



UTMC: Strategy Control 1.1: 2019

Strategy Interface Specification

10 June 2019
Cover + 23 pages

© Copyright 2019

Revision History

Revision Number	Date	Revision Description
1.0	10 th May 2019	Initial version
1.1	10 th June 2019	Added revision history table. API Specification fixes: <ul style="list-style-type: none">- Security section – fixed 'basicAuth' capitalisation- StrategyStatusPublication - fixed name of required field

Foreword

This document is intended to become part of the UTMC Technical Specifications, through inclusion in the TS.004 UTMC objects registry.

This specification has been developed through a collaboration of Mott MacDonald, Siemens, Dynniq, Idox, Atkins, and Highways England.

List of Contents

1	Introduction.....	5
1.1	Scope.....	5
1.2	Separation of concerns	5
2	Platform-Independent Models.....	6
2.1	Usage scenarios.....	6
2.2	Architectural patterns.....	7
2.3	Platform independent model – publication of strategy status	7
2.4	Platform independent model – update of strategy remote request trigger	10
3	Strategy API.....	12
3.1	Technology choices.....	12
3.2	Strategy API – PUT	12
3.2.1	URL format	12
3.2.2	Request Body	12
3.2.3	Status Codes.....	13
3.2.4	Response Body	13
3.3	Strategy API – GET	14
3.3.1	URL format	14
3.3.2	Request Body	14
3.3.3	Status Codes.....	14
3.3.4	Response	15
3.4	Strategy API Security	16
4	API Specification	17

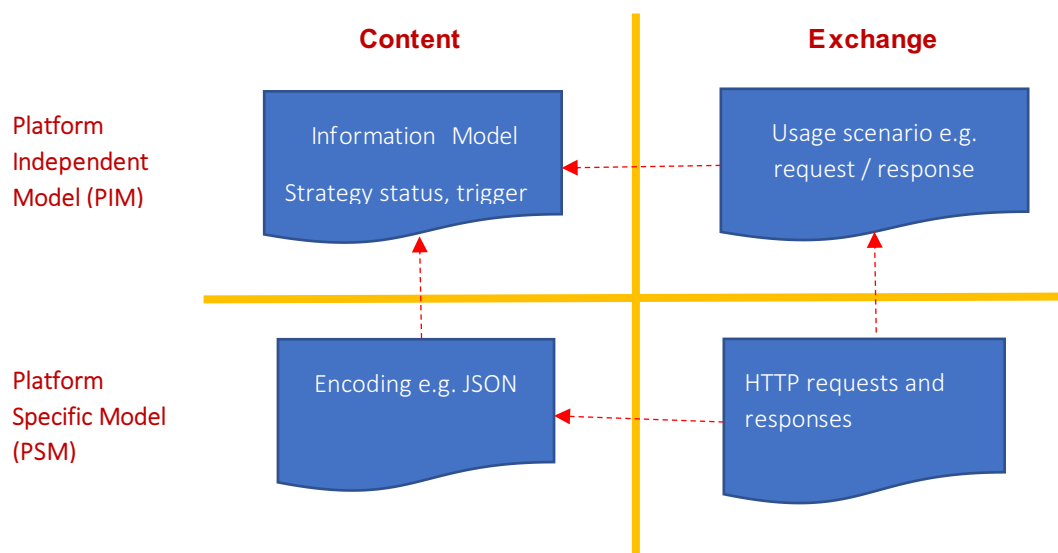
1 Introduction

1.1 Scope

- 1.1.1 This document defines a web-based API (application programming interface) related to traffic management “strategies”.
- 1.1.2 In this specification, the concept of a traffic management “strategy” is modelled on current usage in urban traffic management and control (UTMC) systems. Due to differences in the concept of “strategies” in different UTMC systems, the concept of a strategy in this specification remains at a high level of abstraction. Strategies in UTMC systems have many different features and facilities which are not covered by this cross-system specification. (The UTMC concept of “strategies” pre-dates the publication of the EasyWay Deployment Guideline on Traffic Management Plans which has a different definition of “strategy”)
- 1.1.3 The API allows basic information on strategies to be exchanged, and for one traffic management system to exert an influence over a strategy in a remote system.

