer indicating the number SIA SMB content categories that are mapped to the specified web content filtering lists.
"limit"	Integer indicating the maximum number of web content filtering lists the API can display in the response. **NOTE:** See the "Limiting Retrieved Results for GET Requests" section in Chapter 1, "Introduction to the SSM Restful API," for information about how to set the limit field for **GET** requests.
"content"	Includes the following information for each web content filtering list: • "name"—Identifies the content category that is mapped to the specified list. • "description"—Text string that provides additional information about the list and content category.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows searches for content categories that are mapped to the web content filtering lists called **offensive-i** and **weapons_i**:

```
SSM> POST /service/sb/category/search {"terms": ["offensive_i",
"weapons_i"]}
{
    "total": 2,
    "limit": 2,
    "content": [
        {
            "term": "offensive_i",
            "item": {
                "name": "sb-offensive",
                "description": "Tasteless"
            }
        },
        {
            "term": "weapons_i",
            "item": {
                "name": "sb-weapons",
                "description": "Weapons"
            }
```

```
            }
        ]
}
```

Update the Custom Filters for a Subscriber Device Group

Use the following **POST** API to add one or more content categories to a specific SIA SMB subscriber profile (where a profile is known as a “device group” in the SIA SMB subscriber portal application):

```
SSM> POST /sb/<accountId>/profile/<profileName>/protection
[
    "<categoryName>",
    …
]
```

Fields

Field	Description
<accountId>	Required. Identifies the subscriber account that is associated with the subscriber profile to which you want to add a content category. Replace the ***<accountId>*** argument with the name of the desired subscriber account.
<profileName>	Required. Identifies the profile (device group) to which you want to add content categories. Replace the ***<profileName>*** argument with the name of the desired profile. **NOTE:** The profile name is case-sensitive. **NOTE:** In the SIA SMB subscriber portal application, profiles are called “device groups.”
<categoryName>	Required. Identifies a content category to add to the specified profile. Replace the ***<categoryName>*** argument with the name of the desired content category. To add multiple content categories to a profile, separate each category name with a comma as demonstrated in the following example: `["sb-sports", "sb-violence", "sb-politics"]`

Expected Response

The SSM adds the specified content categories to the profile as requested.

You can use the following **GET** API to verify the category list is updated and accurate; see the "Retrieve a List of Custom Content Categories a Profile Blocks" section for details:

```
SSM> GET /sb/<accountId>/profile/<deviceGroup>/protection
```

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows adds the **sb-sports** filter to an SIA SMB subscriber (called **joe.coffee**) profile called **sb_group_employee**:

```
SSM> POST /sb/joe.coffee/profile/sb_group_employee/protection ["sb-sports"]
```

Retrieve the SIA SMB Content Categories that are Blocked in Each Protection Level

Use the following **GET** API to retrieve the SIA SMB content categories that are blocked in each protection level:

```
SSM> GET /service/sb/protection-level
```

Fields

None.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows all content categories that are blocked in each protection level a subscriber account supports:

```
{
    "content": [
        {
            "name": "<protectionLevel>",
            "categories": [
                {
                    "name": "<categoryName>",
```

```
                    "description": "<textString>"
                },
                 ...
            ]
        },
        ...
    ]
}
```

The table that follows describes the fields in the response body:

Field	Description
"name": "<*protectionLevel*>"	Identifies the protection level for which blocked content categories are displayed.
"categories"	Includes the following information for each blocked content category in the specified protection level: • "name"—Identifies the content category. • "description"—Text string describing the content category.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows retrieves the SIA SMB content categories that are blocked in each protection level a subscriber account supports. Be aware that this is a simplified example that demonstrates what the API response looks like; an SIA SMB service can have many more blocked content categories. In this example, the SIA SMB service supports five protection levels (called **light**, **medium**, **strict**, **custom**, and **none**):

```
SSM> GET /service/sb/protection-level
{
    "content": [
        {
            "name": "light",
            "categories": [
                {
                    "name": "sb-test-webfilter",
                    "description": "Test Category for Web Filter"
                },
```

```
                {
                    "name": "sb-crime",
                    "description": "Criminal Activity"
                }
            ]
        },
        {
            "name": "medium",
            "categories": [
                {
                    "name": "sb-cults",
                    "description": "Cults"
                },
                {
                    "name": "sb-lingerie",
                    "description": "Lingerie & Swimsuits"
                },
                {
                    "name": "sb-sex-education",
                    "description": "Sex Education"
                },
                {
                    "name": "sb-offensive",
                    "description": "Tasteless"
                }
            ]
        },
        {
            "name": "strict",
            "categories": [
                {
                    "name": "sb-self-harm",
                    "description": "Self-Harm"
                },
                {
                    "name": "sb-child-abuse",
                    "description": "Child Abuse Images"
                },
                {
                    "name": "sb-porn",
```

```
                    "description": "Adult content"
                },
                {
                    "name": "sb-chat",
                    "description": "Chat"
                }
            ]
        },
        {
            "name": "custom",
            "categories": [

            ]
        },
        {
            "name": "none",
            "categories": [

            ]
        }
    ]
}
```

Retrieve the Blocked Content Categories in a Protection Level

Use the following **GET** API to retrieve the SIA SMB content categories a specific protection level blocks:

```
SSM> GET /service/sb/protection/<protectionLevel>
```

Fields

Field	Description
<protectionLevel>	Required. Identifies the protection level for which to display blocked content categories. Replace the ***<protectionLevel>*** argument with the name of the protection level for which to display content category information.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output lists the SIA SMB content categories the specified protection level blocks:

```
{
   "total": <integer>,
   "limit": <integer>,
   "content": [
       {
           "name": "<contentCategory>",
           "description": "<textString>"
       },
       ...
   ]
}
```

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows retrieves a list of blocked content categories in a protection level called “**light**”:

```
SSM> GET /service/sb/protection/light
{
   "total": 4,
   "limit": 4,
   "content": [
       {
           "name": "sb-test-webfilter",
```

```
            "description": "Test Category for Web Filter"
        },
        {
            "name": "sb-crime",
            "description": "Criminal Activity"
        },
        {
            "name": "sb-gambling",
            "description": "Gambling sites"
        },
        {
            "name": "sb-weapons",
            "description": "Weapons"
        }
    ]
}
```

Retrieve a List of Custom Content Categories a Profile Blocks

Use the following **GET** API to retrieve a list of the custom content categories a particular profile blocks:

```
SSM> GET /sb/<accountId>/profile/<profileName>/protection
```

Fields

Field	Description
<accountId>	Required. Identifies the subscriber account that owns the profile for which to retrieve the content category list. Replace the ***<accountId>*** argument with the name of the desired SIA SMB subscriber account.
<profileName>	Required. Identifies the profile for which to display the content category list. Replace the ***<profileName>*** argument with the name of the desired device group. **NOTE:** The profile name is case-sensitive.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows all custom content categories a profile blocks:

```
{
    "total": <integer>,
    "limit":  <integer>,
    "content": [
        {
            "name": "<categoryName>",
            "description": "<textString>"
        }
    ]
}
```

The table that follows describes the fields displayed in the response body:

Field	Description
"total"	Integer indicating the number of custom content categories that are currently associated with the specified profile and protection level.
"limit"	Integer indicating the maximum number of custom content categories the API response can include. **NOTE:** See the "Limiting Retrieved Results for GET Requests" section in Chapter 1, "Introduction to the SSM Restful API," for information about how to set the limit field for **GET** requests.
"content"	Includes the following information about each custom content category: • "name"—Identifies a content category in the specified protection level. • "description"—Text string describing the content category.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows shows the blocked custom content categories for an SIA SMB subscriber account (called **joe.coffee**) and device group (called **sb_group_employee**); the **sb_group_employee** device group blocks three custom content categories (called **sb-politics**, **sb-violence**, and **sb-sports**):

```
SSM>  GET /sb/joe.coffee/profile/sb_group_employee/protection
{
    "total": 3,
    "limit": 3,
    "content": [
        {
            "name": "sb-politics",
            "description": "Politics sites"
        },
        {
            "name": "sb-violence",
            "description": "Violence"
        },
        {
            "name": "sb-sports",
            "description": "Sports"
        }
    ]
}
```

Chapter 7: Schedule APIs

This chapter describes the APIs you use to create, update and retrieve configuration information about Content Restriction and Internet Off schedules.

The SIA SMB service profiles can enforce the following types of schedules:

- Content restriction schedules that prevent devices from accessing websites that fall under the specified content categories during a specified timeframe. For example, you can create a content restriction schedule that blocks social media websites during working hours (from 8:00 AM through 5:00 PM) only.
- “Internet Off” schedules that prevent devices from accessing the Internet during the specified timeframe.

You add content restriction and Internet Off schedules to an individual profile’s configuration. After configuring a schedule in a profile, the SIA SMB service enforces that schedule on all devices attached to the profile.

This chapter describes the APIs you use to create, update, and retrieve Content Restriction and Internet Off schedules; use these APIs to perform the following tasks:

- Add Internet-off or Content Restriction schedules to an existing profile
- Update Internet-off or Content Restriction schedules for an existing profile
- Retrieve a list of schedules enforced by a specific SIA SMB subscriber profile
- Retrieve configuration details for a specific Internet Off or Content Restriction schedule
- Remove a Content Restriction or Internet Off schedule from an SIA SMB profile

The APIs described in this chapter are specific to Content Restriction and Internet Off schedules.

Add Internet-off or Content Restriction Schedules to an Existing Profile

Use the following **POST** API to add Internet Off or Content Restriction schedules to an existing profile:

```
SSM> POST /sb/<accountId>/profile/<profileName>/schedule
{
  "name": "<scheduleName>",
  "type": "<scheduleType>",
  "enabled": <boolean>,
  "categories": [
     "<contentCategory>",
     ...
  ],
```

```
 "day": [
   "<scheduleDay>",
   ...
 ],
 "timeStart": "<startTime>",
 "timeEnd": "<endTime>",
 "timeZone": "<zoneId>"
}
```

Fields

Field	Description
<*accountId*>	Required. String that identifies the subscriber account for which to configure a schedule. Replace the **<*accountId*>** argument with the appropriate subscriber account name.
<*profileName*>	Required. Text string name that identifies the profile containing the devices on which to enforce the schedule. Replace the **<*profileName*>** argument with the appropriate profile name. **NOTE:** The profile name is case-sensitive.
"name"	Text string name that identifies the schedule being enforced by the profile.
"type"	Identifies the schedule type.
"enabled"	Boolean value that indicates whether the schedule is enabled (**true**) or disabled (**false**).
"categories"	Identifies the content categories that are restricted during the scheduled time period. **NOTE:** The "categories" field applies to Content Restriction schedules only.
"day"	Identifies the days of the week on which to enforce the schedule.
"timeStart"	Indicates the time at which the engines (Cacheserve and the Proxy server) starts enforcing the schedule; the time is expressed in the following format: <*year-month-day*>**T**<*hours*> **:** <*minutes*> **:** <*seconds*>
"timeEnd"	Indicates the time at which the engines (Cacheserve and the Proxy server) stops enforcing the schedule; the time is expressed in the following format: <*year-month-day*>**T**<*hours*> **:** <*minutes*> **:** <*seconds*>

Field	Description
"timeZone"	Identifies the time zone for the schedule where the timezone is a text string representing a time zone from the Arthur David Olson tz database. For a list of possible time zone identifiers, see the following URL: *https://en.wikipedia.org/wiki/List_of_tz_database_time_zones*

Expected Response

A successful response includes HTTP code 201 without a response body.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows demonstrates how to create and configure schedules for a given profile. This example uses a profile, called **visitors**, for the subscriber account called **joe.coffee**. The schedulers enforces the following protections on devices that are attached to the profile:

- An Internet-off schedule that disables internet access from 9:00 PM through 8:00 AM on weekdays (Monday through Friday).
- A content restriction schedule that disables access to websites containing offensive or violent content during the specified timeframe (which is weekdays between 7:00 AM and 8:00 AM).

```
SSM> POST /sb/joe.coffee/profile/visitors/schedule {"name":
"internet-off-weekday", "type": "internet-off", "enabled": true,
"day":["MO", "TU", "WE", "TH", "FR"], "timeStart": "21:00",
"timeEnd": "08:00"}

SSM> POST /sb/joe.coffee/profile/visitors/schedule {"name":
"homework", "type": "homework", "enabled": true, "categories":
["sb-violence", "sb-offensive"], "day":["MO", "TU", "WE", "TH",
"FR"], "timeStart": "07:00", "timeEnd": "08:00"}
```

Update Internet-off or Content Restriction Schedules for an Existing Profile

Use the following **PUT** API to update Internet Off or Content Restriction schedules for an existing profile:

```
SSM> PUT /sb/<accountId>/profile/<profileName>/schedule/<scheduleName>
{
  "name": "<scheduleName>",
  "type": "<scheduleType>"
  "enabled": <boolean>,
  "categories": [
     "<contentCategory>",
     ...
  ],
 "day": [
   "<scheduleDay>",
   ...
 ],
 "timeStart": "<startTime>",
 "timeEnd": "<endTime>",
 "timeZone": "<zoneId>"
}
```

Fields

Field	Description
<*accountId*>	Required. String that identifies the subscriber account associated with the schedule you want to retrieve. Replace the **<*accountId*>** argument with the appropriate subscriber account name.
<*profileName*>	Required. Text string name that identifies the profile associated with the schedule you want to retrieve. Replace the **<*profileName*>** argument with the appropriate profile name. **NOTE:** The profile name is case-sensitive.
"name"	Text string name that identifies the schedule being enforced by the profile.
"type"	Identifies the schedule type.
"enabled"	Boolean value that indicates whether the schedule is enabled (**true**) or disabled (**false**).

Field	Description
"categories"	Lists the content categories that are restricted during the scheduled time period. **NOTE:** The "categories" field applies to Content Restriction schedules only.
"day"	Displays the days of the week on which the profile is enforced.
"timeStart"	Displays the time at which the engines (Cacheserve and the Proxy server) starts enforcing the protection profile; the time is expressed in the following format: <*year-month-day*>**T**<*hours*> **:** <*minutes*> **:** <*seconds*>
"timeEnd"	Displays the time at which the engines (Cacheserve and the Proxy server) stops enforcing the protection profile; the time is expressed in the following format: <*year-month-day*>**T**<*hours*> **:** <*minutes*> **:** <*seconds*>
"timeZone"	Displays the time zone for the schedule where the timezone is a text string representing a time zone from the Arthur David Olson tz database. For a list of possible time zone identifiers, see the following URL: *https://en.wikipedia.org/wiki/List_of_tz_database_time_zones*

Expected Response

A successful response includes HTTP code 200 without a response body.

Examples

The examples that follow demonstrate how to update Internet-off and content restriction schedulers for a given profile.

Disable an Active Internet-off Schedule

The example that follows disables an active Internet-off schedule (called **Internet-off-weekday**) in a profile (called **visitors**) that belongs to a subscriber account called **joe.coffee.** The updated field is highlighted in the API call:

```
SSM> PUT /sb/joe.coffee/profile/visitors/schedule/internet-off-weekday
{"name":"internet-off-weekday", "type": "internet-off", "enabled": false,
"day":["MO", "TU", "WE", "TH", "FR"], "timeStart": "21:00", "timeEnd":
"08:00"}
```

Add Categories to a Content Restriction (**homework**) Schedule

The example that follows adds additional categories (**sb-shopping** and **sb-weapons**) to an existing content restriction (**homework**) schedule in a profile (called **visitors**) that belongs to a subscriber account called **joe.coffee**. The new categories are highlighted in the API call:

```
SSM> PUT /sb/joe.coffee/profile/visitors/schedule/homework {"name":
"homework", "type": "homework", "enabled": true, "categories":
["sb-violence", "sb-offensive", "sb-shopping", "sb-weapons"],
"day":["MO", "TU", "WE", "TH", "FR"], "timeStart": "07:00",
"timeEnd": "08:00"}
```

Retrieve a List of Schedules Enforced by a Specific SIA SMB Subscriber Profile

Use the following **GET** API to retrieve a list of all Content Restriction and Internet Off schedules enforced by a specified SIA SMB subscriber profile:

```
SSM> GET /sb/<accountId>/profile/<profileName>/schedule
```

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account associated with the schedule you want to retrieve. Replace the ***<accountId>*** argument with the appropriate subscriber account name.
<profileName>	Required. Text string name that identifies the profile associated with the schedule you want to retrieve. Replace the ***<profileName>*** argument with the appropriate profile name. **NOTE:** The profile name is case-sensitive.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body that shows configuration details for all schedules a profile enforces, as shown in the example that follows:

```
{
    "total": <integer>,
    "limit": <integer>,
    "content": [
        {
            "name": "<scheduleName>",
            "type": "<scheduleType>",
            "enabled": "<boolean>",
```

```
            "categories": [
                "<categoryName>",
                ...
            ],
            "day": [
                "<scheduleDay>",
                ...
            ],
            "timeZone": "<zoneId>",
            "timeStart": "<startTime>",
            "timeEnd": "<endTime>"
        }
    ]
}
```

The table that follows describes the fields displayed in the response body:

Field	Description
“total”	Indicates the number of schedules the response includes.
“limit”	Indicates the maximum number of schedules the profiles can enforce.
"name"	Identifies the schedule you retrieved.
"type"	Identifies the type of schedule you retrieved.
"enabled"	Indicates whether the engines (Cacheserve and the Proxy server) are enforcing the schedule (the <*boolean*> argument is set to **true**) or the schedule is currently disabled (the <*boolean*> argument is set to **false**).
"categories"	Identifies one or more content categories to block during the specified times. If the profile blocks multiple content categories, a comma separates each category as shown in the following example: `"categories": ["pi-shopping","pi-entertainment", "pi-videohosting"]`

Field	Description
"day"	Identifies the days of the week on which the engines (Cacheserve and the Proxy server) enforce the profile. The <*scheduleDay*> argument is expressed by one the following keywords: • **"SU"**—Sunday. • **"MO"**—Monday. • **"TU"**—Tuesday. • **"WE"**—Wednesday. • **"TH"**—Thursday. • **"FR"**—Friday. • **"SA"**—Saturday. If the engines (Cacheserve and the Proxy server) need to enforce a schedule on multiple days, a comma separates each day's keyword as shown in the following example: `"day":["MO","WE","FR"]`
"timeZone"	Optional. Identifies the time zone for the schedule. The <*zoneId*> argument is a time zone from the Arthur David Olson tz database. Keep the following in mind when looking at the time zone for a schedule: • For recurring schedules, the <*zoneId*> argument is the subscriber's time zone as specified for the service. • For non-recurring, the <*zoneId*> argument is the subscriber-specified time zone for the schedule.
"timeStart"	Indicates the time of day at which the engines (Cacheserve and the Proxy server) invoke the schedule and start blocking websites belonging to the specified categories. The <*startTime*> argument is expressed in the following format: <*hour*>:<*minutes*>
"timeEnd"	Indicates the time of day at which the engines (Cacheserve and the Proxy server) stop enforcing the schedule. The <*endTime*> argument is expressed in the following format: <*hour*>:<*minutes*>

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows retrieves configuration details for a profile (called **sb_group_employee**) that enforces one schedule (an "Internet Off" schedule called **in-office-weekday**). A subscriber account called **joe.coffee** owns the profile:

```
SSM> GET /sb/joe.coffee/profile/sb_group_employee/schedule
{
    "total": 1,
    "limit": 14,
    "content": [
```

```
        {
            "name": "in-office-weekday",
            "type": "homework",
            "enabled": true,
            "categories": [
                "sb-nudity",
                "sb-offensive",
                "sb-violence"
            ],
            "day": [
                "MO",
                "TU",
                "WE",
                "TH",
                "FR"
            ],
            "timeZone": "America/Cancun",
            "timeStart": "07:00",
            "timeEnd": "18:00"
        }
    ]
}
```

The example that follows shows scheduling information for a profile (called **gallery-visitors**) that enforces multiple schedules (two “Internet Off” schedules and one “Content Restriction” schedule; a subscriber account caller **art.gallery** owns the profile:

```
SSM> GET /sb/art.gallery/profile/gallery-visitors/schedule
{
    "total": 3,
    "limit": 14,
    "content": [
        {
            "name": "internet-off1",
            "type": "internet-off",
            "enabled": true,
            "day": [
                "MO",
                "TU",
                "WE",
                "TH",
                "FR"
            ],
            "timeStart": "07:00",
            "timeEnd": "9:01"
        },
        {
            "name": "internet-off0",
            "type": "internet-off",
            "enabled": true,
            "day": [
```

```
                "MO",
                "WE"
             ],
            "timeStart": "17:01",
            "timeEnd": "19:00"
        }
    ]
}
```

Retrieve Configuration Details for a Specific Internet Off or Content Restriction Schedule

Use the following **GET** API to retrieve configuration details for a particular profile schedule:

```
SSM> GET /sb/<accountId>/profile/<profileName>/schedule/<scheduleName>
```

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account associated with the schedule you want to retrieve. Replace the ***<accountId>*** argument with the appropriate subscriber account name.
<profileName>	Required. Text string name that identifies the profile associated with the schedule you want to retrieve. Replace the ***<profileName>*** argument with the appropriate profile name. **NOTE:** The profile name is case-sensitive.
<scheduleName>	Required. Text string name that identifies the schedule you want to retrieve. Replace the ***<scheduleName>*** argument with the appropriate schedule name. **NOTE:** Schedule names are not case-sensitive.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body that shows configuration details for the requested schedule, as shown in the example that follows:

```
{
   "name": "<scheduleName>",
   "type": "<scheduleType>",
   "enabled": <boolean>,
   "categories": [
       "<categoryName>",
       ...
```

```
    ],
    "day": [
        "<scheduleDay>",
        ...
    ],
    "timeZone": "<zoneId>",
    "timeStart": "<startTime>",
    "timeEnd": "<endTime>"
}
```

