


☐

I'm not robot


reCAPTCHA

I'm not robot!

Moodle web service api example

Web service api example.

Moodle 2.0 You need to know how to setup a web service first.

To see the API Documentation, connect as Admin and go to Administration > Plugins > Web services > API Documentation

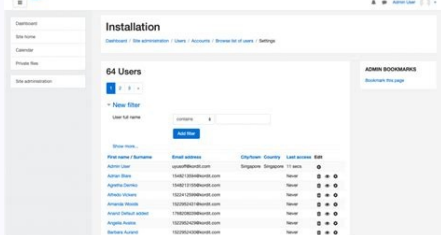
Simple REST request example To quickly test the web service works you can visit the end point from the browser or via curl.

比較

	課題オプライン 評価	CSV Import	XML Grade Publishing	LTi / LIS	Web Service
😊	基本的 柔軟 対話的 容易	基本的 柔軟 対話的	非対話的 自動化可能 とりにあわず 難く	標準化	標準化 クライアント 側の自由度 高い
😞	バルクでない、1活動ずつ、自動化不可。	自動化不可	識別子固定 危険	バルクでない、受験のたびに1名/1活動ずつ	実装・文書化 進行中
認証 認可	Moodle User 認証	Moodle User 認証	IP + key as URL query parameter	OAuth	IP + Key as URL query parameter

29

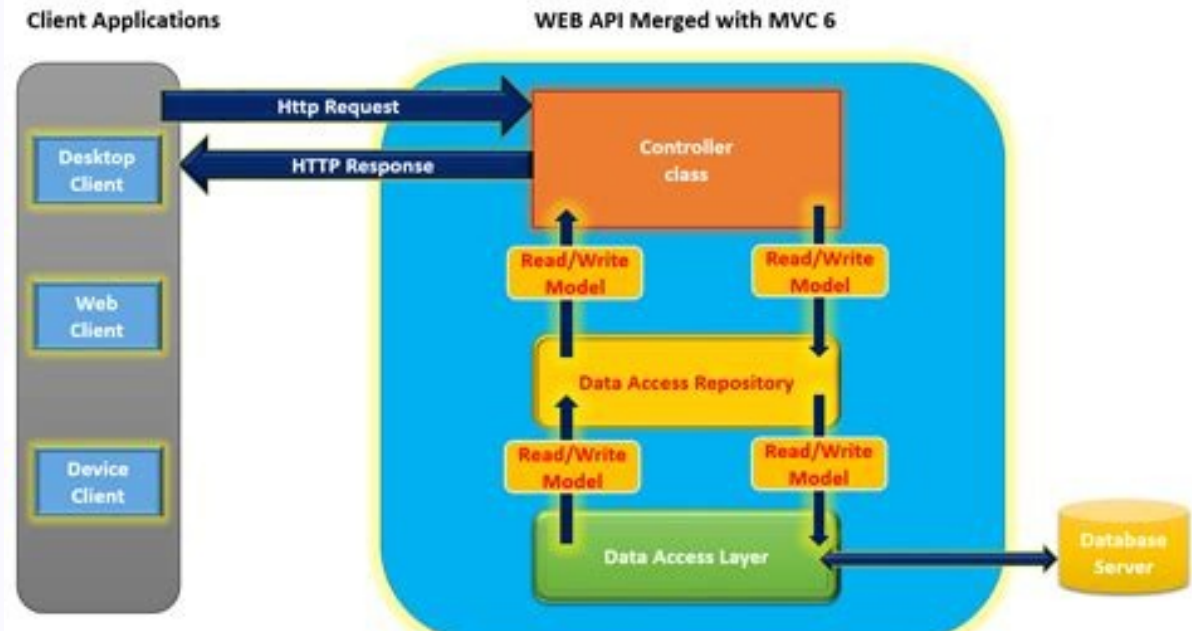
For example to call a function via REST protocol: `curl -i` Customise the parameters `wsurl` and `wsfunction` to match the server side setup. Append additional parameters for the function call as needed with `"?meter_name=param"`, or if your parameter is an array then use `"&array_name[index][parameter_name]=param"`. With the `core` user, create user function's (required) parameters for example: `curl -i ...&moodlewsrestformat=json&wsfunction=core_user_create_users&moodlewsrestformat=json&users[0][username]=tester&users[0][first_name]=Anne&users[0][last_name]=Example...` The `moodlewsrestformat` parameter affects the response format and can be either `xml` (default) or `json`. Officially supported protocols REST The Moodle REST server accepts GET/POST parameters and return XML/JSON values. This server is not RESTful. SOAP The Moodle SOAP server is based on the Zend SOAP server (itself based on the PHP SOAP server). Zend publishes a Zend SOAP client. The current server implementation doesn't fully work with Java/.Net because we didn't generated a fully describe WSDL yet. If you are working on a Java/.Net client, follow or participate to the tracker issues Moodle-28998 / Moodle-28999 XML-RPC The Moodle XML-RPC server is based on Zend XML-RPC server. Zend also publishes a Zend XML-RPC client. AMF (Versions < Moodle 3.0) The Moodle AMF server is based on the Zend AMF server. The test client can be found in Settings blocks > Site Administration > Development > Web service test client > AMF Test client. Demo client examples Demo client sample codes can be downloaded on Github. For HTML5 app creators, you can also find: Node.js apps can use the moodle-client module. A Java Library for REST can be found on Sourceforge. A PHP Library for REST can be found on GitHub. A Go Library for REST can be found on GitHub.