1.2 Separation of concerns

- 1.2.1 Following the DATEX II approach (EN 16157-1 and related standards), the interface has been designed by separately considering concerns as shown in the figure.
- 1.2.2 The platform-independent aspects are defined in section 2, and the platform-specific aspects are defined in section 3.



2 Platform-Independent Models

2.1 Usage scenarios

- 2.1.1 It is assumed that prior to usage of this API, the parties owning or operating the traffic management systems have reached agreement on what strategies will be available for remote influence and status publication.
- 2.1.2 This API supports two main system use cases:
- a) A traffic management system provides information on strategies and their status. The information provided to a recipient should be limited to the set of strategies that the recipient is permitted to receive – there may be further strategies in the providing system that are not disclosed to the recipient.
 - b) A traffic management system sends a request to influence a remote strategy. The intent of the request will often be to apply (or stop applying) traffic management actions, but (due to differences in the concept of a strategy in different systems), the request in this specification has a more general meaning, defined using the concept of a “trigger”, as follows.
- 2.1.3 This specification assumes that a strategy has one or more “triggers”, which are conditions that can be evaluated to be true or false, and which may cause a change in state in the strategy when the trigger, or a combination of triggers, changes state.
- 2.1.4 For a strategy which is to be influenced using this API, this strategy shall have the concept of a “remote request trigger”, whose value (“enabled” or “disabled”) can be set by a remote request using this API.
- 2.1.5 The effect of the trigger depends on how the strategy has been configured. The guarantee of this specification is that when a remote request trigger changes state, the strategy shall enter the corresponding state, or take the corresponding action, that has been agreed in prior configuration.
- 2.1.6 Typical usage may be that when a trigger is set (to “enabled”) through this API, the strategy takes some traffic management action if local conditions and other triggers permit this, and when a trigger is cleared (set to “disabled”) that traffic management action ceases or reverts to a normal pattern, but strategies may have more complex state models.
- 2.1.7 This specification assumes that a strategy in a traffic management system is only made available to a single remote traffic management system in addition to the host system, i.e. a third traffic management system is not allowed to make requests concerning the same strategy.
- 2.1.8 This specification uses the terms requester (organisation attempting to influence a strategy in a remote system) and implementer (organisation providing the strategy service to be influenced). These are typically expected to be the organisations performing the traffic management (e.g. local authorities like Kent, Tyne and Wear, etc.), rather than the companies or individuals developing the systems.

2.2 Architectural patterns

- 2.2.1 This specification follows the RESTful pattern for APIs. This influences not only the technology-specific protocol details defined in section 3, but also the choice of platform-independent exchange communications explained in section 2.3.
- 2.2.2 Information models in this specification have used the DATEX II profile as defined in EN 16157-1, which then supports systematic generation of the encodings defined in section 3.
- 2.2.3 Basic datatypes in these information models are as defined in the DATEX II UML model, “Common” namespace (EN 16157-7).

2.3 Platform independent model – publication of strategy status

- 2.3.1 Figure 2-1 shows the interaction sequence when one system makes a request to get the status of all accessible strategies in the remote system, with a synchronous response.