The table that follows describes the fields displayed in the response body:

Field	Description
"name"	Identifies the schedule you retrieved.
"type"	Identifies the type of schedule you retrieved.
"enabled"	Indicates whether the engines (Cacheserve and the Proxy server) are enforcing the schedule (the <*boolean*> argument is set to **true**) or the schedule is currently disabled (the <*boolean*> argument is set to **false**).
"categories"	Identifies one or more content categories to block during the specified times. If the profile blocks multiple content categories, a comma separates each category as shown in the following example: `"categories": ["pi-shopping","pi-entertainment", "pi-videohosting"]`
"day"	Identifies the days of the week on which the engines (Cacheserve and the Proxy server) enforce the profile. The <*scheduleDay*> argument is expressed by one the following keywords: • "SU"—Sunday. • "MO"—Monday. • "TU"—Tuesday. • "WE"—Wednesday. • "TH"—Thursday. • "FR"—Friday. • "SA"—Saturday. If the engines (Cacheserve and the Proxy server) enforce a schedule on multiple days, a comma separates each day's keyword as shown in the following example: `"day":["MO","WE","FR"]`

Field	Description
"timeZone"	Optional. Identifies the time zone for the schedule. The <*zoneId*> argument is a time zone from the Arthur David Olson tz database. Keep the following in mind when looking at the time zone for a schedule: • For recurring schedules, the <*zoneId*> argument is the subscriber's time zone as specified for the service. • For non-recurring, the <*zoneId*> argument is the subscriber-specified time zone for the schedule.
"timeStart"	Indicates the time of day at which the engines (Cacheserve and the Proxy server) invoke the schedule and starts blocking websites belonging to the specified categories. The <*startTime*> argument is expressed in the following format: <*hour*>:<*minutes*>
"timeEnd"	Indicates the time of day at which the engines (Cacheserve and the Proxy server) stop enforcing the schedule. The <*endTime*> argument is expressed in the following format: <*hour*>:<*minutes*>

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows retrieves configuration details for a schedule called **in-office-weekday** which is associated with the **sb_group_employee** profile and a subscriber account called **joe.coffee**:

```
SSM> GET /sb/joe.coffee/profile/sb_group_employee/schedule/in-office-weekday
{
   "name": "in-office-weekday",
   "type": "homework",
   "enabled": true,
   "categories": [
       "sb-nudity",
       "sb-offensive",
       "sb-violence"
   ],
   "day": [
       "MO",
       "TU",
       "WE",
       "TH",
       "FR"
   ],
   "timeZone": "America/Cancun",
   "timeStart": "07:00",
   "timeEnd": "18:00"
}
```

Remove a Content Restriction or Internet Off Schedule from an SIA SMB Profile

Use the following **DELETE** API to remove the specified schedule from an SIA SMB profile:

```
SSM> DELETE /sb/<accountId>/profile/<profileName>/schedule/<scheduleName>
```

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account from which to delete the desired schedule. Replace the ***<accountId>*** argument with the appropriate subscriber account name.
<profileName>	Required. Text string name that identifies the profile from which to delete the specified schedule. Replace the ***<profileName>*** argument with the appropriate profile name. **NOTE:** The profile name is case-sensitive.
<scheduleName>	Required. Text string name that identifies the schedule you want to delete. Replace the ***<scheduleName>*** argument with the appropriate schedule name. **NOTE:** Schedule names are not case-sensitive.

Expected Response

The SSM deletes the specified schedule as requested.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows removes a schedule called **office-safe** from the profile called **sb_group_visitor** and a subscriber account called **joe.coffee**:

```
SSM> DELETE /sb/joe.coffee/profile/sb_group_visitor/schedule/office-safe
```

Chapter 8: Block Page APIs

This chapter describes the APIs you can use to configure and manage the business-sec-template and business-web-template resources, which control the look and feel of the following SIA SMB block pages:

- The malware and phishing block page—Devices using the SIA SMB service encounter the Malware & Phishing block page when trying to access a website that contains malware or is a phishing website. In the SIA SMB subscriber portal application. The APIs in this chapter describe how to view and manage the business-sec-template resource, which controls the appearance of the Malware & Phishing block page.
- The web filtering block page—Devices using the SIA SMB service encounter the Web Filtering block page when trying to access a website that contains content the SIA SMB service blocks. The APIs in this chapter describe how to view and manage the business-web-template resource, which controls the appearance of the Web Filtering block page.

APIs This chapter describes how to perform the following tasks:

Update Web Filtering Block Page Properties for a Subscriber Account

Use the following **PUT** request to update the business-web-template resource (web filtering block page) properties for a specific subscriber account:

```
SSM> PUT /account/<accountId>/resource/business-web-template
{
    "color":"#<hexColorCode>",
    "web-page-show-name": <boolean>,
    "web-page-show-code": <boolean>,
    "web-page-admin-phone": "<phoneNumber>",
    "web-page-message": "<textStringMessage>",
    "tagline": <textString>,
    "web-page-show-message": <boolean>,
    "web-page-show-admin": <boolean>,
    "web-page-admin-email": "<email-address>",
    "rtl": <boolean>
}
```

After modifying web filtering block page properties, use the following API to verify the web filter block page properties are updated as desired:

`GET /account/<accountId>/resource/business-web-template`

See the "Retrieve Web Filtering Block Page Properties for a Subscriber Account" section for details about this API.

NOTE: To verify changes to the "**rtl**" (right to left) field**,** which specifies the direction of the text which appears on the block page, preview the block page in the Subscriber Management application. See the *SIA SMB Administrator's Guide* for details on previewing the block page for a specific subscriber.

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account for which to update the configuration for a resource.
"color"	Optional. Specifies the brand color applied to the web filtering block page. The default setting is the HTML hexadecimal color value **#0294de**.
"web-page-show-name"	Optional. Dictates whether the web filtering block page contains a field where the subscriber can specify a device name along with a **Send Request** button that sends that name to an administrator in a request for access to the blocked website. Replace the ***<boolean>*** argument with of the following keywords: • **true**—Includes the device name field and **Send Request** button on the block page. • **false**—Hides the device name field and **Send Request** button so that those entities do not appear on the block page.
"web-page-show-code"	Optional. Dictates whether the web filtering block page includes the four-character device registration code along with a link that sends that code to an administrator in a request for access to the blocked website. Replace the ***<boolean>*** argument with of the following keywords: • **true**—Includes the code on the block page. • **false**—Hides the code so it does not appear on the block page.
"web-page-admin-phone"	Optional. Specifies the phone number for an administrator; if the "web-page-show-admin" field is set to **true**, this phone number appears on the block page so the subscriber can contact an SIA administrator for help with block page issues (for example, when a subscriber wants to understand why a particular website cannot be accessed). Replace the ***<phone-number>*** argument with the phone number for an SIA administrator.

Field	Description
"web-page-message"	Optional. Specifies a text string that appears on the specified block page if the "web-page-show-message" field is set to **true**. Replace the ***<text-string-message>*** argument with the desired message.
"tagline"	Optional. Specifies a text string that appears as a single line on the block page. By default the **"tagline"** field is not set.
"web-page-show-message"	Optional. Dictates whether the message specified by the "web-page-message" field appears on the specified block page. Replace the ***<boolean>*** argument with of the following keywords: • **true**—Includes the message on the block page. • **false**—Hides the message so it does not appear on the block page.
"web-page-show-admin"	Optional. Dictates whether administrator contact information (specified by the "web-page-admin-email" and "web-page-admin-phone" fields) appears on the specified block page. Replace the ***<boolean>*** argument with of the following keywords: • **true**—Includes the specified administrator contact information on the block page. • **false**—Hides administrator contact information so it does not appear on the block page.
"web-page-admin-email"	Optional. Specifies an email address for an administrator; if the "web-page-show-admin" field is set to **true**, this email address appears on the block page so the subscriber can contact an SIA administrator for help with block page issues (for example, when a subscriber wants to understand why a particular website cannot be accessed). Replace the ***<email-address>*** argument with the email address for an SIA administrator.
"rtl"	Optional. Indicates the intended direction of the text which appears on the block page. By default the "**rtl"** (right to left) field is set to **false**, indicating the block page text is written and read from left to right. For languages written and read from right to left (for example some middle eastern languages such as Hebrew and Arabic), the "**rtl"** field setting is **true**.

Expected Response

A successful response includes HTTP code 200 indicating the specified block page template is updated as specified in the API request.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows demonstrates how to update a single property in a web filtering block page template; this example updates the email address (**web-page-admin-email**) property for a subscriber account called **art.gallery**:

```
SSM> PUT /account/art.gallery/resource/business-web-template
{"web-page-admin-email":"username@example.com"}
```

The example that follows demonstrates how to update multiple properties in a web filter block page template (**business-web-template**); this example updates web filter block page properties for a subscriber account called **joe.coffee**:

```
SSM> PUT /account/joe.coffee/resource/business-web-template
{"web-page-show-name":true, "web-page-show-code":true,
"web-page-admin-phone":"800-555-0175", "web-page-message": "This website is
blocked due to your current protection level.",
"web-page-show-message":true, "web-page-show-admin":true,
"web-page-admin-email":"joe.coffee@example.com", "rtl":false}
```

Update Malware and Phishing Block Page Properties for a Subscriber Account

Use the following **PUT** request to update the properties in the business-sec-template resource (the malware and phishing block page) for a specific subscriber account:

```
SSM> PUT /account/<accountId>/resource/business-sec-template
{
    "sec-page-admin-email": "<email-address>",
    "color":"#<hexColorCode>",
    "sec-page-show-message": <boolean>,
    "sec-page-message": "<text-string-message>",
    "sec-page-show-admin": <boolean>,
    "tagline": <textString>,
    "sec-page-admin-phone": "<phone-number>",
    "rtl": "<boolean>"
}
```

After modifying malware and phishing block page properties, use the following API to verify the web filter block page properties are updated as desired:

GET /account/<*accountId*>/resource/business-sec-template

See the "Retrieve Malware and Phishing Block Page Properties for a Subscriber Account" section for details about this API.

NOTE: To verify changes to the "**rtl**" (right to left) field**,** which specifies the direction of the text which appears on the block page, preview the block page in the Subscriber Management application. See the *SIA SMB Administrator's Guide* for details on previewing the block page for a specific subscriber.

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account for which to update the configuration for a resource.
"sec-page-admin-email"	Optional. Specifies an email address for an administrator; if the "sec-page-show-admin" field is set to **true**, this email address appears on the block page so the subscriber can contact an SIA administrator for help with block page issues (for example, when a subscriber wants to understand why a particular website cannot be accessed). Replace the ***<email-address>*** argument with the email address for an SIA administrator.
"color"	Optional. Specifies the brand color applied to the malware and phishing block page. The default setting is the HTML hexadecimal color value **#0294de**.
"sec-page-show-message"	Optional. Dictates whether the message specified by the "sec-page-message" field appears on the malware and phishing block page. Replace the ***<boolean>*** argument with of the following keywords: • **true**—Includes the message on the block page. • **false**—Hides the message so it does not appear on the block page.
"sec-page-message"	Optional. Specifies a text string that appears on the malware and phishing block page if the "sec-page-show-message" field is set to **true**. Replace the ***<text-string-message>*** argument with the desired message.
"sec-page-show-admin"	Optional. Dictates whether administrator contact information (specified by the "sec-page-admin-email" and "sec-page-admin-phone" fields) appears on the specified block page. Replace the ***<boolean>*** argument with of the following keywords: • **true**—Includes the specified administrator contact information on the block page. • **false**—Hides administrator contact information so it does not appear on the block page.
"tagline"	Optional. Specifies a text string that appears as a single line on the block page. By default the **"tagline"** field is not set.

Field	Description
"sec-page-admin-phone"	Optional. Specifies the phone number for an administrator; if the "sec-page-show-admin" field is set to **true**, this phone number appears on the block page so the subscriber can contact an SIA administrator for help with block page issues (for example, when a subscriber wants to understand why a particular website cannot be accessed). Replace the ***<phone-number>*** argument with the phone number for an SIA administrator.
"rtl"	Optional. Indicates the intended direction of the text which appears on the block page. By default the "**rtl"** (right to left) field is set to **false**, indicating the block page text is written and read from left to right. For languages written and read from right to left (for example some middle eastern languages such as Hebrew and Arabic), the "**rtl"** field setting is **true**.

Expected Response

A successful response includes HTTP code 200 indicating the specified block page is updated as specified in the API request.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Examples

The example that follows demonstrates how to update a single property in a malware and phishing (**business-sec-template**) block page; this example updates the phone number that appears on the block page (when the **"sec-page-show-admin"** field is set to **true)** for a subscriber account called **art.gallery**:

```
SSM> PUT /account/art.gallery/resource/business-sec-template
{"sec-page-admin-phone":"800-555-0175"}
```

The example that follows demonstrates how to update multiple properties in a malware and phishing block page (**business-sec-template**); this example updates malware and phishing block page properties for a subscriber account called **joe.coffee**:

```
SSM> PUT /account/joe.coffee/resource/business-sec-template
{"sec-page-admin-email": "joe.coffee@example.com",
"sec-page-show-message": "true", "sec-page-message": "This website
is blocked", "sec-page-show-admin": "true", "sec-page-admin-phone":
"800-555-0134"}
```

Retrieve Resource Configuration Details for a Subscriber Account

Use the following **GET** API to retrieve configuration details for the business-sec-template (malware and phishing block page) and business-web-template (web filtering block page) resources:

```
SSM> GET /account/<accountId>/resource
```

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account for which to display resource configuration details.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows configuration details for the specified resource:

```
{
  "total": <integer>,
  "limit": <integer>,
  "content": [
      {
          "name": "business-sec-template",
          "type": "text/html",
          "fields": [
              {
                  "name": "<propertyId>",
                  "defaultValue": "<propertyValue>",
                  "validations": {
                      "maxLength": <integer>
                  }
              },
              ...
          ]
      },
      {
          "name": "business-web-template",
          "type": "text/html",
          "fields": [
              {
                  "name": "<propertyId>",
                  "defaultValue": "<propertyValue>",
                  "validations": {
```

```
                        "maxLength": <integer>
                    }
                },
                ...
            ]
        }
    ]
}
```

The table that follows describes the possible fields in the API response:

Field	Description
"total"	Total number of block page templates that exist for the specified subscriber account.
"limit"	Maximum number of block page templates that can be included in the API response. **NOTE:** See the "Limiting Retrieved Results for GET Requests" section in Chapter 1, "Introduction to the SSM Restful API," for information about how to set the limit field for **GET** requests.
"name"	Text string that identifies the resource for which the API displays configuration values. Possible resources are: • business-sec-template—Displays configuration details for the malware and phishing block page; for example, the text which appears on the block page and so forth. • business-web-template—Displays configuration details for the web filtering block page; for example, the text which appears on the block page and so forth.
"type"	Text string that identifies the type of resource for which configuration information is displayed. The business-sec-template and business-web-template resources are text HTML pages that define the content of the associated block pages (for example, the text message that appears on the block page and contact information for an administrator).

Field	Description
"fields"	Displays the following information for each malware and phishing (business-sec-template) or web filtering (business-web-template) property: ● "name"—Identifies a malware and phishing (business-sec-template) or web filtering (business-web-template) property. Each resource has its own fields; see the “Possible resource fields” table for details about the fields displayed for each resource. ● "defaultValue"—Displays the default value for the specified property. **NOTE:** Some properties do not have default values; this means the property is not set until manually configured. ● "maxLength"—A validation field that specifies the maximum number of characters a property value can contain. For example, the text string specified in the “tagline” field cannot exceed 140 characters, so the "maxLength" value for the “tagline” field is 140. Values that exceed the number set by the "maxLength" field are not valid. **NOTE:** The "maxLength" appears only for applicable properties. See the “business-sec-template properties” section for a description of the possible business-sec-template block page properties. See the “business-web-template properties” section for a description of the possible business-web-template block page properties.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

business-sec-template Properties

The table that follows lists default values and maximum length settings for each property in the business-sec-template block page resource.

Possible Property Fields in the business-sec-template Resource

Field	Description
"color"	Identifies the brand color the SSM applies to the malware and phishing block page. The "defaultValue" is set to HTML hexadecimal color value #0294de; this is the default branding color applied to the malware and phishing block page. The "maxLength" value is set to 30; in other words, the name specified in the “color” field can contain a maximum of 30 characters.

Field	Description
“tagline”	If set, displays the specified text string on the malware and phishing block page; if not set, this field is empty in the API response and no tagline appears on the malware and phishing block page. By default, no tagline is specified. The "maxLength" value for the “tagline” property is set to 140; in other words, the tagline can contain a maximum of 140 characters.
"sec-page-show-message"	Indicates whether the malware and phishing block page displays the text string specified in the "sec-page-message" field. When set to true, the block page includes the message; when set to false, the message does not appear on the block page. The “defaultValue” setting is **true**.
"sec-page-message"	Text string message that appears on the malware and phishing block page if the "sec-page-show-message” field is set to true. If the "sec-page-show-message” field is set to false, the block page does not include this message. The default value for the "sec-page-message" propery is the following text: “This site is blocked because it is a known security threat. Please contact your network administrator to gain access.”
"sec-page-show-admin"	Indicates whether the malware and phishing block page displays the administrator phone number from the "sec-page-admin-phone" field). When set to true, the block page includes the specified phone number; when set to false, the phone number does not appear on the block page. The “defaultValue” setting is **false**.
"sec-page-admin-email"	If specified, displays an administrator email address that appears on the malware and phishing block page if the “sec-page-show-admin" field is set to true. If the “sec-page-show-admin" field is set to false, the malware and phishing block page does not include an administrator email address. By default, no email address is specified. The "maxLength" value for the email address is 255 characters; in other words, a specified email address can contain a maximum of 255 characters

Field	Description
"sec-page-admin-phone"	If specified, displays an administrator phone number that appears on the malware and phishing block page if the "sec-page-show-admin" field is set to true. If the "sec-page-show-admin" field is set to false, the specified phone number does not appear on the block page. By default, no phone number is specified. The "maxLength" value for the phone number is 255 characters; in other words, a specified phone number can be a maximum of 255 characters.

business-web-template Properties

The table that follows lists default values and maximum length settings for each property in the business-web-template block page resource.

Possible Resource Fields from the business-web-template

Field	Description
"color"	Identifies the brand color the SSM applies to the web filtering block page. The default setting is the HTML hexadecimal color value **`#0294de`**. The "maxLength" value for the “color” field is set to 30; in other words, the name specified in the “color” field can contain a maximum of 30 characters. **NOTE:** See the “color” field description for details on how to modify the branding color for the Web Filtering block page.
“tagline”	If set, displays the specified text string on the web filtering block page; if not set, this field is empty in the API response and no tagline appears on the web filtering block page. By default the "tagline" field is not set. **NOTE:** See the "tagline" field description for details on how to specify a tagline that appears on a web filtering block page.
"web-page-show-message"	Indicates whether the web filtering block page displays the text string specified in the "web-page-message" field.