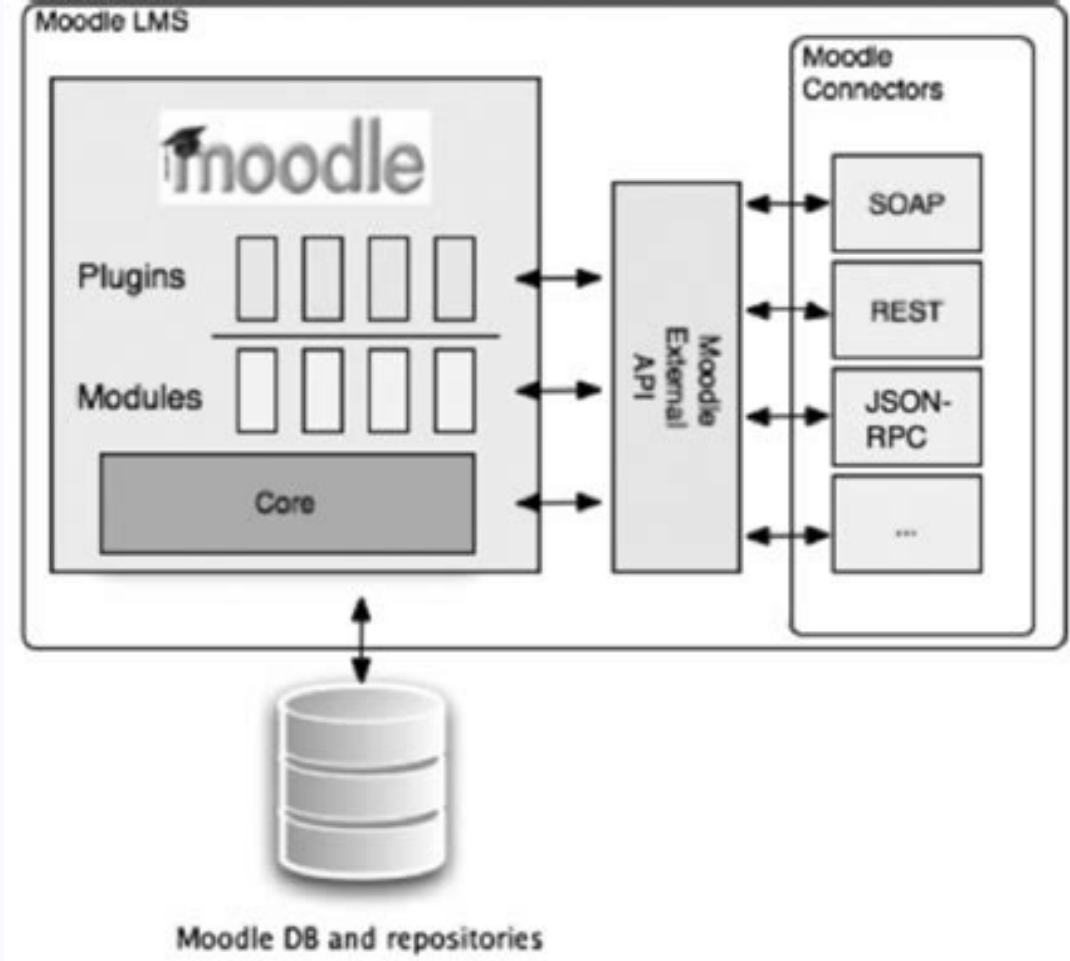
A Python Library for REST can be found on GitHub. How to get a user token Moodle 2.2 Your client can call the script located in /login/token.php with a simple HTTP request. We highly recommend to do it securely with HTTPS.

Moodle側設定手順 (要サイト管理者権限)

- https://docs.moodle.org/28/en/Using_web_services
 - Web Service機能を有効化
 - Web Serviceを定義
 - 教員ユーザに必要な capability を与える
 - 教員ユーザにセキュリティキー(token)を与える

[illegible]

Yes No core_auth core_auth_is_minor 3.5 Requests a check if a user is a digital minor. Yes No core_auth core_auth_request_password_reset 3.4 Requests a password reset. Yes No core_auth core_auth_request_password_reset 3.4 Requests a password reset. Yes No core_auth core_auth_resend_confirmation_email 3.6 Resend confirmation email. Yes No core_auth core_auth_resend_confirmation_email 3.6 Resend confirmation email. Yes No core_backup core_backup_get_async_backup_links 3.7 Gets the data to use when updating the status table row in the UI for when an async backup completes. Yes Yes core_backup core_backup_get_async_backup_links_restore 3.7 Gets the data to use when updating the status table row in the UI for when an async restore completes. Yes Yes core_backup core_backup_get_async_backup_progress 3.7 Get the progress of an Asynchronous backup. Yes Yes core_backup core_backup_get_copy_progress 3.9 Gets the progress of course copy operations. Yes Yes core_backup core_backup_submit_copy_form 3.9 Handles ajax submission of course copy form. Yes Yes core_badges get_user_badges 3.1 Returns the list of badges awarded to a user. No Yes mobile_app core_block_course_blocks 3.3 Returns blocks information for a course.