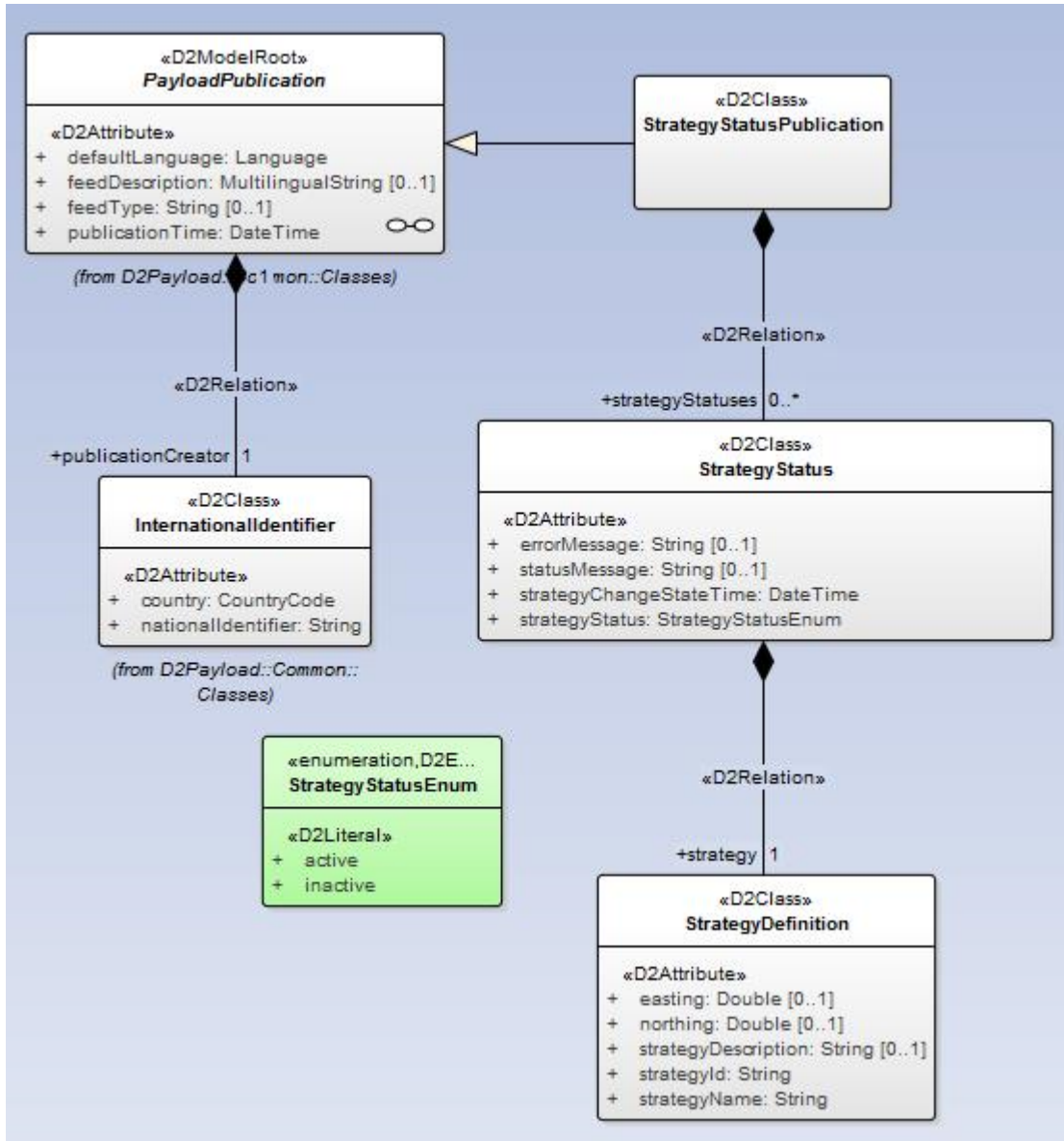
Figure 2-1: Strategy Status Publication sequence diagram



2.3.2 This specification does not include an explicit heartbeat mechanism because the request for strategy status can be issued frequently and used as indication of server health.

2.3.3 Figure 2-2 shows the information model for the content to be returned.

Figure 2-2: Strategy Status Publication information model



2.3.4 The semantics of the PayloadPublication class are defined in the DATEX II UML model, “Common” namespace (EN 16157-7).

StrategyStatusPublication

Publication containing a list of Strategy status.

Field	Description
strategyStatuses	List of strategy status

StrategyStatus

The current status of a Strategy.

Field	Description
strategyStatus	The current status of the Strategy
statusMessage	Information related to the Strategy in its current state (non-error scenarios)
errorMessage	Information related to the Strategy that can be considered an error, e.g. a reason where the Strategy will never become active
strategyChangeStateTime	The time the strategyStatus changed value
strategy	A basic definition of the related Strategy

StrategyDefinition

The definition of a Strategy.

Field	Description
strategyId	The Strategy identifier
strategyName	The name of the Strategy
strategyDescription	A description of the Strategy
easting	Easting coordinate location for Strategy
northing	Northing coordinate location for Strategy

StrategyStatusEnum

A category of status for a strategy.