Field	Description
"web-page-message"	Text string message that appears on the web filtering block page if the "web-page-show-message" field is set to true. If the "web-page-show-message" field is set to false, the block page does not include this message. The "defaultValue" setting for the "web-page-message" field is **true**; in other words, the message that is specified in the "web-page-message" field appears on the web filtering block page by default. **NOTE:** See the "web-page-message" field description for details on how to show and hide a message on a web filtering block page.
"web-page-show-admin"	Indicates whether the web filtering block page displays the administrator phone number from the "web-page-admin-phone" field). When set to true, the block page includes the specified phone number; when set to false, the phone number does not appear on the block page. The "defaultValue" setting is **false**. **NOTE:** See the "web-page-admin-email" and "web-page-admin-phone" field descriptions for details on how to include administrator contact information on a web filtering block page.
"web-page-show-code"	Indicates whether the web filtering block page includes the four-character device registration code along with a link that sends that code to an administrator in a request for access to the blocked website. A value of true indicates the block page includes the device registration code and link; a value of false indicates the block page does not include the registration code or link. The "defaultValue" setting is false.. **NOTE:** See the "web-page-show-code" field description for details on how to include a four-character device registration code and link on a web filtering block page.
"web-page-show-name"	Indicates whether the block page contains a field where the subscriber can specify a device name along with a **Send Request** button that sends that name to an administrator in a request for access to the blocked website. A value of true indicates the block page includes the field and **Send Request** button; a value of false indicates the block page does not display the field or button. The "defaultValue" setting is **false**. **NOTE:** See the "web-page-show-name" field description for details on how to include a device name field with a **Send Request** button on a web filtering block page

Field	Description
"web-page-admin-email"	An administrator email address which appears on the web filtering block page if the “web-page-show-admin" field is set to true. If the “web-page-show-admin" field is set to false, the web filtering block page does not include an administrator email address. The “defaultValue” setting is false. The "maxLength" value is 255; in other words, the specified administrator email address can be a maximum of 255 characters. **NOTE:** See the "web-page-admin-email" field description to specify an administrator email address that appears on the web filtering block page when the "web-page-show-admin" field is set to **true**.
"web-page-admin-phone"	The phone number for an administrator. If the "web-page-show-admin" field is set to true, this phone number appears on the block page. If the "web-page-show-admin" field is set to false, the specified phone number does not appear on the block page. The “defaultValue” setting is false. The "maxLength" value is 255; in other words, a specified phone number can be a maximum of 255 characters. **NOTE:** See the "web-page-admin-phone" field description to set the phone number that appears on the web filtering page when the "web-page-show-admin" field is set to **true**

Example

The example that follows retrieves resource configuration details for a subscriber account called **joe.coffee**:

```
SSM> GET /account/joe.coffee/resource
{
    "total": 2,
    "limit": 2,
    "content": [
        {
            "name": "business-sec-template",
            "type": "text/html",
            "fields": [
                {
                    "name": "color",
                    "defaultValue": "#0294de",
                    "validations": {
                        "maxLength": 30
                    }
                },
                {
```

```
                    "name": "tagline",
                    "validations": {
                        "maxLength": 140
                    }
                },
                {
                    "name": "sec-page-show-message",
                    "defaultValue": true
                },
                {
                    "name": "sec-page-message",
                    "defaultValue": "This site is blocked because
it is a known security threat. Please contact your network
administrator to gain access.",
                    "validations": {
                        "maxLength": 255
                    }
                },
                {
                    "name": "sec-page-show-admin",
                    "defaultValue": false
                },
                {
                    "name": "sec-page-admin-email",
                    "validations": {
                        "maxLength": 255
                    }
                },
                {
                    "name": "sec-page-admin-phone",
                    "validations": {
                        "maxLength": 255
                    }
                }
            ]
        },
        {
            "name": "business-web-template",
            "type": "text/html",
            "fields": [
                {
                    "name": "color",
                    "defaultValue": "#0294de",
                    "validations": {
                        "maxLength": 30
                    }
                },
                {
                    "name": "tagline",
                    "validations": {
                        "maxLength": 140
```

```
                    }
                },
                {
                    "name": "web-page-show-message",
                    "defaultValue": true
                },
                {
                    "name": "web-page-message",
                    "defaultValue": "This website is blocked due to
your current protection level. Please contact your network
administrator to gain access.",
                    "validations": {
                        "maxLength": 255
                    }
                },
                {
                    "name": "web-page-show-admin",
                    "defaultValue": false
                },
                {
                    "name": "web-page-show-code",
                    "defaultValue": false
                },
                {
                    "name": "web-page-show-name",
                    "defaultValue": false
                },
                {
                    "name": "web-page-admin-email",
                    "validations": {
                        "maxLength": 255
                    }
                },
                {
                    "name": "web-page-admin-phone",
                    "validations": {
                        "maxLength": 255
                    }
                }
            ]
        }
    ]
}
```

Retrieve Web Filtering Block Page Properties for a Subscriber Account

Use the following **GET** API to retrieve the current web filtering block page property settings for a subscriber account:

```
SSM> GET /account/<accountId>/resource/business-web-template
```

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account for which to display web filtering (**business-web-template**) block page template properties.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows configuration details for the specified resource:

```
{
    "name": "business-web-template",
    "type": "text/html",
    "fields": {
        "color": "#<hexColorCode>",
        "web-page-show-name": "<boolean>",
        "web-page-show-code": "<boolean>",
        "web-page-admin-phone": "<phoneNumber>",
        "web-page-message": "<textString>",
        "tagline": "<textString>",
        "web-page-show-message": "<boolean>",
        "web-page-show-admin": "<boolean>",
        "web-page-admin-email": "<emailAddress>"
    }
}
```

The table that follows describes the fields in the API response:

Possible Resource Fields from the business-web-template

Field	Description
"name"	Identifies the resource for which the API displays configuration values (in this API, the resource is the **business-web-template,** which is the web filtering block page).
"type"	Text string that identifies the **business-web-template** resource format is **text/html**.
"fields"—The response includes a "fields" section that shows configuration details for each block page property. The rows that follow describe each property field.	
"color"	Identifies the brand color the SSM applies to the web filtering block page. The default setting is the HTML hexadecimal color value **#0294de**.
"web-page-show-name"	Indicates whether the block page contains a field where the subscriber can specify a device name along with a **Send Request** button that sends that name to an administrator in a request for access to the blocked website. A value of true indicates the block page includes the field and **Send Request** button; a value of false indicates the block page does not display the field or button.
"web-page-show-code"	Indicates whether the web filtering block page includes the four-character device registration code along with a link that sends that code to an administrator in a request for access to the blocked website. A value of true indicates the block page includes the device registration code and link; a value of false indicates the block page does not include the registration code or link. The default setting is false.
"web-page-admin-phone"	The phone number for an administrator. If the "web-page-show-admin" field is set to true, this phone number appears on the block page. If the "web-page-show-admin" field is set to false, the specified phone number does not appear on the block page. The default setting is false.
"web-page-message"	Text string message that appears on the web filtering block page if the "web-page-show-message" field is set to true. If the "web-page-show-message" field is set to false, the block page does not include this message.
“tagline”	If set, displays the specified text string on the web filtering block page; if not set, this field is empty in the API response and no tagline appears on the web filtering block page. By default, no tagline is specified.

Field	Description
"web-page-show-message"	Indicates whether the web filtering block page displays the text string specified in the "web-page-message" field. When set to true, the block page includes the message; when set to false, the message does not appear on the block page. The default setting is true..
"web-page-show-admin"	Indicates whether the web filtering block page displays the administrator phone number from the "web-page-admin-phone" field). When set to true, the block page includes the specified phone number; when set to false, the phone number does not appear on the block page. The default setting is false.
"web-page-admin-email"	An administrator email address which appears on the web filtering block page if the "web-page-show-admin" field is set to true. If the "web-page-show-admin" field is set to false, the web filtering block page does not include an administrator email address. The default setting is false.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows retrieves **business-sec-template** (malware and phishing block page) resource settings for a subscriber account called **joe.coffee**:

```
SSM> GET /account/joe.coffee/resource/business-web-template
{
    "name": "business-web-template",
    "type": "text/html",
    "fields": {
        "color": "#0294de",
        "web-page-show-name": "false",
        "web-page-show-code": "false",
        "web-page-admin-phone": "800-555-0134",
        "web-page-message": "This website is blocked due to your current
protection level. Please contact your network administrator to gain
access.",
        "tagline": "",
        "web-page-show-message": "true",
        "web-page-show-admin": "true",
        "web-page-admin-email": "joe.coffee@example.com"
    }
}
```

Retrieve Malware and Phishing Block Page Properties for a Subscriber Account

Use the following **GET** API to retrieve the current malware and phishing block page property settings for a subscriber account:

```
SSM> GET /account/<accountId>/resource/business-sec-template
```

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account for which to display malware and phishing (**business-sec-template**) block page property settings.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows configuration details from the malware and phishing block page template (**business-sec-template**):

```
{
    "name": "business-sec-template",
    "type": "text/html",
    "fields": {
        "sec-page-admin-email": "<emailAddress>",
        "color": "#<hexColorCode>",
        "sec-page-show-message": "<boolean>",
        "sec-page-message": "<textString>",
        "sec-page-show-admin": "<boolean>",
        "tagline": "<textString>",
        "sec-page-admin-phone": "<phoneNumber>"
    }
}
```

The table that follows describes the fields in the API response:

Possible Resource Fields from the business-sec-template

Field	Description
"name"	Identifies the resource for which the API displays configuration values (in this API, the resource is the **business-sec-template,** which is the malware and phishing block page).
"type"	Text string that identifies the **business-sec-template** resource format is **text/html**.
"fields"—The response includes a "fields" section that shows configuration details for each block page property. The rows that follow describe each property field.	
"sec-page-admin-email"	An administrator email address which appears on the malware and phishing block page if the "sec-page-show-admin" field is set to true. If the "sec-page-show-admin" field is set to false, the malware and phishing block page does not include an administrator email address. The default setting is false.
"color"	Identifies the brand color the SSM applies to the malware and phishing block page. The default setting is the HTML hexadecimal color value **#00b3e3**.
"sec-page-show-message"	Indicates whether the malware and phishing block page displays the text string specified in the "sec-page-message" field. When set to true, the block page includes the message; when set to false, the message does not appear on the block page. The default setting is true..
"sec-page-message"	Text string message that appears on the malware and phishing block page if the "sec-page-show-message" field is set to true. If the "sec-page-show-message" field is set to false, the block page does not include this message.
"sec-page-show-admin"	Indicates whether the malware and phishing block page displays the administrator phone number from the "sec-page-admin-phone" field). When set to true, the block page includes the specified phone number; when set to false, the phone number does not appear on the block page. The default setting is false.
"tagline"	If set, displays the specified text string on the malware and phishing block page; if not set, this field is empty in the API response and no tagline appears on the malware and phishing block page. By default, no tagline is specified.

Field	Description
"sec-page-admin-phone"	The phone number for an administrator. If the "sec-page-show-admin" field is set to true, this phone number appears on the block page. If the "sec-page-show-admin" field is set to false, the specified phone number does not appear on the block page. The default setting is false.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows retrieves **business-sec-template** resource settings for a subscriber account called **joe.coffee**:

```
SSM> GET /account/joe.coffee/resource/business-sec-template
{
    "name": "business-sec-template",
    "type": "text/html",
    "fields": {
        "sec-page-admin-email": "joe.coffee@example.com",
        "color": "#0294de",
        "sec-page-show-message": "true",
        "sec-page-message": "This site is blocked because it is a
known security threat. Please contact your network administrator to
gain access.",
        "sec-page-show-admin": "true",
        "tagline": "",
        "sec-page-admin-phone": "800-555-0134"
    }
}
```

Retrieve a Preview of a Specific Subscriber Resource

Use the following **GET** API to return an HTML preview of the block pages associated with a specific SIA SMB subscriber account:

```
SSM> GET /account/<accountId>/resource/<resourceId>/preview
```

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account for which to display block page configuration details.
<resourceId>	Required. String that identifies the block page for which to retrieve configuration information. Replace the ***<resourceId>*** argument with one of the following resource object keywords: • **business-sec-template**—Retrieves malware and phishing filtering block page configuration details. • **business-web-template**—Retrieves web filtering block page configuration details.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The response body shows the HTML code for the requested resource. Be aware the response for a block page resource can be extensive.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows shows sample output for a request for an HTML preview of the **business-web-template** resource for the subscriber called **joe.coffee**. Because the API response is extensive, this example uses the **limit** field to limit the response to five (**5**) resources:

```
SSM> GET /account/joe.coffee/resource/business-web-template/preview?limit=5

<!DOCTYPE html>
<html lang="en">
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <meta charset="UTF-8">
    <title>Access Blocked</title>

    <script>

        const settings = {
            showMessage: getBoolean('true'),
```

```
            showAdmin:   getBoolean('true'),
            email:       'blocked@sportal.com',
            phone:       '11-22-33-44-55'
        };


        document.addEventListener("DOMContentLoaded", ready);

        function getBoolean(value) {
            return value === "true" || value === true;
        }

        function ready() {
            showLogo();
            showMessage();
            showAdminInfo();
        }

        function showLogo() {
          const imgElement = document.getElementById('logo');




          const imgPreview = localStorage.getItem('sb-branding-logo');
          imgElement.src = imgPreview ||
"/account/joe.coffee/resource/sb-logo/preview";


          imgElement.onerror = function() {
              imgElement.classList.add('hidden');
          }
        }

        function showMessage() {
            settings.showMessage &&
show(document.getElementById('message'));
```

```
        }

        function showAdminInfo() {
            if (settings.showAdmin)
show(document.getElementById('admin-info'));
            if (settings.email)
show(document.getElementById('admin-email'));
            if (settings.phone)
show(document.getElementById('admin-phone'));
        }

        function show(el) {
            el && el.classList.remove('hidden');
        }
    </script>
    <style>
            * {
                box-sizing: border-box;
            }

            .rtl {
                direction: rtl;
            }

            body {
                font-family: Open Sans, Arial, sans-serif;
                font-size: 14px;
                line-height: 22px;
                font-weight: 300;
                color: #212121;
                display: flex;
                flex-direction: column;
            }

            body, html {
                margin: 0;
                padding: 0;
```

```
        overflow: auto;
    }

    h1, h2, p {
        margin: 0;
    }

    h1 {
        font-size: 24px;
        font-weight: normal;
        line-height: 36px;
        text-align: center;
    }

    h2 {
        font-size: 20px;
        line-height: 30px;
        font-weight: 500;
    }

    header {
        min-height: 80px;
        display: flex;
        align-items: center;
        border-bottom: 1px solid #D3DCE1;
        padding: 12px 24px;
        margin-bottom: 48px;
    }

    main {
        display: flex;
        flex-direction: column;
        align-items: center;
    }

    section {
        text-align: center;
```

```
        }

        a {
            text-decoration: none;
        }

        .action-link {
            color: inherit;
            cursor: pointer;
            vertical-align: middle;
            display: inline-block;
        }

        .action-link:hover {
            filter: brightness(80%);
        }

        .hidden {
            display: none !important;
        }
    </style>
    <style>
        .header__tagline {
            color: #65666A;
            text-align: left;
            min-width: 0;
            flex: 1 1;
            overflow-wrap: break-word;
        }

        .header__logo img {
            max-width: 240px;
            max-height: 60px;
            margin-inline-end: 24px;
        }

        .rtl .header__tagline {
```

```
        text-align: right;
    }

    .title {
        display: flex;
        flex-direction: column;
        align-items: center;
        margin-bottom: 16px;
    }

    .icon__wrapper {
        position: relative;
        margin-bottom: 32px;
    }

    .icon__place-mat {
        width: 80px;
        height: 80px;
        border-radius: 50%;
        opacity: 16%;
    }

    .icon__svg {
        position: absolute;
        top: 19px;
        left: 20px;
    }

    .message {
        width: 50%;
        text-align: center;
        margin-bottom: 40px;
        overflow-wrap: break-word;
    }

    .admin-info {
        width: 50%;
```

```
            margin-bottom: 36px;
        }

        .admin-info__header {
            margin-bottom: 16px;
        }

        .admin-contact {
            display: flex;
            align-items: flex-start;
            justify-content: center;
            margin-bottom: 8px;
        }

        .admin-contact .action-link {
            line-height: normal;
            word-break: break-word;
            direction: ltr;
        }

        .phone-number {
            direction: ltr;
        }

        .icon-container {
            margin-right: 10px;
            display: inline-block;
            vertical-align: middle;
        }

        .rtl .icon-container {
            margin-left: 10px;
            margin-right: 0;
        }
    </style>
    <style>
        @media (max-width: 768px) {
```

```
                main {
                    padding: 0 16px;
                }

                header {
                    margin-bottom: 24px;
                    flex-wrap: wrap;
                }

                .header__tagline {
                    flex: initial;
                    min-width: auto;
                    overflow: hidden;
                }

                .message, .admin-info {
                    width: 100%;
                }
            }
        </style>
    <style>
      .primary-color {
          color: ;
          color: #DE0274;
        }
      .primary-bg-color {
          background: ;
          background: #DE0274;
        }
    </style>
</head>

<body>
    <header>
        <div class="header__logo"><img id="logo"></div>
        <h1 class="header__tagline">Block Page Sample Title</h1>
    </header>
```

```
    <main>
        <div class="title">
            <div class="icon__wrapper primary-color">
                <div class="icon__place-mat primary-bg-color"></div>
                <svg width="40px" height="40px" focusable="false"
class="icon__svg" xmlns="http://www.w3.org/2000/svg"
                     viewBox="0 0 448 512">
                    <path fill="currentColor"
                          d="M400 224h-24v-72C376 68.2 307.8 0 224 0S72 68.2
72 152v72H48c-26.5 0-48 21.5-48 48v192c0 26.5 21.5 48 48 48h352c26.5 0
48-21.5 48-48V272c0-26.5-21.5-48-48-48zm-104 0H152v-72c0-39.7 32.3-72
72-72s72 32.3 72 72v72z"></path>
                </svg>
            </div>
            <h1 class="title__header">Malware and Phishing</h1>
        </div>

        <div id="message" class="message text-center hidden">website is
blocked</div>

        <section id="admin-info" class="admin-info hidden">
            <h2 class="admin-info__header">Contact Admin</h2>
            <div id="admin-email" class="admin-contact hidden">
                <div class="icon-container primary-color">
                    <svg width="16px" height="16px" focusable="false"
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 512 512">
                        <path fill="currentColor" d="M502.3 190.8c3.9-3.1
9.7-.2 9.7 4.7V400c0 26.5-21.5 48-48 48H48c-26.5 0-48-21.5-48-48V195.6c0-5
5.7-7.8 9.7-4.7 22.4 17.4 52.1 39.5 154.1 113.6 21.1 15.4 56.7 47.8 92.2
47.6 35.7.3 72-32.8 92.3-47.6 102-74.1 131.6-96.3 154-113.7zM256 320c23.2.4
56.6-29.2 73.4-41.4 132.7-96.3 142.8-104.7 173.4-128.7 5.8-4.5 9.2-11.5
9.2-18.9v-19c0-26.5-21.5-48-48-48H48C21.5 64 0 85.5 0 112v19c0 7.4 3.4 14.3
9.2 18.9 30.6 23.9 40.7 32.4 173.4 128.7 16.8 12.2 50.2 41.8 73.4
41.4z"></path>
                    </svg>
                </div>
```

```
                <a id="admin-email-link" target="_top" class="action-link
primary-color" href="mailto:blocked@sportal.com">blocked@sportal.com</a>
            </div>

            <div id="admin-phone" class="admin-contact hidden">
                <div class="icon-container primary-color">
                    <svg width="16px" height="16px" focusable="false"
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 512 512">
                        <path fill="currentColor" d="M497.39
361.8l-112-48a24 24 0 0 0-28 6.9l-49.6 60.6A370.66 370.66 0 0 1 130.6
204.11160.6-49.6a23.94 23.94 0 0 0 6.9-28l-48-112A24.16 24.16 0 0 0
122.6.61l-104 24A24 24 0 0 0 0 48c0 256.5 207.9 464 464 464a24 24 0 0 0
23.4-18.6l24-104a24.29 24.29 0 0 0-14.01-27.6z"></path>
                        <path fill="currentColor" d="M497.39
361.8l-112-48a24 24 0 0 0-28 6.9l-49.6 60.6A370.66 370.66 0 0 1 130.6
204.11160.6-49.6a23.94 23.94 0 0 0 6.9-28l-48-112A24.16 24.16 0 0 0
122.6.61l-104 24A24 24 0 0 0 0 48c0 256.5 207.9 464 464 464a24 24 0 0 0
23.4-18.6l24-104a24.29 24.29 0 0 0-14.01-27.6z"></path>
                    </svg>
                </div>
                <a id="admin-phone-link" target="_top" class="action-link
phone-number primary-color" href="tel:11-22-33-44-55">11-22-33-44-55</a>
            </div>
        </section>
    </main>
</body>

</html>
<script>if(window.top === window.self){var xhr = typeof
window.XDomainRequest !== "undefined" ? new XDomainRequest() : new
XMLHttpRequest();xhr.open("POST", window.location.protocol + "//" +
"/secure/beacon".replace(/\{\{id\}\}/, ""), true);xhr.send(null);}</script>
```

Chapter 9: Device APIs

This chapter describes the APIs for performing the following tasks:

- Add devices to a subscriber account
- Update the configuration for a subscriber device
- Update multiple devices for a subscriber account
- Retrieve devices associated with the specified subscriber account
- Remove a specific device from a subscriber account

Add Devices to a Subscriber Account

Use the following **POST** API to add one or more devices to a subscriber account:

```
SSM> POST /account/<accountId>/device
[
  {
    "name": "<deviceName>",
    "address": "<deviceAddress>",
    "profile": "<profileName>",
  }
  ...
]
```

Fields

Field	Description
<accountId>	Required. Identifies the subscriber account to which you want to add a device. Replace the ***<accountId>*** argument with the desired account identifier.
"name"	Required. Unique string that identifies the device you want to add to the subscriber account. Replace the ***<deviceName>*** argument with a name for the device you are adding.
"address"	Required. Specifies the MAC or MSISDN (Mobile Station International Subscriber Directory Number) address assigned to the device you want to add. Replace the ***<deviceAddress>*** argument with the device address. **NOTE:** When updating a device, you cannot modify the address.

Field	Description
“profile”	Optional. Specifies the name of the SIA SMB profile to attach to the device Replace the ***<profileName>*** argument with the name of the profile you want to attach to the device. **NOTE:** The profile name is case-sensitive. If the API request does not include the **“profile”** construct, the API attaches the device to the **sb_group_visitor** profile by default. **NOTE:** In the SIA SMB subscriber portal application, profiles are called “device groups.”

Expected Response

A successful response includes HTTP code 201 with a JSON based response body indicating the API is successful. The API response body includes data about the devices you added to the specified subscriber account as demonstrated in the following example:

```
{
    "processed": [
        {
            "name": "<deviceName>",
            "address": "<deviceAddress>",
            "profile": "<profileName>",
            "last-seen": null
        },
        ...
    ],
    "status": "success"
}
```

The table that follows describes the fields in the API response; the API shows these fields for each device you successfully add to the subscriber account:

Field	Description
"name"	Identifies the device the API added to the subscriber account.
“address”	Identifies the MAC or MSISDN address assigned to the device.
“profile”	Identifies the profile attached to the specified device. **NOTE:** In the SIA SMB subscriber portal application, profiles are called “device groups.”