Yes Yes moodle_mobile app core block core block get dashboard blocks 3.6 Returns block information for the given user dashboard. No Yes moodle_mobile app core blockfetch addable blocks core block fetch addable blocks 3.11 Returns all addable blocks in a given page. Yes Yes moodle_mobile app core blog core blog get_entries 3.6 Returns blog entries. No Yes moodle_mobile app core blog core blog view_entries 3.6 Triggers the blog entries viewed event. Yes No moodle_mobile app core calendar core calendar create_calendar_events 2.5 Create calendar events No Yes moodle_mobile app core calendar core calendar delete_calendar_events 2.5 Delete calendar events Yes Yes moodle_mobile app core calendar core calendar get_action_events by course 3.3 Get calendar action events by course Yes Yes moodle_mobile app core calendar core calendar get_action_events by courses 3.3 Get calendar action events by courses Yes Yes moodle_mobile app core calendar core calendar get_action_events by timesort 3.3 Get calendar action events by timesort Yes Yes moodle_mobile app core calendar core calendar get_allowed_event_types 3.7 Get the type of events a user can create in the given course. No Yes moodle_mobile app core calendar core calendar get_calendar_access_information 3.7 Convenience function to retrieve some permissions/access information for the given course calendar. No Yes moodle_mobile app core calendar core calendar get_calendar_day_view 3.4 Fetch the day view data for a calendar Yes Yes moodle_mobile app core calendar core calendar get_calendar_event_by_id 3.4 Get calendar event by id Yes Yes moodle_mobile app core calendar core calendar get_calendar_events 2.5 Get calendar events No Yes moodle_mobile app core calendar core calendar get_calendar_monthly_view 3.4 Fetch the monthly view data for a calendar Yes Yes moodle_mobile app core calendar core calendar get_calendar_upcoming_view 3.4 Fetch the upcoming view data for a calendar Yes Yes moodle_mobile app core calendar core calendar get_timestamps 3.8 Fetch unix timestamps for given date times. Yes Yes core calendar core calendar submit_create_update_form 3.4 Submit form data for event form Yes Yes moodle_mobile app core calendar core calendar update_event_start_day 3.4 Update the start day (but not time) for an event. Yes Yes moodle_mobile app core calendar core calendar export_token 3.10 Return

[illegible]

tracker issue in the section below.

New Web Services entries are automatically generated for each new major version via this script: API Roadmap Tracker issue: MDL-29934 The Web services API allows you to expose your plugin's functions (usually external functions) as Web services. Once you have done this, your plugin's functions will be accessible to other systems through Web services using one of a number of protocols, like XML-RPC, REST or SOAP. Exposing functions as Web service functions is done in one file called services.php. array(// local_PLUGINNAME_FUNCTIONNAME is the name of the web service function that the client will call. 'classname' => 'component\external\optional\subnamespaces\classname', // create this class in componentdir\classes\external 'methodname' => 'FUNCTIONNAME'. // implement this function into the above class 'description' => 'This documentation will be displayed in the generated API documentation (Administration > Plugins > Webservices > API documentation)', 'type' => 'write', // the value is 'write' if your function does any database change, otherwise it is 'read'. 'ajax' => true, // true/false if you allow this web service function to be callable via ajax 'capabilities' => 'moodle/xxx.yyy, addon/xxx.yyy', // List the capabilities required by the function (those in a require_capability() call) (missing capabilities are displayed for authorised users and also for manually created tokens in the web interface, this is just informative). 'services' => array(MOODLE_OFFICIAL_MOBILE_SERVICE, 'local_mobile') // Optional, only available for Moodle 3.1 onwards. List of built-in services (by shortname) where the function will be included. Services created manually via the Moodle interface are not supported.)); // OPTIONAL // During the plugin installation/upgrade, Moodle installs these services as pre-build services. // A pre-build service is not editable by administrator. \$services = array('MY_SERVICE' => array('functions' => array('local_PLUGINNAME_FUNCTIONNAME'), 'restrictedusers' => 0, // if 1, the administrator must manually select which user can use this service. // (Administration > Plugins > Web services > Manage services > Authorised users) 'enabled'=>1, // if 0, then token linked to this service won't work 'shortname'=>'myservice' //the short name used to refer to this service from elsewhere including when fetching a token)); Don't forget to increment the version number in the version.php file of your plugin whenever services.php changes, otherwise Moodle will not detect the changes.Among other things, that tutorial explains how you define the parameters and return value of your function.