Enumeration Literal	Description
active	The Strategy is applying non-normal traffic management
inactive	The Strategy is applying either no traffic management or normal condition traffic management

2.4 Platform independent model – update of strategy remote request trigger

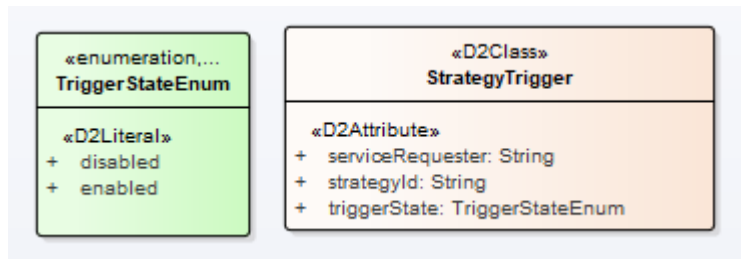
2.4.1 Figure 2-3 shows the interaction sequence when one system requests an update in state for the remote request trigger. In a RESTful sense, the remote request trigger should be viewed as a resource, and the requesting system is updating the value of this resource.

Figure 2-3: Strategy Trigger update sequence diagram



2.4.2 Figure 2-4 shows the information model for the update of the trigger.

Figure 2-4: Strategy Trigger update information model



2.4.3 The *serviceRequester* (and in section 3 the *serviceImplementer*) are based on the String type rather than the standard DATEX II InternationalIdentifier class. This is due to the initial implementation of this standard only being needed to support GB and to keep the URL format of the GET request simple to avoid having GB/SYSA/GB/SYSB.

StrategyTrigger

An update to set the state of the strategy trigger.

Field	Description
strategyId	The Id of the Strategy
serviceRequester	The unique identifier of the requester
triggerState	The state of the trigger to be applied

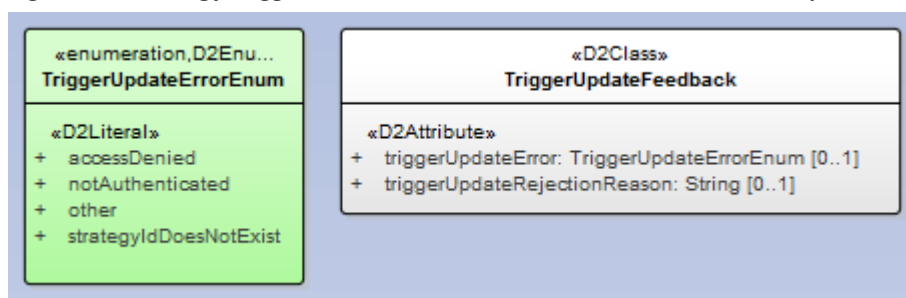
TriggerStateEnum

The state of the trigger.

Enumeration Literal	Description
disabled	The trigger is in the disabled state
enabled	The trigger is in the enabled state

2.4.4 Figure 2-5 shows the information model for a response in the case that the trigger update is not a simple success.

Figure 2-5: Strategy Trigger Feedback information model when not a simple success



TriggerUpdateFeedback

Further information about the rejection of a trigger update.

Field	Description
triggerUpdateError	An indication why the trigger update is rejected
triggerUpdateRejectionReason	Additional feedback text about the reason for rejecting the trigger update

TriggerUpdateErrorEnum

Categorisation of the reason for the rejection of a trigger update.

Enumeration Literal	Description
accessDenied	The supplied strategyId is not accessible by the supplied triggerRequester identifier
notAuthenticated	The supplied triggerRequester identifier is incorrect
other	Other reason for service request failure – specify in feedback reason
strategyIdDoesNotExist	The supplied strategyId does not exist

3 Strategy API

3.1 Technology choices

- 3.1.1 The strategy API shall use HTTP.
- 3.1.2 Where the API requires data other than in the URL that identifies the resource, it shall be encoded in JSON.
- 3.1.3 The JSON format has been systematically generated from the DATEX II-compliant UML model, using the JSON schema mapping defined for DATEX II.
- 3.1.4 A full definition of the API in the form of an OpenAPI document is given in Section 4.
- 3.1.5 The *serviceImplementer* URL parameter used in the following calls is present to satisfy routing needs of some suppliers.

3.2 Strategy API – PUT

This method updates the state of the strategy trigger with the request body.