Field	Description
"last-seen"	Indicates the last time the specified device accessed the network, where the date and time is expressed in the following format: *<year-month-day>T<hour:min:seconds><z-time>Z* **NOTE:** The "last-seen" value is automatically set from the reporting pipeline and is **null** when the device is initially created.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows adds a device (called **my-phone-os13** and associated with the address **00:0a:95:9d:64:16**) to a subscriber account called **joe.coffee**. In this example, the API specifies the date and time at which the device was last seen on the network and applies the profile called **sb_group_visitor** with the device:

```
SSM> POST /account/joe.coffee/device [{"name": "my-phone-os13",
"address": "00:0a:95:9d:64:16", "profile": "sb_group_visitor"}]
{
    "processed": [
        {
            "name": "my-phone",
            "address": "00:0a:95:9d:64:16",
            "profile": "sb_group_visitor",
            "last-seen": null
        }
    ],
    "status": "success"
}
```

The example that follows demonstrates how to add multiple devices in one API request. This example adds two devices to a subscriber account called **art.gallery**:

```
SSM> POST /account/art.gallery/device [{"name":
"art-gallery-phone11-A2161-OS13", "address": "00:0a:95:9d:64:17",
"profile": "sb_group_visitor"}, {"name": "art-gallery-tablet2-OS12",
"address": "00:0a:33:3d:40:20", , "profile": "sb_group_visitor"}]
{
    "processed": [
        {
            "name": "art-gallery-phone11-A2161-OS13",
            "address": "00:0a:95:9d:64:17",
            "profile": "sb_group_visitor",
            "last-seen": null
        },
        {
```

```
            "name": "art-gallery-tablet2-OS12",
            "address": "00:0a:33:3d:40:20",
            "profile": "sb_group_visitor",
            "last-seen": null
        }
    ],
    "status": "success"
}
```

Update the Configuration for a Subscriber Device

Use the following **PATCH** API to partially update the configuration of a subscriber device:

```
SSM> PATCH /account/<accountId>/device/<deviceAddress>
{
    "name": "<deviceName>",
    "address": "<deviceAddress>",
    "profile": "<profileName>",
}
```

NOTE: Although all fields in the request body are optional, you need to specify at least one field in the API call.

Fields

Field	Description
<accountId>	Required. Identifies the subscriber account that is associated with the device you want to update. Replace the ***<accountId>*** argument with the desired account identifier.
“address“	Required.Specifies the address that is associated with the device you want to update.
"name"	Optional. Updates the name of the device. Replace the ***<deviceName>*** argument with a name for the device you are updating.
“profile”	Optional. Updates the profile that is attached to the device Replace the ***<profileName>*** argument with the name of the profile you want to attach to the device. **NOTE:** The profile name is case-sensitive. The default profile is the **sb_group_visitor** profile. **NOTE:** In the SIA SMB subscriber portal application, profiles are called “device groups.”

Expected Response

The SSM returns the following output indicating the request is successful; the output includes the device configuration fields the API updated, along with the new value for each update field:

```
{
    "successful": [
        {
            "field": "<fieldName>",
            "value": "<fieldValue>"
        },
        ...
    ],
    "status": "success"
}
```

The table that follows describes the fields in the API response; the response body includes each field the API updates:

Field	Description
"field"	Identifies the field the API updated.
“value”	Displays the updated value for the specified field.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows updates the **profile** that is attached to a device that has the address **17739381674** and is associated with a subscriber account called **joe.coffee**:

```
SSM> PATCH /account/joe.coffee/device/17739381674 {"profile":
"sb_group_employee"}
{
    "successful": [
        {
            "field": "profile",
            "value": "sb_group_employee"
        }
    ],
    "status": "success"
}
```

Update Multiple Devices for a Subscriber Account

Use the following **PATCH** API to update multiple devices belonging to a subscriber account in a single API call:

```
SSM> PATCH /account/<accountId>/device
{

    "update": [
        {
            "name": "<deviceName>",
            "address": "<deviceAddress>",
            "profile": "<profileName>"
        }
        ...
    ]
}
```

NOTE: When updating a device, be aware this **PATCH** API simply modifies the specified fields only and does not overwrite the entire device configuration. When adding a new device, the device needs to exist; this API does not create a device object if a physical device does not exist.

Fields

Field	Description
<accountId>	Required. Identifies the subscriber account associated with the device you want to update. Replace the ***<accountId>*** argument with the desired account identifier.
"name"	Required. Identifies the logical device you want to add, remove, or update. Replace the ***<deviceName>*** argument with a string that identifies the appropriate device.
"address"	Required for adding and updating devices. Specifies the MAC or MSISDN (Mobile Station International Subscriber Directory Number) address assigned to the device you want to add, remove, or update. **NOTE:** When updating a device, you cannot modify the address.
"profile"	Optional. Updates the profile that is attached to the device. Replace the ***<profileName>*** argument with the name of the profile to attach to the device. **NOTE:** The profile name is case-sensitive. The default profile is the **sb_group_visitor** profile. **NOTE:** In the SIA SMB subscriber portal application, profiles are called “device groups.”

Expected Response

A successful response includes the HTTP code 200 with a JSON based response body indicating the device is updated as specified in the API request.

For API requests updating one or more existing devices, the response body includes general configuration information for each device as shown in the following example:

```
{
    "processed": [
        {
            "name": "<deviceName>",
            "address": "<deviceAddress>",
            "profile": "<profileName>",
            "last-seen": <date>T<time>Z
        }
        ...
    ],
    "status": "success"
}
```

The table that follows describes the possible fields the API response can include:

Field	Description
"name"	Identifies the field the API updated.
“address”	Displays the address that is associated with the device.
“profile”	Identifies the profile that is attached to the device. **NOTE:** In the SIA SMB subscriber portal application, profiles are called “device groups.”
"last-seen"	Indicates the date and time when the device last accessed the network, expressed in the following format: <*year-month-day*>T<*hour*:*min*:*seconds*><*z-time*>Z **NOTE:** The "last-seen" value is automatically set from the from reporting pipeline; the value is **null** when the device is initially created.
"status"	Indicates whether the API request was successful.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows updates the following devices, which are associated with a subscriber account called **joe.coffee**:

- The device that has the line address **17739381674**.
- The device that has the hardware address **00:0a:95:9d:64:16**.

```
SSM> PATCH /account/joe.coffee/device {"update":
[{"name":"joe-tablet-os12-rename", "address":"17739381674",
"profile": "sb_group_employee"}, {"name":
"my-phone-os13", "address":"00:0a:95:9d:64:16", "profile":
"sb_group_employee"}]}
{
    "processed": [
        {
            "name": "joe-tablet-os12-rename",
            "address": "17739381674",
            "profile": "sb_group_employee",
            "last-seen": null
        },
        {
            "name": "my-phone-os13",
            "address": "00:0a:95:9d:64:16",
            "profile": "sb_group_employee",
            "last-seen": null
        }
    ],
    "status": "success"
}
```

Retrieve Devices Associated with the Specified Subscriber Account

Use the following **GET** API to retrieve a list of all devices currently associated with a subscriber account:

```
SSM> GET /account/<accountId>/device
```

Fields

Field	Description
<accountId>	Required. Identifies the subscriber account for which to display a list of associated devices. Replace the ***<accountId>*** argument with the desired account identifier.

Expected Response

A successful response returns an HTTP code 200 with a JSON based response body displaying a list of devices associated with the specified account as shown in the following example:

```
{
    "total": <integer>,
    "limit": <integer>,
    "content": [
        {
            "name": "<deviceName>",
            "address": "<deviceAddress>"
            "profile": "<profileName",
            "last-seen": "<date>T<time>Z",
            "type": "<addressType>"
        },
       ...
    ]
}
```

The table that follows describes the fields in the response body:

Field	Description
“total”	Integer indicating the maximum number of devices which are currently associated with the specified subscriber account. **NOTE:** See the "Limiting Retrieved Results for GET Requests" section in Chapter 1, "Introduction to the SSM RESTful API" for information about how to set the limit field for **GET** requests.
“limit”	Integer indicating the maximum number of devices the API response can include. **NOTE:** See the "Limiting Retrieved Results for GET Requests" section in Chapter 1, “Introduction to the SSM Restful API," for information about how to set the limit field for **GET** requests.
“name”	Identifies a device associated with the specified subscriber account.
"address"	Displays the MSISDN (Mobile Station International Subscriber Directory Number) or MAC address assigned to the device.
“profile”	Identifies the profile that is attached to the specified device. **NOTE:** In the SIA SMB subscriber portal application, profiles are called “device groups.”
"last-seen"	Indicates the last time the specified device accessed the network, where the date and time is expressed in the following format: `<year-month-day>T<hour:min:seconds><z-time>Z`

Field	Description
"type"	Identifies the type of the device; the following device types are possible: • roaming • hardware-address • line

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows retrieves a list of devices for a subscriber account called **joe.coffee**:

```
SSM> GET /account/paul.perdevice/device/
{
    "total": 6,
    "limit": 0,
    "content": [
        {
            "name": "jane-tablet2-OS12",
            "address": "00:0a:33:3d:40:20",
            "profile": "kids",
            "last-seen": null,
            "type": "hardware-address"
        },
        {
            "name": "joe-tablet-A2161-OS13",
            "address": "00:0a:95:9d:64:17",
            "profile": "teenagers",
            "last-seen": null,
            "type": "hardware-address"
        },
        {
            "name": "Guest phone",
            "address": "00:16:6f:25:1e:50",
            "profile": "adult-guests",
            "last-seen": "2024-06-27T21:07:24Z",
            "type": "hardware-address"
        },
        {
            "name": "kids-tablet-os12",
            "address": "1773938167 4",
            "profile": "kids",
            "last-seen": null
            "type": "line"
        },
        {
            "name": "Joe's phone",
            "address": "78228566993",
```

```
            "profile": "teenagers",
            "last-seen": null,
            "type": "line"
        },
        {
            "name": "Jane's phone",
            "address": "97555983510",
            "profile": "kids",
            "last-seen": null,
            "type": "line"
        }
    ]
}
```

Remove a Specific Device from a Subscriber Account

Use the following **DELETE** API to remove a device from a subscriber account:

```
SSM> DELETE  /account/<accountId>/device/<deviceAddress>
```

Fields

Field	Description
<accountId>	Required. Identifies the subscriber account from which to remove the device. Replace the ***<accountId>*** argument with the desired account identifier.
<deviceAddress>	Required. Specifies the address that is associated with the device you want to remove from a subscriber account. Replace the ***<deviceAddress>*** argument with the desired device address.

Expected Response

The SSM deletes the specified device as requested.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows deletes a device (associated with the MAC address **00:0a:95:9d:64:16**) from a subscriber account called **joe.coffee**:

```
SSM> DELETE /account/joe.coffee/device/00:0a:95:9d:64:16
```

Chapter 10: Logical Device APIs

Before configuring logical devices, you need to understand what the terms “logical device” and “device identifier” refer to.

A logical device is a name given to a list of device identifiers that are associated with a single device. A device identifier is the network ID for a device as logged by CacheServe. A device can have one of the following device identifiers, depending on whether the device connects to the network over a mobile or fixed line:

- MSISDN—Devices that use a SIM card for a cellular (mobile line) connection use the associated MSISDN as a device identifier. In GSM 1800 standard, it represents Country Code + National Destination Code + Subscriber Number, for example 13905559442. The MSISDN is unique to the subscriber; however, the SIM card can be switched between devices. SIA SMB subscribers can associate a device with any user that has the SIM card.
- MAC address—A MAC address identifies a device that uses a wireless or Ethernet port to connect to a network over a fixed line. The MAC address is unique to the device and is assigned to the device's network module by the device manufacturer. SIA SMB subscribers can associate the device with a known device user.

In converged networks, an individual device can have two instances: one for the mobile line (which uses the MSISDN as the device identifier), and another instance for the fixed line (which uses the MAC address as the device identifier). If a device has two instances (two device identifiers) in your SIA SMB service, you can merge those instances under a single name, which is known as a logical device. A logical device represents a device that can access a converged network through fixed and mobile lines.

Merging device instances combines all the changes made to a device that exists in two different device groups or that has different users operating the device. By merging the device instances, you avoid overlapping device data. Applying a profile (device group) to a logical device seamlessly enforces the device group rules on that device, regardless of whether the device is connected through a fixed or mobile line. The SIA SMB subscriber application alerts subscribers when two device instances can be merged.

Consider the following rules and restrictions before configuring logical devices:

- The fixed and mobile lines need to be provisioned before you merge the fixed and mobile line instances associated with a device.
- A device can have one or both of the following device identifiers:
    - MAC address
    - MSISDN

  If a device has two device identifiers (a MAC address and an MSISDN), we recommend merging those identifiers under one logical device name.

- The logical device name and device identifier need to be unique within your system.
- You can associate a logical device with one device group only.
- By default, device groups support an unlimited number of devices per device group. If desired, you can limit the number of devices an individual device group can apply to. A device group cannot apply to more devices than the limit allows. Be aware that half of the limit is reserved for logical devices that have two identifiers (instances).
- If you have SIA Remote enabled in your system, be aware that you cannot use the logical device APIs to add, update, or remove SIA Remote devices. With SIA Remote, you can add devices through the SIA Remote client only; the APIs for managing and deleting SIA Remote devices are described in Chapter 11, "SIA Remote Service APIs." In addition, because SIA remote devices are always represented by a single device identifier, there is no need to merge SIA Remote device identifiers.

This chapter describes the APIs for managing logical devices that belong to an Akamai Secure Internet Access (SIA) subscriber account. The APIs in this chapter describe how to perform the following tasks:

- Add a logical device to a subscriber account
- Update the configuration for a logical device
- Update the device and user names associated with a logical device
- Display information about the logical devices associated with a specific subscriber account
- Retrieve configuration details for a logical device
- Delete a logical device

Add a Logical Device to a Subscriber Account

Use the following **POST** API to add a logical device to a subscriber account:

```
SSM> POST /account/<accountId>/logical-device
{
    "add": [
        {
            "name":"<deviceName>",
            "profile":"<profileName>",
            "identifiers":["<deviceId>", …],
        },
        …
    ]
}
{
    "remove": [
        {
            "name":"<deviceName>"
        },
```

```
        …
    ]
}
{
    "update": [
        {
            "name":"<deviceName>",
            "profile":"<profileName>",
            "identifiers":["<deviceId>", …],
        },
        …
    ]
}
```

Consider the following information when using this API:

- If you are removing a logical device from a subscriber account, the **"name"** construct is the only required construct, and the **"remove"** option removes the device from the subscriber account:

```
SSM>  POST /account/joe.coffee/logical-device
{"remove":[{"name":"myphone-09"}]}
{
    "succeeded": [
        {
            "name": "myphone-09",
            "last-seen": null,
            "device-details": [
                {
                    "identifier": "8005552616",
                    "name": "myphone-09",
                    "type": "hardware-address"
                }
            ]
        }
    ],
    "status": "success"
}
```

- You can use the **"update"** option to update details for the logical device itself or the device instances that belong to the logical device. The **"update"** option is also useful for replacing the device instances that belong to a logical device. The example that follows demonstrates what happens when the API updates the device instances that belong to a logical device called **myphone-09**. In this example, the API replaces two identifiers (**8005552617** and **8005552618**) with the identifier **8005552619**. In other words, the API removes the existing device instances from the logical device, and adds a new device instance to the logical device. The identifiers are highlighted in the API responses:

The following **GET** API shows output from the **myphone-09** device, where the device has two identifiers (**8005552617** and **8005552618**):

```
SSM> GET /account/joe.coffee/logical-device/myphone-09
```

```
{
    "name": "myphone-09",
    "identifiers": [
        "8005552617",
        "8005552618"
    ],
    "profile": "sb_group_visitor",
    "last-seen": null,
    "device-details": [
        {
            "identifier": "8005552617",
            "name": "myphone-09",
            "type": "hardware-address"
        },
        {
            "identifier": "8005552618",
            "name": "myphone-09",
            "type": "hardware-address"
        }
    ]
}
```

The following **POST** API updates the **myphone-09** logical device, replacing the two identifiers with a single, new identifier (**8005552619**):

```
SSM> POST /account/joe.coffee/logical-device {"update":[{"name":
"myphone-09", "identifiers":[8005552619]}]}
```

```
{
    "succeeded": [
        {
            "name": "myphone-09",
            "identifiers": [
                "8005552619"
            ],
            "last-seen": null,
            "device-details": [
                {
                    "identifier": "8005552619",
                    "name": "myphone-09",
                    "type": "hardware-address"
                }
            ]
        }
    ],
    "status": "success"
}
```

The following **GET** API response verifies the **myphone-09** is updated to have the identifier **8005552619** (and no other identifiers):

```
SSM> GET /account/joe.coffee/logical-device/myphone-09
{
    "name": "myphone-09",
    "identifiers": [
        "8005552619"
    ],
    "profile": "sb_group_visitor",
    "last-seen": null,
    "device-details": [
        {
            "identifier": "8005552619",
            "name": "myphone-09",
            "type": "hardware-address"
        }
    ]
}
```

Fields

Field	Description
<accountId>	Required. Identifies the subscriber account for which to add, update, or remove a logical device. Replace the ***<accountId>*** with the desired account identifier.
"name"	Required. Specifies a logical device name. Replace the ***<deviceName>*** argument with a string that identifies the logical device you are adding, updating, or removing. The name can be a unique name of your choosing or a combination of the username, device model, and operating system (OS) that is installed on the device. **NOTE:** If a device name includes one or more spaces, replace each space with the **%20** notation when specifying the ***<deviceName>*** argument. For example, if the device name is **my iphone**, replace the ***<deviceName>*** argument with **my%20iphone**.
"profile"	Optional. Identifies the profile to attach to the logical device; replace the ***<profileName>*** argument with the name of the desired profile. **NOTE:** The profile name is case-sensitive. If the API call does not include the **"profile"** construct, the SIA SMB service adds the logical device to the **sb_group_visitor** profile by default. **NOTE:** In the SIA SMB subscriber portal application, profiles are called “device groups.”

Field	Description
"identifiers"	Required if you are adding or updating a logical device. Specifies one or more identifiers that are associated with the device instances you are adding to the logical device. Replace the ***<deviceId>*** argument with a value that is used to identify the device. To add multiple identifiers, separate each with a comma as shown in the following example: **`"identifiers":["8005552618, 8005552640"]`**

Expected Response

The logical device is added to, updated for, or removed from the specified subscriber account, and the API response includes general information about the logical device as shown in the following example:

```
{
    "succeeded": [
        {
            "name": "<deviceName>",
            "identifiers": [
                "<deviceId>", …
            ],
            "profile": "<profileName>",
            "last-seen": <date-and-time>,
            "device-details": [
                {
                    "identifier": "<deviceId>",
                    "name": "<deviceName>",
                    "type": "<deviceType>"
                }
            ]
        }
    ],
    "status": "success"
}
```

The table that follows describes the fields displayed in the API response:

Field	Description
"name"	Identifies the logical device the API added, updated, or removed.
"identifiers"	Lists the identifiers of the device instances in the affected logical device.
"profile"	Identifies the profile that is attached to the affected logical device. **NOTE:** In the SIA SMB subscriber portal application, profiles are called “device groups.”
"last-seen"	Indicates the last time the specified logical device accessed the network.
"device-details"	Displays the following information for each device instance (identifier) that is associated with the specified logical device: ● "identifier"—Specifies the identifier that is associated with the device instance for which data is displayed. ● "name"—Specifies the name that is assigned to the device instance for which data is displayed. ● "type"—Identifies the type of address that is associated with the specified device instance. The following values are possible ○ “hardware-address”—Indicates the device instance accesses the network through a fixed line. ○ “line”—Indicates the device instance accesses the network through a mobile line. ○ “roaming”—Indicates the device instance accesses the network through SIA Remote application. **NOTE:** The “device-details” data is for visualization purposes only. Although it is possible to do so, we do not recommend sending “device-details” data in any API request as sending such data can cause strange behavior in your deployment.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Examples

The example that follows demonstrates how to add a logical device to a subscriber account; this example creates a logical device to a subscriber account called **joe.coffee**. This example names the logical device **800-555-2640-40** and adds a single device instance (with the identifier **8005552640**) to that logical device:

```
SSM> POST /account/joe.coffee/logical-device
{"add":[{"name":"800-555-2640-40", "identifiers":["8005552640"]}]}
{
    "succeeded": [
        {
            "name": "800-555-2640-40",
            "identifiers": [
                "8005552640"
            ],
            "last-seen": null,
            "device-details": [
                {
                    "identifier": "8005552640",
                    "name": "800-555-2640-40",
                    "type": "hardware-address"
                }
            ]
        }
    ],
    "status": "success"
}
```

The example that follows adds a logical device to a subscriber account called **joe.coffee**. This logical device has a single device instance (which is identified by the identifier **8005552678**). This example assigns performs the following configuration for the new logical device:

- Assigns a profile called **sb_group_visitor** to the logical device.
- Assigns the name **800-555-2678-19** to the logical device, where **8005552678** is the identifier assigned to the device instance.

```
SSM> POST /account/joe.coffee/logical-device
{"add":[{"name":"800-555-2678-19","profile":"sb_group_visitor",
"identifiers":["8005552678"]}]}
{
    "succeeded": [
        {
            "name": "800-555-2678-19",
            "identifiers": [
                "8005552618"
```

```
            ],
            "profile": "sb_group_visitor",
            "last-seen": null,
            "device-details": [
                {
                    "identifier": "8005552678",
                    "name": "800-555-2678-19",
                    "type": "hardware-address"
                }
            ]
        }
    ],
    "status": "success"
}
```

The example that follows demonstrates how to add and remove logical devices to and from a subscriber account in one API call. This example adds a logical device called **800-555-2650-50** from a subscriber account called **art.gallery** and removes a device called **800-555-2640-40** from that same account:

```
SSM> POST /account/art.gallery/logical-device
{"add":[{"name":"800-555-2650-50", "identifiers":["8005552650"]}],
"remove":[{"name":"800-555-2640-40", "identifiers":["8005552640"]}]}
{
    "succeeded": [
        {
            "name": "800-555-2640-40",
            "identifiers": [
                "8005552640"
            ],
            "last-seen": null,
            "device-details": [
                {
                    "identifier": "8005552640",
                    "name": "800-555-2640-40",
                    "type": "hardware-address"
                }
            ]
```

```
        },
        {
            "name": "800-555-2650-50",
            "identifiers": [
                "8005552650"
            ],
            "last-seen": null,
            "device-details": [
                {
                    "identifier": "8005552650",
                    "name": "800-555-2650-50",
                    "type": "hardware-address"
                }
            ]
        }
    ],
    "status": "success"
}
```

Update the Configuration for a Logical Device

Use the following **PUT** API to update the configuration for a logical device:

```
SSM> PUT /account/<accountId>/logical-device/<deviceName>
{
    "name":"<deviceName>",
    "profile":"<profileName>",
    "identifiers":["<deviceId>", …],

}
```

Fields

Field	Description
<*accountId*>	Required. Identifies the subscriber account for which to update a logical device. Replace the **<*accountId*>** with the desired account identifier.
<*deviceName*>	Required. Identifies the logical device you are updating. Replace the **<*deviceName*>** argument with a string that identifies the device. **NOTE:** If a device name includes one or more spaces, replace each space with the **%20** notation when specifying the **<*deviceName*>** argument. For example, if the device name is **my iphone**, replace the **<*deviceName*>** argument with **my%20iphone**.
"name"	Required. Specifies a name for the logical device you are updating. Use this field to change the name of a logical device. Replace the **<*deviceName*>** argument with the desired name. **NOTE:** If a device name includes one or more spaces, replace each space with the **%20** notation when specifying the **<*deviceName*>** argument. For example, if the device name is **my iphone**, replace the **<*deviceName*>** argument with **my%20iphone**.
"profile"	Optional. Identifies a profile to attach to the logical device; the profile you specify replaces the profile that is currently attached to the logical device. Replace the **<*profileName*>** argument with the name of the desired profile. **NOTE:** The profile name is case-sensitive. **NOTE:** In the SIA SMB subscriber portal application, profiles are called “device groups.”
"identifiers"	Required. Specifies one or more identifiers that are associated with the device instances you are updating. Replace the **<*deviceId*>** argument with a value that is used to identify the device instance. To add multiple identifiers, separate each with a comma as shown in the following example: **`"identifiers":["8005552618, 8005552640"]`**

Expected Response

A successful response includes HTTP code 200, along with the following JSON response body indicating the logical device updated as specified by the API request:

```
{
    "succeeded": [
        {
            "name": "<deviceName>",
            "identifiers": [
```

```
                "<deviceId>", …
            ],
            "profile": "<profileName>",
            "last-seen": <date-and-time>,
            "device-details": [
                {
                    "identifier": "<deviceId>",
                    "name": "<deviceName>",
                    "type": "<addressType>"
                }
            ]
        }
    ],
    "status": "success"
}
```

The table that follows describes the fields displayed in the API response:

Field	Description
"name"	Identifies the logical device the API updated.
"identifiers"	Lists the device identifiers that are included in the specified logical device.
"profile"	Identifies the profile (device group) that is attached to the specified logical device.
"last-seen"	Indicates the last time the specified logical device accessed the network.

Field	Description
"device-details"	Displays the following information for each device instance that is associated with the device: ● "identifier"—Device identifier that is associated with the device.. ● "name"—Identifies the device instance. ● "type"—Identifies the type of address that is associated with the device instance. The following address types are possible: ○ hardware-address—Indicates the device instance accesses the network through a fixed line. ○ line—Indicates the device instance accesses the network through a mobile line ○ roaming—Indicates the device instance accesses the network through SIA Remote application. **NOTE:** The “device-details” data is for visualization purposes only. Although it is possible to do so, we do not recommend sending “device-details” data in any API request as sending such data can cause strange behavior in your deployment.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows updates the device **name** for a logical device (called **800-555-2640-40**) that is associated with a subscriber account called **joe.coffee**:

```
SSM> PUT /account/joe.coffee/logical-device/800-555-2640-40
{"name":"my-tabletv10","identifiers":["8005552640"]}
{
    "succeeded": [
        {
            "name": "my-tabletv10",
            "identifiers": [
                "8005552640"
            ],
            "profile": "sb_group_visitor",
            "last-seen": null,
            "device-details": [
                {
                    "identifier": "8005552640",
                    "name": "800-555-2640-40",
                    "type": "hardware-address"
                }
```

```
                ]
            }
        ],
        "status": "success"
}
```

Update the Device and User Names Associated with a Logical Device

Use the following **PATCH** API to update the configuration of a particular logical device:

```
SSM> PATCH /account/<accountId>/logical-device/<deviceName>

 {
    "name":"<deviceName>",
    "profile":"<profileName>",
    "identifiers":["<deviceId>", …]
}
```

Fields

Field	Description
<accountId>	Required. Identifies the subscriber account for which to update a logical device. Replace the **<accountId>** with the desired account identifier.
"name"	Required. Identifies the logical device you want to update. Replace the **<deviceName>** argument with the name of the logical device. **NOTE:** If a device name includes one or more spaces, replace each space with the **%20** notation when specifying the **<deviceName>** argument. For example, if the device name is **my iphone**, replace the **<deviceName>** argument with **my%20iphone**.
"profile"	Optional. Identifies a profile to attach to the logical device; the profile you specify replaces the profile that is currently attached to the logical device. Replace the **<profileName>** argument with the name of the desired profile. **NOTE:** The profile name is case-sensitive. **NOTE:** In the SIA SMB subscriber portal application, profiles are called “device groups.”

Field	Description
"identifiers"	Required. Specifies one or more device identifiers that are associated with the device instances you are updating. Replace the ***<deviceId>*** argument with a value that is used to identify the device instance. To add multiple device identifiers, separate each identifier with a comma as shown in the following example: **`"identifiers":["8005552618, 8005552640"]`**

Expected Response

A successful response includes HTTP code 200, along with the following JSON response body indicating the logical device is updated as specified by the API request:

```
{
    "succeeded": [
        {
            "name": "<deviceName>",
            "identifiers": [
                "<deviceId>", …
            ],
            "profile": "<profileName>",
            "last-seen": <date-and-time>,
            "device-details": [
                {
                    "identifier": "<deviceId>",
                    "name": "<deviceName>",
                    "type": "<deviceType>"
                }
            ]
        }
    ],
    "status": "success"
}
```

The table that follows describes the fields displayed in the API response:

Field	Description
"name"	Name of the logical device the API updated.
"identifiers"	Lists the device identifiers that are associated with the specified logical device.
"profile"	Identifies the profile that is attached to the specified logical device. **NOTE:** In the SIA SMB subscriber portal application, profiles are called “device groups.”
"last-seen"	Indicates the last time the specified logical device accessed the network. If the logical device contains more than one device instance, the "last-seen" field displays the last time one of those logical devices accessed the network.
"device-details"	Displays the following information for each device identifier that is associated with the device: ● "identifier"—Device identifier that is associated with the device instance. ● "name"—Name that identifies the device instance. A device instance can share a name with the logical device or have a user-configured alternative name that distinguishes the device instance from other devices instances in the logical device. ● "type"—Identifies the type of address that is associated with the device instance the API updated. The following address types are possible: ○ hardware-address—Indicates the device instance accesses the network through a fixed line. ○ line—Indicates the device instance accesses the network through a mobile line. ○ roaming—Indicates the device instance accesses the network through SIA Remote application. **NOTE:** The “device-details” data is for visualization purposes only. Although it is possible to do so, we do not recommend sending “device-details” data in any API request as sending such data can cause strange behavior in your deployment.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows updates a logical device (called **jane-doe-phone11-A2161-OS13**) that is associated with a subscriber account called **joe.coffee**:

```
SSM> PATCH /account/joe.coffee/logical-device/jane-doe-Phone11-A2161-OS13
{"name":"jane-doe-phone11-A2161-OS13","identifiers":["8005552808"],
"profile":"sb_group_visitor"}]}
{
    "succeeded": [
        {
            "name": "jane-doe-phone11-A2161-OS13",
            "identifiers": [
                "8005552808"
            ],
            "profile": "sb_group_visitor",
            "last-seen": null,
            "device-details": [
                {
                    "identifier": "8005552808",
                    "name": "jane-doe-phone11-A2161-IOS13",
                    "type": "hardware-address"
                }
            ]
        }
    ],
    "status": "success"
}
```

Display Information About the Logical Devices Associated with a Specific Subscriber Account

Use the following **GET** API to display detailed information for all logical devices associated with a specific subscriber account:

```
SSM> GET /account/<accountId>/logical-device
```

Fields

Field	Description
<accountId>	Required. Specifies the subscriber account for which to display logical device details. Replace the ***<accountId>*** argument with the desired subscriber account identifier.

Expected Response

The API returns detailed information about all logical devices associated with the specified subscriber account, including data for device instances that are contained in each logical device.

```
{
  "total": <integer>,
  "limit": <integer>,
  "content": [
    {
      "name": "<deviceName>",
      "identifiers": [
        "<deviceId>", …
      ],
      "profile": "<profileName>",
      "last-seen": "<date-and-time>",
      "device-details": [
        {
          "identifier": "<deviceId>",
          "name": "<deviceName>",
          "type": "<addressType>",
          "username": "<firstName lastName>",
          "model": "<deviceModel>",
```

```
            "software-version": "<version>",
            "os": "<version>",
            "status": "<deviceStatus>"
          }
        ]
      },
  ...
}
```

NOTE: In the API response, the fields in the "device-details" section depend on the device type. While the "identifier", "name", and "type" fields are included for all device types (hardware-address, line, and roaming), the following additional fields are included for roaming devices only: "username", "model", "software-version", "os", and "status".

The table that follows describes the fields displayed in the API response:

Field	Description
"total"	Total number of logical devices that exist under the specified subscriber account.
"limit"	Indicates the maximum number of logical devices that can be included in the API response. **NOTE:** See the "Limiting Retrieved Results for GET Requests" section in Chapter 1, “Introduction to the SSM Restful API," for information about how to set the limit field for **GET** requests.
"name"	Identifies the logical device for which information is displayed.
"identifiers"	Lists the device identifiers that are associated with the logical device.
"profile"	Identifies the profile attached to the specified logical device. **NOTE:** In the SIA SMB subscriber portal application, profiles are called “device groups.”

Field	Description
"last-seen"	Indicates the last time the logical device accessed the network. Express the date and time in the following format: **<*year-month-day*>T<*hour:min:seconds*><*z-time*>Z** The following example expresses the **"last-seen"** construct for a logical device that last accessed the network on October **30**, **2024** at **8:43** PM (and **35** seconds), with a Z-time offset of **563**: **"last-seen": "2024-10-30T20:43:35.563Z"** **NOTE:** This field is for information purposes and, if a logical device contains multiple device instances (device identifiers), can refer to either device instance.
"device-details"	Displays data for each device instance contained in the specified logical device. The "device-details" Fields table describes the fields displayed for each device instance.

The table that follows describes the fields the API response includes for each device instance in a logical device:

"device-details" Fields

Field	Description
"identifier"	Displays the device identifier that is associated with the device instance for which data is displayed.
"name"	Displays the name associated with the device instance. A device instance can share a name with the logical device or have a user-configured alternative name that distinguishes the device instance from the other device instances in the logical device.
"type"	Indicates the type of address that is associated with the device instance. A device instance can have one of the following address types: • hardware-address—Indicates the device instance accesses the network through a fixed line. • line—Indicates the device instance accesses the network through a mobile line. • roaming—Indicates the device instance accesses the network through a SIA Remote application.
"username"	Displays the first and last name of the device user that is associated with the specified device instance. **NOTE:** The API response includes this field for roaming devices only.
"model"	Identifies the model type for device instance. **NOTE:** The API response includes this field for roaming devices only.

Field	Description
"software-version"	Identifies the software version that is installed on the specified device instance. **NOTE:** The API response includes this field for roaming devices only.
"os"	Identifies the operating system (OS) installed on the specified device instance. **NOTE:** The API response includes this field for roaming devices only.
"status"	Indicates whether the device instance is currently enabled or disabled. **NOTE:** The API response includes this field for roaming devices only.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows displays detailed information for the logical devices that are associated with a subscriber account called **joe.coffee**:

```
SSM> GET /account/joe.coffee/logical-device
{
  "total": 3,
  "limit": 0,
  "content": [
    {
      "name": "jdoe - Phone9,3",
      "identifiers": [
        "fec1255a-613e-4f26-b7b6-2c2121ac73eb"
      ],
      "profile": "sb_group_visitor",
      "last-seen": "2024-10-30T20:43:35.563Z",
      "device-details": [
        {
          "identifier": "fec1255a-613e-4f26-b7b6-2c2121ac73eb",
          "name": "jdoe - Phone9,3",
          "type": "roaming",
          "username": "jane doe",
          "model": "Phone9,3",
          "software-version": "ios-1.0.0",
          "os": "iOS Version 13.5.1 (Build 17F80)",
```

```
          "status": "enabled"
        }
      ]
    },
    {
      "name": "ssmith - phone",
      "identifiers": [
        "8005552808"
      ],
      "profile": "sb_group_visitor",
      "last-seen": "2024-10-29T23:04:21.827Z",
      "device-details": [
        {
          "identifier": "8005552808",
          "name": "ssmith - phone
          "type": "hardware-address"
        }
      ]
    },
    {
      "name": "msmith - tablet13",
      "identifiers": [
        "74049349999"
      ],
      "profile": "sb_group_visitor",
      "last-seen": "2024-09-30T01:14:45.143Z",
      "device-details": [
        {
          "identifier": "74049349999",
          "name": "msmith - tablet13",
          "type": "line",
        }
      ]
    }
  ]
}
```

Retrieve Configuration Details for a Logical Device

Use the following **GET** API to retrieve the configuration for a particular logical device:

```
SSM> GET  /account/<accountId>/logical-device/<deviceName>
```

Fields

Field	Description
<accountId>	Required. Specifies the name of the subscriber account that is associated with the logical device for which to retrieve configuration details. Replace the ***<accountId>*** argument with the string that identifies the subscriber account.
<deviceName>	Required. Specifies the name of the logical device for which to retrieve configuration details. Replace the ***<deviceName>*** argument with the string that identifies the logical device. **NOTE:** If a logical device name includes one or more spaces, replace each space with the **%20** notation when specifying the ***<deviceName>*** argument. For example, if the device name is my iphone, replace the ***<deviceName>*** argument with **my%20iphone**.

Expected Response

A successful response returns an HTTP code 200 with a JSON based response body displaying configuration details for the specified logical device as shown in the following example:

```
{
   "name": "<deviceName>",
   "identifiers": [
       "<deviceId>", …
   ],
   "profile": "<profileName>",
   "last-seen": <date-and-time>,
   "device-details": [
       {
           "identifier": "<deviceId>",
           "name": "<firstName lastName>",
           "type": "<deviceType>"
       }
   ]
}
```

The table that follows describes the fields in the response body:

Field	Description
"name"	Identifies the logical device for which configuration details are displayed.
"identifiers"	Displays the device identifiers that are associated with the device instances the logical device contains.
“profile”	Identifies the profile attached to the specified logical device. **NOTE:** In the SIA SMB subscriber portal application, profiles are called “device groups.”
“last-seen”	Indicates the last time the specified logical device accessed the network, where the date and time is expressed in the following format: `<year-month-day>T<hour:min:seconds><z-time>Z` If a logical device contains multiple device instances (device identifiers), the “last-seen” field displays the last time one of those device instances accessed the network.
“device-details”	Displays the following information for each device identifier that is associated with the device: ● "identifier"—Device identifier that is associated with the device instance for which information is displayed. ● "name"—Displays the name that is assigned to the device instance. A device instance can share a name with the logical device or have a user-configured alternative name that distinguishes the device instance from the other device instances in the logical device. ● "type"—Identifies the type of address that is associated with the device instance the API updated. The following address types are possible: ○ hardware-address—Indicates the device instance accesses the network through a fixed line. ○ line—Indicates the device instance accesses the network through a mobile line. ○ roaming—Indicates the device instance accesses the network through SIA Remote application.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows retrieves configuration details for a logical device (called **jane-doe-Phone11-A2161-IOS13**) that is associated with a subscriber account called **joe.coffee**:

```
SSM> GET /account/joe.coffee/logical-device/jane-doe-Phone11-A2161-IOS13
{
   "name": "jane-doe-Phone11-A2161-IOS13",
   "identifiers": [
       "00:B0:00:00:00:05"
   ],
   "profile": "Default",
   "last-seen": null,
   "device-details": [
       {
           "identifier": "00:B0:00:00:00:05",
           "name": "phone",
           "type": "hardware-address"
       }
   ]
}
```

Delete a Logical Device

Use the following **DELETE** API to remove a logical device from a subscriber account:

```
SSM> DELETE /account/<accountId>/logical-device/<deviceName>
```

Fields

Field	Description
<accountId>	Required. Identifies the subscriber account that is associated with the logical device you want to delete. Replace the ***<accountId>*** with the string that identifies the subscriber.
<deviceName>	Required. Identifies the logical device you want to delete. Replace the ***<deviceName>*** argument with the string that identifies the logical device. **NOTE:** If a device name includes one or more spaces, replace each space with the **%20** notation when specifying the ***<deviceName>*** argument. For example, if the device name is my iphone, replace the ***<deviceName>*** argument with **my%20iphone**.

Expected Response

The SSM removes the specified logical device from the specific subscriber account and returns an HTTP code 200.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows deletes a logical device (called **jane-doe-Phone11-A2161-IOS13**) from a subscriber account called **joe.coffee**:

```
SSM> DELETE  /account/joe.coffee/logical-device/jane-doe-Phone11-A2161-IOS13
```

Chapter 11: SIA Remote Service APIs

The Akamai Secure Internet Access (SIA) Remote client provides DNS-based security to protect data assets and reputations, block security threats, and filter inappropriate content.

Mobile devices use deep links to access the SIA Remote client application. After integrating deep link generation on your subscriber support (as described in the *SIA 19.2.1 Products: AMC-X Overview and Deployment Design* and *SIA 19.2.1 Products: Installation Using AMC-X* manuals), you use SSM Remote Service APIs to generate a deep link connection for your SIA Remote service subscribers from your support portal. The SIA Remote service supports the following types of deep links:

- Time-bound (expiring) deep links, which require a time-to-live (TTL) timestamp and no longer work after the TTL passes.
- Non-expiring (fixed) deep links that do not expire. We recommend using non-expiring deep links in MDM environments so that you do not need to regenerate the deep link and update the MDM configuration every time the link expires.

  Caution! Because non-expiring deep links contain secret data, do not use non-expiring deep links in MDM deployments where an end user can easily access that data. Because non-expiring deep links do not expire, the data embedded in the links is not encrypted.

The CAS agent uses entitlement codes to authenticate deep links. Each deep link type has its own entitlement code; in other words, time-bound deep links have entitlement codes that are separate from non-expiring deep link entitlement codes.

NOTE: Because time-bound and non-expiring deep links have different entitlement codes, rotate expiring and non-expiring entitlement codes independently from one another. In other words, rotate the non-expiring entitlement code independently of any existing expiring entitlement.

This chapter describes the Representational State Transfer (RESTful) Subscriber Services Manager (SSM) APIs for activating and managing the SIA Remote service on SIA SMB subscriber accounts in deployments that are licensed and configured for SIA Remote service support.

Prerequisites

Before using the APIs described in this chapter, ensure your system meets the following prerequisites:

- Your system is licensed for the SIA Remote service.
- Your system is configured to support Akamai's SIA Remote service as described in the following documents:
    - *SIA 19.2.1 Products: Installation Using AMC-X*
    - *SIA 19.2.1 Products: AMC-X Overview and Deployment Design*

The APIs in this chapter describe how to perform the following tasks:

- Configure SIA Remote options for a subscriber account.
- Activate the SIA Remote service for a subscribe account.
- Generate a time-bound deep link for an SIA Remote service subscriber client.
- Generate a non-expiring deep link for an SIA Remote service subscriber client.
- Update SIA Remote options for a subscriber account.
- Update the device and user names associated with an SIA Remote service device.
- Rotate the entitlement code for time-bound deep links.
- Rotate the entitlement code for non-expiring deep links.
- Retrieve the SIA Remote service configuration for a subscriber account.
- Retrieve configuration details for an SIA Remote service subscriber device.
- Remove an SIA Remote service object from the subscriber account.
- Unenroll a device from the SIA Remote service.
- Deactivate the SIA Remote service for a subscriber account.

Configure SIA Remote Options for a Subscriber Account

Use the following **POST** API to configure the number of devices allowed on an SIA Remote service subscriber's account and, optionally, whether the subscriber devices can access a specified internal domain and whether the subscriber account supports non-expiring deep links:

```
SSM> POST /account/<accountId>/spson

{
      "allowed-devices": <number-of-devices>,
      "mdm-deeplink-enabled": <boolean>,
      "internal-domains":["<domain>, …"]
}
```

NOTE: While local domains apply to all subscribers in an ISP, internal domains apply to a specific subscriber only. The **POST /account/<*accountId*>/spson** API configures internal domain access for all subscribers in an ISP.

Fields

Field	Description
<***accountId***>	Required. Specifies the subscriber for which to add SIA Remote service-enabled devices; replace the **<*accountId*>** argument with the desired subscriber.
"allowed-devices"	Required. Specifies the maximum number of SIA Remote service devices allowed for the specified subscriber account. Replace the **<*number-of-devices*>** argument with an integer in the range assigned to your ISP. **NOTE:** This field is valuable for ISPs offering tiered service support (for example, one price tier for 5 devices, a higher price tier for 10 devices and so forth).
"mdm-deeplink-enabled"	Optional. Enables and disables non-expiring deep link support on the subscriber account. Replace the **<*boolean*>** argument with one of the following values: • **true**—Enables non-expiring deep link support on the specified subscriber account and generates an entitlement code for the deep links. • **false**—Disables non-expiring deep link support on the specified subscriber account. **NOTE:** We recommend enabling non-expiring deep links for subscribers in MDM environments so that you do not need to regenerate the deep link and update the MDM configuration every time the link expires. The default value is **false**.
"internal-domains"	Optional. Allow devices belonging to the specified subscriber account to access to your corporation's internal domains. Replace the **<*domain*>** argument with an internal domain name; you can specify multiple domains, separating each domain with a comma. Be aware this field has a maximum limit of 999 characters. **NOTE:** Use the **GET /account/<*accountId*>/spson** API to verify internal domain access for a subscriber; see the "Retrieve the SIA Remote Service Configuration for a Subscriber Account" section details.

Expected Response

The SSM sets the number of devices allowed on the specified subscriber account as requested.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Examples

The example that follows configures the number of SIA Remote service devices allowed for a subscriber account called **sub123** to be **5**. Because the API does not include the **"mdm-deeplink-enabled": true** construct, the account supports time-bound deep links only:

```
SSM> POST /account/sub123/spson {"allowed-devices": 5}
```

The example that follows enables non-expiring deep links and generates an entitlement code on a subscriber account called **joe.coffee** and configures the account to support up to four (**4**) SIA Remote service devices:

```
SSM> POST /account/joe.coffee/spson {"allowed-devices": 4,
"mdm-deeplink-enabled": true}
```

The example that follows configures the number of SIA Remote service devices allowed for the subscriber account called **sub123** to be **5** and enables access to the internal domain **akamai.com**:

```
SSM> POST /account/sub123/spson {"allowed-devices": 10
"internal-domains":["akamai.com"]}
```

Activate the SIA Remote Service for a Subscriber Account

Use the following **POST** API to activate the SIA Remote service for a specific subscriber account:

```
SSM> POST /control/<accountId>/service
{
    "service":"spson"
}
```

NOTE: The spson (SIA Remote) object needs to exist in your system before you enable the **spson** (SIA Remote) service.

After activating the SIA Remote service on a subscriber account, you can control (enable and disable) the enforcement of the SIA Remote service on the subscriber's account. See the "Differences Between Activating and Enabling a Service" section in Chapter 2, "Subscriber Account APIs," for more information about activating and enabling services on a subscriber account.

NOTE: Before activating the SIA Remote service for a subscriber account, you need to configure the number of devices allowed on the account and, optionally, internal domain access as described in the "Configure SIA Remote Options for a Subscriber Account" section.

Fields

Field	Description
<accountId>	Required. String that identifies the subscriber account for which to activate the SIA Remote service.

Expected Response

The SSM activates the SIA Remote service for the subscriber.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Examples

The example that follows enables the SIA Remote service for the subscriber called **sub123**:

```
SSM> POST /control/sub123/service {"service":"spson"}
```

Generate a Time-bound Deep Link for an SIA Remote Service Subscriber Client

Use the following **POST** API to generate a time-bound deep link connection for an SIA Remote service subscriber; the deep link connection installs the client application on the subscriber's device and registers the device:

```
SSM> POST /account/<accountId>/spson/deeplink
```

After generating the deep link connection, an account administrator needs to provide the generated link to the appropriate subscriber device (the device to add to the SIA Remote service). The administrator can send the deep link connection through a designated communication channel (for example, an email, a text message, and so forth). After receiving the deep link connection on the appropriate device, the subscriber clicks that link to download and install the SIA Remote client application on the device (if the application is not already installed) and is prompted to enter a first and last name to register the device.

Consider the following information before configuring deep links:

- You need to implement client clean-up before deploying a new time-bound deep link.
- Do not store deep links in any configuration file or logs. If a client needs to determine whether a deep link has changed, use the hash value of the link instead of the link itself.
- Enable deep links during the setup phase when configuring your environment to support the off-network feature.

Time-bound (expiring) deep links use the following format:

```
${<ApplicationUrl>}?apidomain=${<CasApiDomain>}&regid=
${<registrationId>}&entcode=${2-$<entitlementCode>}&devid=
${<deviceId>}
```

Table 11-1: Deep Link Variables

Variable	Description
<ApplicationUrl>	Specifies the deep link URL for the application.
<CasApiDomain>	Identifies the CAS agent API endpoint that uses entitlement codes to authenticate deep links.
<registrationId>	Specifies the registration ID associated with the deep link, where the registration ID is an alphanumeric string.
<entitlementCode>	Specifies the secret entitlement code the SIA Remote service uses to authenticate client service registration requests (deep links). The entitlement code is formatted as follows: • For time-bound deep links, the entitlement code uses the format 2-${<code>}, where 2 identifies the link as a time-bound link and the <code> variable is the entitlement code for the deep link. • For non-expiring deep links, the entitlement code uses the format `3-${<code>}`, where 3 identifies the link as a non-expiring link and the *<code>* variable is the entitlement code for the deep link.
<deviceId>	The *<deviceId>* variable specifies registration information for the device requesting remote access to the network. **NOTE:** The *<deviceId>* variable is included in time-bound deep links only.

Fields

Field	Description
<accountId>	Required. Identifies the subscriber account for which to generate a time-bound deep link; replace the ***<accountId>*** argument with the name of the desired subscriber account.

Expected Response

The API returns the deep link URL along with the time-to-live (TTL) for the connection as shown in the example that follows:

```
{
   "link": "<deepLink>",
   "deeplink-ttl": <ttl>
}
```

The table that follows describes the fields in the API output:

Field	Description
"link"	The URL that generates deep link connections.
"deeplink-ttl"	The lifetime of deep links in seconds; when the TTL expires, the connection is no longer valid and the connection ends. The default TTL is **86400** seconds.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows generates a deep link connection for a subscriber called **sub123**:

```
SSM> POST /account/sub123/spson/deeplink

{
   "link":
"https://cx2-static.spson-test.akamaiapis.net/apps?apidomain=customer1.spsonc
as-preprod.akamaiap
is.net&regid=27egbjy5xcftd5cy7iyq-gi2nlj5skqvrpqidccuextl6ee-6rmczkrvbm2q4lsh
nbtt7g25wi&entcode=2-a27-160359
9266-66zwnn6gl3t44broio2q-6kvf4cfnvf2zbdfaikus5gdai5ywewvayy4if3tkae6rttl46op
cm6tfh4-x7frfbbnekccira3g6aebqu
s3m",
   "deeplink-ttl": 86400
}
```

Generate a Non-expiring Deep Link for an SIA Remote Service Subscriber Client

Use the following **POST** API to generate a non-expiring deep link connection for an SIA Remote service subscriber; the deep link connection installs the client application on the subscriber's device and registers the device:

```
SSM> POST /account/<accountId>/spson/mdm_deeplink
```

After generating the deep link connection, an account administrator needs to provide the generated link to the appropriate subscriber device (the device to add to the SIA Remote service). The administrator can send the deep link connection through a designated communication channel (for example, an email, a text message, and so forth). After receiving the deep link connection on the appropriate device, the subscriber clicks that link to download and install the SIA Remote client application on the device (if the application is not already installed) and prompts the user to enter a first and last name to register the device.

Non-expiring deep links use the following format:

```
${<ApplicationUrl>}?apidomain=${<CasApiDomain>}&regid=
${<registrationId>}&entcode=${3-${<entitlementCode>}}
```

Table 11-2: Deep Link Variables

Variable	Description
<*ApplicationUrl*>	Specifies the deep link URL for the application.
<*CasApiDomain*>	Identifies the CAS agent API endpoint that uses entitlement codes to authenticate deep links.
<*registrationId*>	Specifies the registration ID associated with the deep link, where the registration ID is an alphanumeric string.
<*entitlementCode*>	Specifies the secret entitlement code the SIA Remote service uses to authenticate client service registration requests (deep links). The entitlement code is formatted as follows: • For time-bound deep links, the entitlement code uses the format 2-${<code>}, where 2 identifies the link as a time-bound link and the <code> variable is the entitlement code for the deep link. • For non-expiring deep links, the entitlement code uses the format `3-${<code>}`, where 3 identifies the link as a non-expiring link and the <*code*> variable is the entitlement code for the deep link
<*deviceId*>	The <*deviceId*> variable specifies registration information for the device requesting remote access to the network. **NOTE:** The <*deviceId*> variable is included in time-bound deep links only.

Fields

Field	Description
<accountId>	Required. Identifies the subscriber account for which to generate a non-expiring deep link; replace the ***<accountId>*** argument with the name of the desired subscriber account.

Expected Response

The API returns the non-expiring deep link URL the subscriber can use to register a remote device as shown in the example that follows:

```
Response:
{
"link":"<deepLink>"
}
```

If the non-expiring deep link feature is not enabled for the specified subscriber account, the API returns returns a 503 error as shown in the following sample response:

```
Response:
{
    "error": {
        "status":503,
        "reason":"FEATURE_NOT_SUPPORTED",
        "message":"feature not supported: MDM Deeplink is not
supported"
    }
}
```

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Examples

The example that retrieves the non-expiring deep link for a subscriber called **joe.coffee**:

```
SSM> POST /account/joe.coffee/spson/mdm_deeplink

Response:
{
"link":"https://akacloud.spsonstatic-preprod.akamaiapis.net/apps?ap
idomain=eng7.spsoncas-preprod.akamaiapis.net&regid=4f5d34ae-11ab-4e
45-9677-ced7357d9246&entcode=3-9053589c-3f2e-486f-aad0-9bb3325d0985
"
}
```

Update SIA Remote Options for a Subscriber Account

Use the following **PATCH** API to update the maximum number of devices allowed and internal domain access for a specific subscriber account:

```
SSM> PATCH /account/<accountId>/spson
{
      "allowed-devices":<number-of-devices>,
      "internal-domains":["<domain>, …"]
      "mdm-deeplink-enabled": <boolean>
}
```

NOTE: While local domains apply to all subscribers in an ISP, internal domains apply to a specific subscriber only. The **PATCH /account/<*accountId*>/spson** API configures internal domain access for all subscribers in an ISP.

Fields

Field	Description
<***accountId***>	Required. Specifies the subscriber account for which to update the number of allowed SIA Remote service-enabled devices; replace the **<*accountId*>** argument with the desired subscriber account identifier.
"allowed-devices"	Required if updating maximum number of SIA Remote service devices allowed. Specifies the maximum number of SIA Remote service devices allowed for the specified subscriber account. Replace the **<*number-of-devices*>** argument with an integer in the range assigned to your ISP. **NOTE:** This field is valuable for ISPs offering tiered service support (for example, one price tier for 5 devices, a higher price tier for 10 devices and so forth).
"internal-domains"	Required if updating subscriber access to internal domains. Replaces the list of domains the devices associated with the subscriber account can access. Replace the **<*domain*>** argument with a desired domain name. To specify multiple domains, separate each domain with a comma as shown in the following example: **"internal-domains":[domain-1.com, domain-2.com]** See the "Considerations for Setting the 'internal-domains' Field" section that follows this table for details on setting the "internal-domains" field.

Field	Description
"mdm-deeplink-enabled"	Optional. Enables and disables the non-expiring deep link feature for a subscriber account. Replace the ***<boolean>*** argument with one of the following values: • **true**—Enables the non-expiring deep link feature for the specified subscriber account. When set to **true**, the SSM creates an entitlement code for generating a type-3 non-expiring deep link for the subscriber account. • **false**—Disables the non-expiring deep link feature for the specified subscriber account. Setting the **"mdm-deeplink-enabled"** field to **false** removes the entitlement code from the Client Authentication System (CAS) configuration. The default value is **false**.

Considerations for Setting the “internal-domains” Field

Consider the following information when setting the **"internal-domains"** field:

- This API replaces the entire list of internal domains a subscriber can access. To add access to additional domains, the **"internal-domains"** field needs to include all internal domains the subscriber needs to access. For example, if you want to add access to **domain_3** to a subscriber that already access to **domain-1**.com and **domain-2.com**, you need to include all three domains in the **"internal-domains"** field as shown in the following example:

  ```
  "internal-domains":[domain-1.com, domain-2.com, domain-3.com]
  ```

- To remove access to an individual domain, exclude that domain from the **"internal-domains"** field. For example, if you want to remove access to domain-1.com (while keeping access to domain-2.co and domain-2.com), excluded domain-1.com from the **"internal-domains"** field as shown in the following example:

  ```
  "internal-domains":[domain-2.com, domain-3.com]
  ```

- To remove access to all internal domains, leave the **"internal-domains"** field empty as shown the the following example:

  ```
  "internal-domains":[]
  ```

- Replacing the ***<domain>*** argument with **null** (as shown the the following example) has no effect on prior internal domains settings; any previously specified domains remain in tact:

  ```
  "internal-domains":["null"]
  ```

- To verify the SSM is sending the specified domain names to the CAS, use the **GET /account/<*accountId*>/spson** API; see the “Retrieve the SIA Remote Service Configuration for a Subscriber Account” section for details.

Expected Response

The SSM updates the subscriber account as requested.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Examples

The example that follows updates the number of devices allowed on the subscriber account **sub123** to be **3**:

```
SSM> PATCH /account/sub123/spson {"allowed-devices": 3}
```

The example that follows updates a subscriber account (called **sub123**) with access to the internal domain **akamai.com**:

```
SSM> PATCH /account/sub123/spson {"internal-domains": ["akamai.com"]}
```

The example that follows enables MDM on a subscriber account called **joe.coffee**:

```
SSM> PATCH /account/joe.coffee/spson {"mdm-deeplink-enabled": true}
```

The example that follows disables MDM on a subscriber account called **joe.coffee**:

```
SSM> PATCH /account/joe.coffee/spson {"mdm-deeplink-enabled": false}
```

Update the Device and User Names Associated with an SIA Remote Service Device

Use the following **PATCH** API to update the device and user names associated with a specific device:

```
SSM> PATCH  /account/<accountId>/spson/devices/<uuid>
{
  "name": "<device-name>",
  "username": "<first-name last-name>"
}
```

Fields

Field	Description
<*accountId*>	Required. Identifies the subscriber associated with the SIA Remote service device for which to display configuration details.
<*uuid*>	Required. Specifies the universally unique identifier (UUID) associated with the device you want to update.Replace the **<*uuid*>** argument with the 128-bit number that is used to identify the device.
"name"	Required. Identifies the logical device. Replace the **<*device-name*>** argument with the device name. The name can be a unique name or a combination of a username, device model, and operating system (OS) installed on the device.
"username"	Required. Identifies the first and last name of the subscriber associated with the specified device. Replace the **<*first-name last-name*>** argument with the subscriber name; include a space between the first and last name.

Expected Response

The SSM updates the device as requested.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows updates the device and user names associated with a device that belongs to the subscriber account called **joe.coffee** and has the UUID **2b6f0cc904d137be2e1730235f5664094b831186**:

```
SSM> PATCH /account/joe.coffee/spson/devices/2b6f0cc904d137be2e1730235f5664094b831186
{"name": "sub123", "username": "jane doe"}
```

Rotate the Entitlement Code for Time-bound Deep Links

Use the following **PUT** API to rotate the entitlement code for time-bound deep links:

```
SSM> PUT /account/<accountId>/spson/entitlement
```

NOTE: Because time-bound and non-expiring deep links have different entitlement codes, rotate expiring and non-expiring entitlement codes independently from one another. To rotate the entitlement code for a non-expiring deep link, use the API described in the “Rotate the Entitlement Code for Non-expiring Deep Links” section.

Fields

Field	Description
<accountId>	Required. Identifies the subscriber account for which to rotate the entitlement code; replace the ***<accountId>*** argument with the name of the subscriber.

Expected Response

The API performs the following tasks:

- Generates and saves a new entitlement code that gets sent to the CAS.
- Generates an expiring deep link using the new entitlement code.

When you rotate the deep link, the previous deep link value is no longer valid. Any attempt to register a remote device with the old deep link results in an "Activation information invalid" message in the SIA Remote application.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Examples

The example that follows rotates the entitlement code and generates a new time-bound deep link for a subscriber account called **joe.coffee**:

```
SSM> PUT /account/joe.coffee/spson/entitlement
```

Rotate the Entitlement Code for Non-expiring Deep Links

Use the following **PUT** API to rotate the entitlement code for non-expiring (fixed) deep links:

```
SSM> PUT /account/<accountId>/spson/mdm_entitlement
```

NOTE: Because time-bound and non-expiring deep links have different entitlement codes, rotate expiring and non-expiring entitlement codes independently from one another. This section describes the API that rotates the entitlement code for a non-expiring deep link. To rotate the entitlement code for a time-bound deep link, use the API described in the "Rotate the Entitlement Code for Time-bound Deep Links" section.

Fields

Field	Description
<accountId>	Required. Identifies the subscriber account for which to rotate the entitlement code; replace the ***<accountId>*** argument with the name of the subscriber account.

Expected Response

The API performs the following tasks:

- Generates and saves a new entitlement code that gets sent to the CAS.
- Generates a non-expiring deep link using the new entitlement code.

When you rotate the entitlement code for a non-expiring deep link, the previous non-expiring deep link value is no longer valid. Any attempt to register a remote device with the old non-expiring deep link results in an "`Activation information invalid`" message in the SIA Remote application.

If the non-expiring deep link feature is not enabled for the subscriber, the API returns a "SERVICE_NOT_ENABLED" error as demonstrated in the following example:

```
SSM> PUT /account/joe.coffee/spson/mdm_entitlement
Response:
{
    "error": {
        "status": 503,
        "reason": "SERVICE_NOT_ENABLED",
        "message": "service not enabled: MDM Deeplink feature is
not enabled for the subscriber" }
```

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Examples

The example that follows rotates the entitlement code and generates a new non-expiring deep link for a subscriber called **joe.coffee**:

```
SSM> PUT /account/joe.coffee/spson/mdm_entitlement
```

Retrieve the SIA Remote Service Configuration for a Subscriber Account

Use the following **GET** API to retrieve the SIA Remote service configuration for a specific subscriber:

```
SSM> GET /account/<accountId>/spson
```

Fields

Field	Description
<accountId>	Required. Identifies the subscriber for which to retrieve SIA Remote service configuration details. Replace the ***<accountId>*** argument with the name of the subscriber for which to display SIA Remote service configuration information.

Expected Response

The API returns SIA Remote service-related information for the specified subscriber as shown in the example that follows:

```
{
    "allowed-devices": <number-of-devices>,
    "customer-id": "<uuid>",
    "entitlement": "<entitlement-code>",
    "mdm-deeplink-enabled": <booelan>,
    "mdm-entitlement":"<entitlementCode>",
    "internal-domains": [
        "<domain>",
        …
    ]
}
```

The table that follows describes the fields displayed in the API response:

Field	Description
"allowed-devices"	Maximum number of SIA Remote service devices the subscriber can have.
"customer-id"	Unique identifier that represents a global entitlement declared for the subscriber.
"entitlement"	Secret code SIA Remote service uses to authenticate the client's SIA Remote service registration requests. The code is valid only for those devices that have SIA Remote service-enabled and has an expiration time.
"mdm-deeplink-enabled"	Indicates whether non-expiring deep links are enabled for the specified subscriber account.
"mdm-entitlement"	Displays the entitlement UUID for the non-expiring deep link. **NOTE:** If non-expiring deep links are disabled for the subscriber account (if the **"mdm-deeplink-enabled"** field is set to **false**), the response does not include the "mdm-entitlement" value.

Field	Description
"internal-domains"	Lists the internal and local domains devices can access.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows retrieves SIA Remote service configuration details for a subscriber called **sub123**:

```
SSM> GET /account/sub123/spson
{
    "allowed-devices": 3,
    "customer-id": "772a0c7b-99a3-40f0-b63f-c12c26bb6584",
    "entitlement": "c7866937-3699-4006-970a-0be462c47bc2",
    "mdm-deeplink-enabled":false
    "internal-domains": [
        "akamai.com"
    ]
}
```

The example that follows retrieves SIA Remote service configuration details for a subscriber called **joe.coffee**; this subscriber account has non-expiring deep links enabled, so the response includes the "mdm-entitlement" field:

```
SSM> GET /account/joe.coffee/spson
Response:
{
      "allowed-devices":4,
      "customer-id":"4f5d34ae-11ab-4e45-9677-ced7357d9246",
      "entitlement":"9c94ca91-4545-4e94-b91d-f1fc72645c9c",
      "mdm-deeplink-enabled":true
      "mdm-entitlement":"9053589c-3f2e-486f-aad0-9bb3325d0985"
}
```

Retrieve Configuration Details for an SIA Remote Service Subscriber Device

Use the following **GET** API to retrieve configuration information for a specific subscriber device that is registered for the SIA Remote service:

```
SSM> GET /account/<accountId>/spson/devices/<device_UUID>
```

Fields

Field	Description
<accountId>	Required. Identifies the subscriber associated with the SIA Remote service device for which to display configuration details.
<device_UUID>	Optional. Specifies the UUID of the SIA Remote service device for which to display configuration details.

Expected Response

The API returns detailed information for the specified device as shown in the example that follows:

```
{
"name": "<deviceName>",
"user_name": "<deviceUsernameOwner>",
"model": "<deviceModel>",
"os": "<osVersion>",
"software_version": "<softwareVersion>",
"status": "<deviceSatus>"
}
```

The table that follows describes the fields displayed in the API response:

Field	Description
"name"	Identifies the SIA Remote service roaming device.
"user_name"	Identifies the SIA Remote device owner.
"model"	Identifies the SIA Remote device model.
"os"	Identifies the SIA Remote device operating system.
"software_version"	Identifies the software version installed on the device.
"software_version"	Provides information in regards to the SIA Remote device status.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows displays configuration information for the SIA Remote service subscriber called **sub123** that has the UUID **2b6f0cc904d137be2e1730235f5664094b831186**:

```
SSM> GET /account/sub123/spson/devices/2b6f0cc904d137be2e1730235f5664094b831186

{
"name": "jdoe - iPhone9,3 - 1729776153100",
"user_name": "jon doe",
"model": "iPhone9,3",
"os": "iOS 14",
"software_version": "ios-1.1.0",
"status": "on-net"
}
```

Remove an SIA Remote Service Object from a Subscriber Account

Use the following **DELETE** API to remove the SIA Remote service object from a specific subscriber account:

```
SSM> DELETE /account/<accountId>/spson
```

Fields

Field	Description
<accountId>	Specifies the subscriber from which to remove SIA Remote service support. Replace the ***<accountId>*** argument with the desired subscriber identifier.

Expected Response

The SSM removes SIA Remote service support from the specified subscriber account.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows removes SIA Remote service support for a subscriber called **sub123**:

```
SSM> DELETE /account/sub123/spson
```

Unenroll a Device from the SIA Remote Service

Use the following **DELETE** API to remove a specific device from the SIA Remote service:

```
SSM> DELETE /account/<accountId>/spson/devices/<device-uuid>}
```

Use this API when a subscriber no longer uses the device or when the device is suspected to be compromised.

Fields

Field	Description
<accountId>	Required. Specifies the subscriber from which to remove SIA Remote service support. Replace the ***<accountId>*** argument with the desired subscriber identifier.
<device-uuid>	Required. Specifies the universally unique identifier (UUID) associated with the device to unenroll from a subscriber's SIA Remote service. Replace the ***<device-uuid>*** argument with the 128-bit number that is used to identify the device.

Expected Response

The SSM removes SIA Remote service support from the specified subscriber account.

NOTE: For a list of possible response and error codes, see Chapter 15, "API Response Codes."

Example

The example that follows unenrolls a device that has the UUID **2b6f0cc904d137be2e1730235f5664094b831186** from SIA Remote service support for a subscriber called **sub123**:

```
SSM> DELETE /account/sub123/spson/devices/2b6f0cc904d137be2e1730235f5664094b831186
```

Deactivate the SIA Remote Service for a Subscriber Account

Use the following **DELETE** API to deactivate SIA Remote service for a subscriber account:

```
SSM> DELETE /control/<accountId>/service/spson
```

Fields

Field	Description
<accountId>	Specifies the subscriber on which to deactivate SIA Remote service support. Replace the ***<accountId>*** argument with the desired subscriber identifier.

Expected Response

The SSM deactivates SPS Remote service support from the specified subscriber account.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

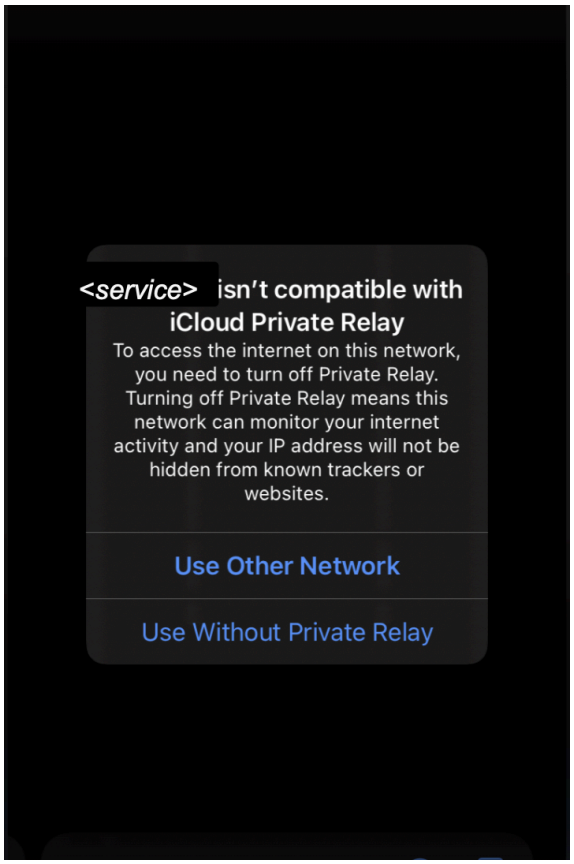
The example that follows removes SIA Remote service support for a subscriber called **sub123**:

```
SSM> DELETE /control/sub123/service/spson
```

Chapter 12: iCloud Private Relay Blocking APIs

iCloud Private Relay (iOS Release 16.1) is an opt-in service Apple provides to paid iCloud users that, when enabled on a device, routes DNS around SIA service protections. The SIA SMB application supports an iCloud Private Relay blocking feature you can enable and disable as desired. The iCloud Private Relay blocking feature has the following effect:

- When the iCloud Private Relay blocking feature is enabled in your SIA SMB service, the service blocks iCloud Private Relay on devices accessing the SIA SMB service. In addition, eligible Apple device users receive the following prompt to accept iCloud Private Relay blocking in order to connect to your network:



- When the iCloud Private Relay blocking feature is disabled in your SIA Consumer service, DNS traffic from devices that have iCloud Private Relay enabled gets directed to an alternate DNS resolver without SIA SMB protections.

See the ”Manage iCloud Private Relay Blocking Support” section in the SIA SMB Administrator’s Guide for details on implementing the iCloud Private Relay blocking service in your deployment.

This chapter describes the SSM APIs for enabling and disabling iCloud Private Relay blocking on individual subscriber accounts; these APIs are available for SIA SMB services that have the iCloud Private Relay Blocking feature enabled. This chapter describes the SSM APIs for performing the following tasks:

- Enable iCloud Private Relay Blocking for a Subscriber Account
- Disable iCloud Private Relay Blocking for a Subscriber Account

NOTE: To determine whether the iCloud Private Relay blocking feature is enabled for a subscriber account, use the **`GET /account/<accountId>`** API and locate the "bypass-private-dns": { "apple-private-relay": <*value*>} field in the API response. See the "Retrieve Subscriber Account Service Details" section in Chapter 2, "Subscriber Account APIs" for details about the **`GET /account/<accountId>`** API.

Enable iCloud Private Relay Blocking for a Subscriber Account

Use the following **PATCH** API to enable iCloud Private Relay blocking for a subscriber account:

```
SSM> PATCH /account/<accountId>
{"bypass-private-dns":{"apple-private-relay":true}}
```

Fields

Field	Description
<accountId>	Specifies the subscriber account for which to enable iCloud Private Relay blocking. Replace the ***<accountId>*** argument with the desired subscriber identifier.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body indicating iCloud Private Relay blocking is enabled. The example that follows shows the JSON output returned in the API response body:

```
{
   "successful": [
       {
           "field": "bypass-private-dns",
           "value": {
               "apple-private-relay": true
           }
       }
   ],
   "status": "success"
}
```

In the response, the "apple-private-relay" field indicates whether iCloud Private Relay blocking is enabled for a subscriber account. A **true** value indicates iCloud Private Relay blocking is enabled for the subscriber account, while a **false** value indicates iCloud Private Relay blocking is disabled for the subscriber account.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows enables iCloud Private Relay blocking for a subscriber account called **joe.coffee**:

```
SSM> PATCH /account/joe.coffee
{"bypass-private-dns":{"apple-private-relay":true}}
{
   "successful": [
       {
           "field": "bypass-private-dns",
           "value": {
               "apple-private-relay": true
           }
       }
   ],
   "status": "success"
}
```

Disable iCloud Private Relay Blocking for a Subscriber Account

Use the following **PATCH** API to disable iCloud Private Relay blocking for a subscriber account:

```
SSM> PATCH /account/<accountId>
{"bypass-private-dns":{"apple-private-relay":false}}
```

Fields

Field	Description
<accountId>	Specifies the subscriber account for which to disable iCloud Private Relay blocking. Replace the ***<accountId>*** argument with the desired subscriber identifier.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body indicating iCloud private Relay blocking is disabled. The example that follows shows the JSON output returned in the API response body:

```
{
    "successful": [
        {
            "field": "bypass-private-dns",
            "value": {
                "apple-private-relay": false
            }
        }
    ],
    "status": "success"
}
```

In the response, the "apple-private-relay" field indicates whether iCloud Private Relay blocking is disabled for a subscriber account. A **false** value indicates iCloud Private Relay blocking is disabled for the subscriber account, while a **true** value indicates iCloud Private Relay blocking is enabled for the subscriber account.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows disables iCloud Private Relay blocking for a subscriber account called **joe.coffee**:

```
SSM> PATCH /account/joe.coffee
{"bypass-private-dns":{"apple-private-relay":false}}
{
   "successful": [
       {
           "field": "bypass-private-dns",
           "value": {
               "apple-private-relay": false
           }
       }
   ],
   "status": "success"
}
```

Chapter 13: Check Site APIs

This chapter describes the Check Site APIs for performing the following tasks:

- Retrieve the list node names associated with a URL
- Retrieve domain name details for multiple URLs

Retrieve the List Node Names Associated with a URL

Use the following **POST** API to display the list node domains associated with a particular uniform resource locator (URL):

```
SSM> POST /check/list/node
{
  "url": "<urlString>"
}
```

Fields

Field	Description
<urlString>	Required. Specifies the URL for which to retrieve domain name details. Replace the ***<urlString>*** argument with the desired URL as demonstrated in the following examples: • **`example.com`** • **`www.example.com`** **NOTE:** In addition to Top Level Domains (TLDs), this API supports Internationalized Domain Names (IDNs).

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body; this output shows information about each domain type:

```
{
    "result": [
        {
            "type": "<domainType>",
            "node": "<domainName>"
        },
        …
    ]
}
```

The table that follows describes the fields in the API response:

Field	Description
“type”	String that identifies the domain type for the specified URL; possible domain types are: • core-domain • fully qualified domain name (FQDN) • host-path • naked-host-path • public-suffix
“node”	String that identifies the domain name for the specified domain type.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows retrieves the domain names for the URL **example.com**:

```
SSM> POST /check/list/node {"url": "example.com"}


{
    "result": [
        {
            "type": "core-domain",
            "node": "example.com"
        },
        {
            "type": "fqdn",
            "node": "example.com"
        },
        {
```

```
            "type": "host-path",
            "node": "example.com"
        },
        {
            "type": "naked-host-path",
            "node": "example.com"
        },
        {
            "type": "public-suffix",
            "node": "com"
        }
    ]
}
```

Retrieve Domain Name Details for Multiple URLs

Use the following **POST** API to display the core-domain, fully qualified domain name (FQDN), host-and naked-host-path, and public suffix for the specified URLs:

```
SSM> POST /check/list/nodes
{
  "urls": ["<urlString>", ...]
}
```

Fields

Field	Description
<urlString>	Required. Specifies the URL for which to retrieve domain name details. Replace the ***<urlString>*** argument with the desired URL as demonstrated in the following examples: • **`example.com`** • **`www.example.com`** When specifying multiple URLs, separate each URL with a comma as shown in the following example: **`"urls": ["example-1.com", "example-2.com", "example-3.com"]`** **NOTE:** In addition to Top Level Domains (TLDs), this API supports Internationalized Domain Names (IDNs).

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the output returned in the response body; this information is displayed for each URL you specify in the API:

```
{
    "failures": [

    ],
    "result": [
        {
            "url": "<url>",
            "nodes": [
                {
                    "type": "<domainType",
                    "node": "<domainName>"
                },
                ...
            ]
        },
        ...
    ]
}
```

The table that follows describes the fields in the API response:

Field	Description
"url"	Identifies the URL for which domain type and node details are displayed. There is a "url" field for each domain specified in the API call.
“type”	String that identifies the domain type for the specified URL; possible domain types are: • core-domain • fully qualified domain name (FQDN) • host-path • naked-host-path • public-suffix
“node”	String that shows the domain name for the specified domain type.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows retrieves domain type and name information for three URLs (**example-1.com**, **example-2.com**, and **akamai.com**):

```
SSM> POST /check/list/nodes {"urls": ["example-1.com", "example-2.com",
"akamai.com"]}
{
    "failures": [

    ],
    "result": [
        {
            "url": "example-1.com",
            "nodes": [
                {
                    "type": "core-domain",
                    "node": "example-1.com"
                },
                {
                    "type": "fqdn",
                    "node": "example-1.com"
                },
                {
                    "type": "host-path",
                    "node": "example-1.com"
                },
                {
                    "type": "naked-host-path",
                    "node": "example-1.com"
                },
                {
                    "type": "public-suffix",
                    "node": "com"
                }
            ]
        },
        {
            "url": "example-2.com",
            "nodes": [
                {
                     "type": "core-domain",
                     "node": "example-2.com"
                },
                {
                    "type": "fqdn",
                    "node": "example-2.com"
                },
                {
                    "type": "host-path",
```

```
                    "node": "example-2.com"
                },
                {
                    "type": "naked-host-path",
                    "node": "example-2.com"
                },
                {
                    "type": "public-suffix",
                    "node": "com"
                }
            ]
        },
        {
            "url": "akamai.com",
            "nodes": [
                {
                    "type": "core-domain",
                    "node": "akamai.com"
                },
                {
                    "type": "fqdn",
                    "node": "akamai.com"
                },
                {
                    "type": "host-path",
                    "node": "akamai.com"
                },
                {
                    "type": "naked-host-path",
                    "node": "akamai.com"
                },
                {
                    "type": "public-suffix",
                    "node": "com"
                }
            ]
        }
    ]
}
```

Chapter 14: SSM Reporting APIs

This chapter describes the SSM API calls you can use to retrieve statistics relating to a series of subscriber reporting resources. This chapter includes the calls you can use to perform the following tasks:

- Retrieve the number of queries sent by the devices within a specific interval
- Return the number of domain requests by device
- Retrieve the number of content request blocks for one or all device groups
- Retrieve the number of malware and phishing threat blocks
- Retrieve the number of botnets blocked for device groups

Retrieve the Number of Queries Sent by Devices within a Specific Interval

Use one of the following **GET** APIs to retrieve the number of queries that were sent within a specific interval by the devices associated with a subscriber account:

- To retrieve the number of queries sent in a specific interval and during a specific time period (where you specify the start and stop times for data collection), use the following syntax:

```
SSM> GET /series/dns/requests-by-device-<interval>?view=<accountId>&
    start_inclusive=<startTime>&end_exclusive=<endTime>
```

- To retrieve the number of queries sent in a specific interval and during a specific duration of time (for example, a number of hours, days, years, months, and so forth), use the following syntax:

```
SSM> GET /series/requests-by-device-<interval>?view=<accountId>&
    duration=<time>
```

With this API format, use a question mark (**?**) to separate the request header from the request body parameters; in the request body, separate each field with an ampersand (**&**).

Fields

Field	Description
<*interval*>	Required. Specifies the interval during which the SSM collects query data. Replace the **<*interval*>** argument with one of the following interval keywords: • **15m**—Specifies a fifteen minute interval. • **2h**—Specifies a two-hour interval. • **8h**—Specifies an eight-hour interval.
view=<*accountId*>	Required. Identifies the subscriber account (also known as a "view") for which to display data. Replace the **<*accountId*>** argument with the subscriber account ID.
start_inclusive=<*startTime*>	Required if the API request does not include the **duration** construct. Replace the **<*startTime*>** argument with the time (in seconds, formatted for UNIX) at which the SSM starts collecting query data. The maximum time period for which you can collect query data is one month. **NOTE:** The **start_inclusive** and **end_exclusive** constructs are mutually exclusive with the **duration** construct. An API request cannot include **start_inclusive** and **end_exclusive** constructs with the **duration** construct.
end_exclusive=<*endTime*>	Required if the API request includes the **start_inclusive** construct. String that identifies the time at which SSM stops collecting data. Replace the **<*endTime*>** argument with the end time in the UNIX format.
duration=<*timePeriod*>	Required if the API request does not include the **start_inclusive** and **end_exclusive** constructs. Specifies the time period during which the SSM collects data. Replace the **<*time-period*>** argument with a time period in the ISO-8601 format. The maximum time period is one month. **NOTE:** The **duration** construct is mutually exclusive with the **start_inclusive** and **end_exclusive** constructs. An API request cannot include the **duration** construct with the **start_inclusive** and **end_exclusive** constructs.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body:

```
[
    {
        "time": <string>,
        "value": <integer>,
    }
    ...
]
```

The table that follows describes the fields displayed in the API response body:

Field	Description
"time"	Time at which a query was sent.
"value"	Number of queries sent during the specified interval.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows displays the number of queries that were sent by devices associated a subscriber account called **joe.coffee**:

```
SSM> GET /series/dns/requests-by-device-15m?view=joe.coffee&start_inclusive=
0&end_exclusive=2000000000
[
    {
        "time": 1638925200,
        "value": 146
    },
    {
        "time": 1638926100,
        "value": 179
    },
    {
        "time": 1638927000,
        "value": 202
```

```
    },
    {
        "time": 1638927900,
        "value": 186
    },
    {
        "time": 1638928800,
        "value": 207
    }
]
```

Return the Number of Domain Requests by Device

Use one of the following **GET** APIs to retrieve the number of domain requests sent from devices during a specific time period; this API gathers data from all devices assigned to a specified subscriber account:

- To return the total number of domain requests that occurred over a specified time period (where you specify the start and stop times for the time period), use the following syntax:

```
SSM> GET /series/dns/requests-by-device?view=<accountId>
    &start_inclusive=<startTime>&end_exclusive=<endTime>
```

- To return the total number of domain requests that occurred within a specific duration of time (for example, a number of hours, days, weeks, months, and so forth), use the following syntax:

```
SSM> GET /series/dns/requests-by-device?view=<accountId>
    &duration=<time-period>
```

Fields

Field	Description
view=<*accountId*>	Optional. Identifies the subscriber account (also known as a “view”) for which to display data. Replace the **<*accountId*>** argument with the subscriber account ID. If the API request does not include the **view** construct, the response includes domain request data for all subscriber accounts in the SIA SMB service.

Field	Description
start_inclusive=<*startTime*>	Optional. Replace the **<*startTime*>** argument with the time (in seconds, formatted for UNIX) at which the SSM starts collecting domain request data. The maximum time period for which you can collect domain request data is one month. **NOTE:** The **start_inclusive** and **end_exclusive** constructs are mutually exclusive with the **duration** construct. An API request cannot include **start_inclusive** and **end_exclusive** constructs with the **duration** construct.
end_exclusive=<*endTime*>	Required if the API request includes the **start_inclusive** construct. String that identifies the time at which SSM stops collecting domain requests. Replace the **<*endTime*>** argument with the end time in the UNIX format.
duration=<*time-period*>	Optional. Specifies the time period during which the SSM collects domain request data. Replace the **<*time-period*>** argument with a time period in the ISO-8601 format. The maximum time period is one month. **NOTE:** The **duration** construct is mutually exclusive with the **start_inclusive** and **end_exclusive** constructs. An API request cannot include the **duration** construct with the **start_inclusive** and **end_exclusive** constructs.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body:

```
[
    {
        "time":string,
        "Value":integer
    }
    ...
]
```

The table that follows describes the fields displayed in the API response body:

Field	Description
"time"	Time at which the device sent a domain request.
"value"	Number of domain requests.

Example

The example that follows shows how to display the number of domain requests made by devices that are associated with a subscriber account called **joe.coffee**:

```
SSM> GET /series/dns/requests-by-device?view=joe.coffee&duration=PT1H
{"view":"joe.coffee"} [{"time":1547577000,"value":128},
{"time":1547577900,"value":143},
{"time":1547578800,"value":109},{"time":1547579700,"value":128},
{"time":1547580600,"value":124},{"time":1547581500,"value":117},
{"time":1547582400,"value":118},{"time":1547583300,"value":117},
{"time":1547584200,"value":112},{"time":1547585100,"value":135},
{"time":1547586000,"value":132},{"time":1547586900,"value":124}]
]
```

Retrieve the Number of Content Request Blocks for One or All Device Groups

Use the one of the following **GET** APIs to retrieve the number of content requests blocked for all device groups or a specific device group:

- To retrieve the number of content request blocks that occurred between a specific start and stop time, use the following syntax:

  ```
  SSM> GET /series/secure/content/blocks-by-group?view=<accountId>
      &group_name=<groupId>&start_inclusive=<startTime>
      &end_exclusive=<endTime>
  ```

- To retrieve content request block data from a specific duration of time (for example, a number of hours, days, years, months, and so forth), use the following syntax:

  ```
  SSM> GET /series/secure/content/blocks-by-group?view=<accountId>
      &group_name=<groupId>&duration=<time-period>
  ```

Fields

Field	Description
view=<*accountId*>	Required. Identifies the subscriber account (also known as a "view") for which to display data. Replace the **<*accountId*>** argument with the subscriber account ID.
group_name=<*groupId*>	Optional. String that identifies the device group for which to display the total number of content request blocks; the response includes content block information for all devices in the specified device group. Replace the **<*groupId*>** argument with the name of the desired group.
start_inclusive=<*startTime*>	Required if the API request does not include the **duration** construct. Replace the **<*startTime*>** argument with the time (in seconds, formatted for UNIX) at which the SSM starts collecting content request block data. The maximum time period for which you can collect content request block data is one month. **NOTE:** The **start_inclusive** and **end_exclusive** constructs are mutually exclusive with the **duration** construct. An API request cannot include **start_inclusive** and **end_exclusive** constructs with the **duration** construct.
end_exclusive=<*endTime*>	Required if the API request includes the **start_inclusive** construct. String that identifies the time at which SSM stops collecting content request blocks. Replace the **<*endTime*>** argument with the end time in the UNIX format.
duration=<*time-period*>	Required if the API request does not include the **start_inclusive** and **end_exclusive** constructs. Specifies the time period during which the SSM collects content request block information. Replace the **<*time-period*>** argument with a time period in the ISO-8601 format. The maximum time period for which you can collect content request block information is one month. **NOTE:** The **duration** construct is mutually exclusive with the **start_inclusive** and **end_exclusive** constructs. An API request cannot include the **duration** construct with the **start_inclusive** and **end_exclusive** constructs.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body:

```
[
     {
          "time":<string>,
          "value":<integer>
     },
          ...
]
```

The table that follows describes the fields displayed in the API response body:

Field	Description
"time"	Time at which the device or device group encountered the content request block.
"value"	Number of content requests that were blocked.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows shows how to display the number of content request blocks encountered by the view called **joe.coffee** for the previous day (as specified by the **duration=P1D** construct):

```
SSM> GET /series/secure/content/blocks-by-group?view=joe.coffee&duration=P1D
[
     {
          "time": 1635468300,
          "value": 1
     },
     {
          "time": 1635470100,
          "value": 2
     },
     {
          "time": 1635485400,
          "value": 1
     },
     {
          "time": 1635488100,
          "value": 1
]
```

Retrieve the Number of Malware and Phishing Threat Blocks

Use one of the following **GET** API to retrieve the number of malware and phishing threats the SSM blocked for all device groups or for a specific group:

- To retrieve the number of malware and phishing threats the SSM blocked between a specific start and stop time, use the following syntax:

```
SSM> GET /series/secure/threat/blocks-by-group?view=<accountId>
    &group_name=<groupId>&start_inclusive=<startTime>
    &end_exclusive=<endTime>
```

- To retrieve number of malware and phishing threats the SSM blocked within a specific duration of time (for example, a number of hours, days, years, months, and so forth), use the following syntax:

```
SSM> GET /series/secure/threat/blocks-by-group?view=<accountId>
    &group_name=<groupId>&duration=<time-period>
```

Fields

Field	Description
view=<*accountId*>	Required. Identifies the subscriber account (also known as a “view”) for which to display data. Replace the **<*accountId*>** argument with the subscriber account ID.
group_name=<*groupId*>	Optional. String that identifies the device group for which to display the total number of malware and phishing threat encounters for all devices. Replace the **<*groupId*>** argument with the name of the desired group. **NOTE:** If you do not include the **group_name=<*groupId*>** construct in the API call, the response body displays data for all device groups that are associated with the specified subscriber account (view).

Field	Description
start_inclusive=<*startTime*>	Required if the API request does not include the **duration** construct. Replace the **<*startTime*>** argument with the time (in seconds, formatted for UNIX) at which the SSM starts collecting blocked malware and phishing threat encounters. The maximum time period for which you can collect malware and phishing threat encounters is one month. **NOTE:** The **start_inclusive** and **end_exclusive** constructs are mutually exclusive with the **duration** construct. An API request cannot include **start_inclusive** and **end_exclusive** constructs with the **duration** construct.
end_exclusive=<*endTime*>	Required if the API request includes the **start_inclusive** construct. String that identifies the time at which SSM stops collecting malware and phishing threat encounters. Replace the **<*endTime*>** argument with the end time in the UNIX format.
duration=<*time-period*>	Required if the API request does not include the **start_inclusive** and **end_exclusive** constructs. Specifies the time period during which the SSM collects malware and phishing threat encounters. Replace the **<*time-period*>** argument with a time period in the ISO-8601 format. The maximum time period for which you can collect malware and phishing threat encounters is one month. **NOTE:** The **duration** construct is mutually exclusive with the **start_inclusive** and **end_exclusive** constructs. An API request cannot include the **duration** construct with the **start_inclusive** and **end_exclusive** constructs.

Expected Response

A successful response includes HTTP code 200 with a JSON based response body. The example that follows shows the JSON output returned in the API response body:

```
[
    {"time":<string>,
     "value":<integer>}
    ...
]
```

The table that follows describes the fields displayed in the API response body:

Field	Description
"time"	Time at which the device encountered the malware or phishing threat.
"value"	Number of blocked malware or phishing threat attempts.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows shows how to display the number of malware and phishing threats encountered by the view called **joe.coffee**:

```
SSM> GET /series/secure/threat/blocks-by-group?view=joe.coffee&duration=P1D
[{"time":1547066700,"value":4},{"time":1547067600,"value":4},
{"time":1547068500,"value":2},{"time":1547069400,"value":5},
{"time":1547070300,"value":3}]
```

Retrieve Details About Botnet Blocks that Occurred During a Specific Time Period

Use one of the following **GET** APIs to retrieve details about the botnets a subscriber account blocked during a specified time period:

- To retrieve details about the botnet blocks that occurred between a specific start and stop time, use the following syntax:

```
SSM> GET /series/secure/threat/stops?view=<accountId>
    &start_inclusive=<startTime>&end_exclusive=<endTime>
```

- To retrieve details about the botnet blocks that occurred within a specific duration of time (for example, a number of hours, days, years, months, and so forth), use the following syntax:

```
SSM> GET /series/secure/threat/stops?view=<accountId>
     &duration=<time-period>
```

Fields

Field	Description
view=<*accountId*>	Required. Identifies the subscriber account (also known as a “view”) for which to display botnet block data. Replace the **<*accountId*>** argument with the subscriber account ID.
start_inclusive=<*startTime*>	Required if the API request does not include the **duration** construct. Replace the **<*startTime*>** argument with the time (in seconds, formatted for UNIX) at which the SSM starts collecting botnet block data. The maximum time period for which you can collect device access information is one month. **NOTE:** The **start_inclusive** and **end_exclusive** constructs are mutually exclusive with the **duration** construct. An API request cannot include **start_inclusive** and **end_exclusive** constructs with the **duration** construct.
end_exclusive=<*endTime*>	Required if the API request includes the **start_inclusive** construct. Replace the **<*endTime*>** argument with the time (in seconds, formatted for UNIX) at which SSM stops collecting botnet block data.
duration=<*time-period*>	Required if the API request does not include the **start_inclusive** and **end_exclusive** constructs. Replace the **<*time-period*>** argument with a time period during which the SSM collects botnet block data; specify the time period in the ISO-8601 format. The maximum time period for which you can collect device access information is one month. **NOTE:** The **duration** construct is mutually exclusive with the **start_inclusive** and **end_exclusive** constructs. An API request cannot include the **duration** construct with the **start_inclusive** and **end_exclusive** constructs.

Expected Response

A successful response includes HTTP code 200 with a JSON-based response body that provides information about each botnet a subscriber account blocked during a specified time period. The example that follows shows the JSON output returned in the API response body:

```
[
    {
        "time": <string>,
        "value": [
            {
                "name": "<threatName>",
                "threat-id": <threatId>,
                "timestamp": <timestamp>,
                "device-id": "<deviceId>"
            }
        ]
    },
  ...
]
```

The table that follows describes the fields displayed in the API response body:

Field	Description
"time"	Time at which the device or device group blocked the botnet.
"value"	Displays the following information about an individual botnet block: ● "name"—Name of the botnet that was blocked. ● "threat-id"—Identification number associated with the blocked botnet. ● "timestamp"—UNIX-formatted timestamp indicating when the botnet block occurred. ● "device-id"—Identifies the device on which the botnet block occurred.

NOTE: If a device did not access the network during the specified time period, that device is not included in the API response body.

NOTE: For a list of possible response and error codes, see Chapter 15, “API Response Codes.”

Example

The example that follows retrieves detailed information about the botnets a subscriber account called **joe.coffee** blocked over a time period of one day:

```
SSM> GET /series/secure/threat/stops?view=joe.coffee&duration=P1D
[
    {
        "time": 1730115000,
        "value": [
            {
                "name": "htyfrrhgsxc.co",
                "threat-id": 1154,
                "timestamp": 1730115822000,
                "device-id": "00:16:bd:a8:2c:15"
            }
        ]
    },
    {
        "time": 1730116800,
        "value": [
            {
                "name": "nxjsdridslxmddf.work",
                "threat-id": 1105,
                "timestamp": 1730117086000,
                "device-id": "47786545636"
            }
        ]
    },
    {
        "time": 1730117700,
        "value": [
        "time": 1730180700,
            {
                "name": "kfktidmckslklk.co",
                "threat-id": 1037,
                "timestamp": 1730117986000,
                "device-id": "47786545636"
            }
        ]
    },
    {
        "time": 1730120400,
        "value": [
            {
                "name": "fjhgjtummxs.us",
                "threat-id": 1113,
                "timestamp": 1730121222000,
                "device-id": "10364484737"
            }
        ]
```

```
    },
    {
        "time": 1730124900,
        "value": [
            {
                "name": "fhglyiupf.pl",
                "threat-id": 1276,
                "timestamp": 1730125722000,
                "device-id": "18309779131"
            },
            {
                "name": "dfhirefjd.com",
                "threat-id": 1046,
                "timestamp": 1730125186000,
                "device-id": "00:16:78:3c:f0:7c"
            }
        ]
    },
    {
        "time": 1730130300,
        "value": [
            {
                "name": "dfhirefjd.com",
                "threat-id": 1031,
                "timestamp": 1730131122000,
                "device-id": "18309779131"
            }
        ]
    },
    {
        "time": 1730139300,
        "value": [
            {
                "name": "xiatyernsmxkjerw.aa",
                "threat-id": 1196,
                "timestamp": 1730140122000,
                "device-id": "28888317571"
            }
        ]
    },
    {
        "time": 1730147400,
        "value": [
            {
                "name": "htyfrrhgsxc.co",
                "threat-id": 1031,
                "timestamp": 1730147686000,
                "device-id": "38233255753"
            }
        ]
    },
```

```
    {
        "time": 1730148300,
        "value": [
            {
                "name": "porotidmscfcfmj.info",
                "threat-id": 1113,
                "timestamp": 1730148586000,
                "device-id": "00:16:78:3c:f0:7c"
            }
        ]
    },
    {
        "time": 1730149200,
        "value": [
            {
                "name": "kfktidmckslklk.co",
                "threat-id": 1033,
                "timestamp": 1730149486000,
                "device-id": "38233255753"
            }
        ]
    },
    {
        "time": 1730151000,
        "value": [
            {
                "name": "fuierhfie.com",
                "threat-id": 1105,
                "timestamp": 1730151286000,
                "device-id": "47786545636"
            }
        ]
    },
    {
        "time": 1730159100,
        "value": [
            {
                "name": "xiatyernsmxkjerw.aa",
                "threat-id": 1037,
                "timestamp": 1730159386000,
                "device-id": "47786545636"
            }
        ]
    },
    {
        "time": 1730163600,
        "value": [
            {
                "name": "pldiodjklcs.uk",
                "threat-id": 1037,
                "timestamp": 1730164422000,
```

```
                "device-id": "10364484737"
            }
        ]
    },
    {
        "time": 1730175300,
        "value": [
            {
                "name": "qdxcgsoj.uk",
                "threat-id": 1196,
                "timestamp": 1730175586000,
                "device-id": "38233255753"
            }
        ]
    },
    {
        "time": 1730178900,
        "value": [
            {
                "name": "dfhirefjd.com",
                "threat-id": 1196,
                "timestamp": 1730179186000,
                "device-id": "74638664007"
            }
        ]
    },
    {
        "time": 1730180700,
        "value": [
            {
                "name": "pldiodjklcs.uk",
                "threat-id": 1031,
                "timestamp": 1730180986000,
                "device-id": "38233255753"
            }
        ]
    },
    {
        "time": 1730181600,
        "value": [
            {
                "name": "mutseldfsxcsdf.ww",
                "threat-id": 1031,
                "timestamp": 1730181886000,
                "device-id": "38233255753"
            }
        ]
    },
    {
        "time": 1730189700,
        "value": [
```

```
            {
                "name": "htyfrrhgsxc.co",
                "threat-id": 1031,
                "timestamp": 1730190522000,
                "device-id": "18309779131"
            }
        ]
    },
    {
        "time": 1730194200,
        "value": [
            {
                "name": "mutseldfsxcsdf.ww",
                "threat-id": 1046,
                "timestamp": 1730194486000,
                "device-id": "47786545636"
            }
        ]
    }
]
```

Chapter 15: API Response Codes

The table that follows describes the possible codes a Subscriber Services Manager (SSM) API response can contain:

Response Code	Description
200 OK	The API request is successful.
201 Created	The API request is successful and the requested object is created in your system.
400 Bad_Request	The server could not process the API request because part of the request was invalid. For example, you will receive a 400 Bad_Request error message if you specify an invalid time zone value in the POST /control API.
401 Unauthorized	The API request requires user authentication.
403 Forbidden	The server understood but is refusing to fulfill the API request.
404 Not Found	The API does not exist or the request contains a typo.
409 Conflict	The SSM could not complete the API request due to a conflict with the current state of the target resource. For example, the API returns a 409 "SERVICE_ALREADY_STARTED" error when you try to enable a service on a subscriber account for which the service is already enabled.
424 Failed Dependency	The SSM returns a 424 Failed Dependency error if the specified Postgres cluster is misconfigured or does not exist.
500 Internal Server Error	The SSM could not complete the API request because an unexpected condition occurred. For example, when checking the health of the connection between SSM and the Postgres cluster.
503 Service Unavailable	The SSM could not complete the API request because a server-side error occurred (for example, the server hosting the SSM or your SSM services is not available).

Appendix: Using the nom-http Client

The nom-http client interacts with Akamai destinations such as the Subscriber Services Manager (SSM), enabling you to send individual API requests or API requests in batches so that you can obtain configuration information quickly. Using the nom-http client is typically more convenient than performing the same task using the curl client.

You can launch the nom-http client from the UNIX shell prompt or from your AMC-X jump host tool. This appendix describes how to launch the nom-http client from the UNIX shell and use the **nom-http** command. To launch the nom-http client and access the SSM API from your AMC-X jump host tool, see the "Access the SSM API nom-http Client from your AMC-X Jump Host Tool" section in Chapter 1, "Introduction to the SSM RESTful API."

The nom-http client is installed in the following location: */usr/local/nom/sbin*. To access the nom-http client from the UNIX shell, enter the **/usr/local/nom/sbin/nom-http** command at the UNIX shell prompt, including an HTTP method and any required or optional request body contents in the command string.

To access the SSM API, you need to specify port **9090**, along with your SSM username and password as demonstrated in the following steps:

1. Use the following command at the UNIX shell prompt to access the directory containing the **nom-http** client:

```
# cd /usr/local/nom/sbin/
```

2. Enter the **nom-http** command and specify port **9090**, along with your SSM username and password, as demonstrated in the example that follows:

```
# nom-http -s localhost#9090 -u <username> -p <password>
SSM>
```

nom-http Syntax Description

The **nom-http** command takes the following syntax:

```
# /usr/local/nom/sbin/nom-http <constructs…> <apiMethod> <apiBody> <path>
```

The table that follows describes the variables in the **nom-http** command:

Variable	Description
<constructs…>	Replace the ***<constructs…>*** argument with the desired keywords or constructs described in the "<constructs> variable—possible values" table.
<apiMethod>	Optional. Replace the ***<apiMethod>*** argument with one of the following HTTP methods: • **GET** • **POST** • **PUT** • **PATCH** • **DELETE**
<apiBody>	Optional. If the API call has required or optional fields, replace the ***<apiBody>*** argument with the required and, if desired, optional fields in the request body. Some APIs do not have a request body, and some APIs have an optional request body. This manual describes the required and optional fields for each SSM API; to see how to enter the request body for an API, see the description of that API in this manual.
<path>	Optional. Provides a path for the API call. Replace the argument with the desired path.

The table that follows describes the values you can use in place of the ***<constructs>*** variable:

<constructs> Variable—Possible Values

Keyword or construct	Description

-h
or
--help
	Displays usage information. The examples that follow demonstrate the syntax to use when specifying the **nom-http** command with the **-h** or **--help** keyword:
	`# nom-http -h`
	`# nom-http --help`

-s <*host*>:<*port*>
or
--server <*host*>:<*port*>
	Sets the host and, optionally, port on which the SSM listens for API calls. You can omit this construct if the following statements are true:
	• The nom-http client is operating on the same host as the server.
	• Only one service on the host accepts API calls.
	If the above statements are not true, you need to specify port **9090** for SSM.
	The examples that follow demonstrate the syntax to use when entering the **nom-http** command with the **-s** or **--server** construct:
	`# nom-http -s localhost#9090`
	`# nom-http --server localhost#9090`

-f
or
--format <*argument*>
	On the request, sets the Content-Type and Accept headers to a valid content- type. The default setting is JSON.
	The examples that follow demonstrate the syntax to use when entering the **nom-http** command with the **-f** or **--format** construct:
	`# nom-http -u admin -p admin1 -f text/csv`
	`# nom-http -u name -p pass123 --format text/csv`

-i
or
--interactive
	Reads lines of [method] [path] [content] from stdin (standard input). There is rarely a need to explicitly call this argument.
	The examples that follow demonstrate the syntax to use when entering the **nom-http** command with the **-i** or **--interactive** construct:
	`# nom-http -i -s pm01.example.com -u admin \ -p admin1`
	`# nom-http --interactive -s pm02.example.com \ -u name -p password`

Keyword or construct	Description
-p or **--password <*password*>**	Specifies the authentication password; use this construct in conjunction with the **-u** or **–username** constructs. The examples that follow demonstrate the syntax to use when entering the **nom-http** command with the **-p** or **--password** construct: **# nom-http -u admin -p admin1** **# nom-http --username admin --password pw123**
-r or **--raw**	Outputs the response body unparsed and unformatted. The examples that follow demonstrate the syntax to use when entering the **nom-http** command with the **-r** or **--raw** construct: **# nom-http -u admin -p admin1 -r >> \ output_file** **# nom-http --username name --password admin1 \ --raw >> output_file**
--secure	Connects with HTTPS Transport Layer Security (TLS). The example that follows demonstrate the syntax to use when entering the **nom-http** command with the **--secure** construct: **# nom-http -s pm01.example.com -u admin \ -p admin1 --secure**
-shell	Invokes the interactive Groovy scripting shell. The example that follows demonstrate the syntax to use when entering the **nom-http** command with the **--shell** construct: **# nom-http -u admin -p admin1 --shell**
-u or **--user <*userName*>**	Authentication username. The examples that follows demonstrate the syntax to use when entering the **nom-http** command with the **-u** or **--user** construct: **# nom-http -u admin -p admin1** **# nom-http --user name --password pw123**

Examples

The example that follows uses the nom-http client to send an API request to an SSM running on **localhost**; the API request retrieves time zone and address information for a subscriber account called **joe.coffee**. This example specifies the server and port because the **localhost** is running more than one service that can receive commands from the nom-http client:

```
# /usr/local/nom/sbin/nom-http -s localhost:9090 -u admin -p admin1
\ GET /control/joe.coffee
    {
        "id":"joe.coffee",
        "timeZone":"America/Cancun",
        "addresses":["10.1.1.10/8"]
    }
```

The example that follows demonstrates how to send multiple requests in one nom-http command. This command performs the following tasks:

- Adds the SIA Remote (**spson**) service to two subscriber accounts (called **joe.coffee** and **art.gallery**).
- Deletes the SIA Remote service from the account called **ursula.perdevice**.

```
# /usr/local/nom/sbin/nom-http -s localhost:9090 -u admin -p admin1 <<EOF
POST /control/joe.coffee/service {"service":"spson"} POST
/control/art.gallery/service {"service":"spson"} DELETE
/control/ursula.perdevice/service/spson EOF
```

Operating Modes

You can use the nom-http client in interactive and shell mode.

Interactive Mode

In interactive mode, the nom-http client maintains one or more persistent connections to the server for the duration of the session, making the client suitable for sending many API requests efficiently. The following example demonstrates how to access an interactive session:

```
# /usr/local/nom/sbin/nom-http -s localhost:9090 -u admin -p admin1
```

You can send JSON request bodies on multiple lines by putting the opening curly-brace on the first line as demonstrated in the following example:

```
POST /control {
"id": "joe.coffee","time-zone":
"America/Los_Angeles"
}
```

Shell Mode

In shell mode, the nom-http client uses the Groovy scripting language. The shell supports many advanced features. Execute the **help** command to see a list of possible functions. Use the underscore (_) variable to access the result of the previous interpreted statement.

Shell mode supports the following variables for ease of use:

- **http**
- **host**
- **port**
- **user**
- **password**

The **http** object is already set up to connect to the configured server. The **http** object has **get**, **post**, **put**, and **delete** methods which accept a path argument.

The **http** methods return Request Objects. To retrieve the response body, call the **read()** method.

The following example demonstrates how to access shell mode:

```
# /usr/local/nom/sbin/nom-http -s localhost -u admin -p admin1 --shell
groovy:000>
```

The shell provides a **Repeat** class which allows for simple repeated execution of code in a thread. Because the http object is multi-threaded, you can create complex simultaneous queries to the server as demonstrated in the following example:

```
# /usr/local/nom/sbin/nom-http -s localhost -u admin -p admin1 --shell
groovy:000>  t = Repeat.period(1000) { println http.get
("/control/joe.coffee") }
groovy:000>
```

Control is immediately returned to the shell. Because the is assigned to the variable **t**, you can control the task, for example by using **t.cancel()** to stop the task.