3.2.1 URL format

URL: /api/utmc/strategy/trigger/{serviceImplementer}/{strategyId}

Parameters:

- serviceImplementer – the name of the service implementer
- strategyId – the Id of the Strategy

3.2.2 Request Body

An example JSON body for the request.

```
{
  "triggerState": {
    "value": "disabled"
  },
  "serviceRequester": "<id of requester>"
}
```

3.2.3 Status Codes

The following status codes apply to this HTTP action.

Status Code	Description
200	Indicates the request completed successfully and the trigger state update has been accepted.
403	Indicates the request was unsuccessful. The response body will contain further details explaining why the request was unsuccessful.

3.2.4 Response Body

If the response is a 200 OK, there will be no response body.

If the response is a 403 Forbidden, a response body will contain further details explaining why the request was unsuccessful.

```
{
  "triggerUpdateError": {
    "value": "accessDenied"
  },
  "triggerUpdateRejectionReason": "Your account
does not have permission to access this strategy"
}
```

3.3 Strategy API – GET

This method retrieves multiple strategy status records for the specified requester identification.

3.3.1 URL format

URL: /api/utmc/strategy/status/{serviceImplementer}/{serviceRequester}

Parameters:

- serviceImplementer – the name of the service implementer
- serviceRequester – unique id of the requester

3.3.2 Request Body

None.

3.3.3 Status Codes

The following status codes apply to this HTTP action.

Status Code	Description
200	Request completed successfully. If a valid query returned no results, the response body contains only an empty Strategy Status Publication.
403	Invalid request details

3.3.4 Response

If the response is a 403 Forbidden, there will be no response body.

If the response is a 200 OK, a response body will contain a Strategy Status Publication.

```
{
  "lang": "en",
  "publicationTime": "2019-03-20T10:54:09.535Z",
  "publicationCreator": {
    "country": "GB",
    "nationalIdentifier": "Highways England National UTMC"
  },
  "strategyStatuses": [
    {
      "strategyStatus": {
        "value": "active"
      },
      "statusMessage": "Strategy applied 6 of 10 measures",
      "strategyChangeStateTime": "2019-03-20T10:54:09.535Z",
      "strategy": {
        "strategyId": "STR00000001",
        "strategyName": "Test Strategy One",
        "strategyDescription": "A description of test strategy one",
        "easting": 111111.1234,
        "northing": 111222.1234
      }
    },
    {
      "strategyStatus": {
        "value": "inactive"
      },
      "errorMessage": "Strategy overridden by operator action",
      "strategyChangeStateTime": "2019-03-20T10:54:09.535Z",
      "strategy": {
        "strategyId": "STR00000002",
        "strategyName": "Test Strategy One",
        "strategyDescription": "A description of test strategy two",
        "easting": 111333.1234,
        "northing": 111444.1234
      }
    }
  ]
}
```

3.4 Strategy API Security

3.4.1 Access via the REST API is restricted by BasicAuth.

3.4.2 Data transmission should use HTTPS.

3.4.3 Specific instantiations of this protocol may agree further measures, such as the use of a virtual private network (VPN).

4 API Specification

The following is the full OpenAPI specification for the Strategy Interface.

```
{
  "openapi": "3.0.0",
  "info": {
    "title": "Strategy API",
    "description": "An API to allow influence of remote Strategies",
    "version": "1.0.0"
  },
  "servers": [
    {
      "url": "/api/utmc/strategy",
      "description": "UTMC Strategy API"
    }
  ],
  "tags": [
    {
      "name": "trigger",
      "description": "Update strategy trigger"
    },
    {
      "name": "status",
      "description": "Access to strategy status"
    }
  ],
  "security": [
    {
      "basicAuth": []
    }
  ],
  "paths": {
    "/trigger/{serviceImplementer}/{strategyId}": {
      "put": {
        "tags": [
          "trigger"
        ],
        "summary": "Update a strategy trigger",
        "operationId": "updateTrigger",
        "parameters": [
          {
            "name": "serviceImplementer",
            "in": "path",
            "description": "the name of the service implementer",
            "required": true,
            "style": "simple",
            "explode": false,
            "schema": {
              "type": "string"
            }
          },
          {
            "name": "strategyId",
            "in": "path",
            "description": "ID of Strategy",
            "required": true,
            "style": "simple",
            "explode": false,
```

```

        "schema": {
            "type": "string"
        }
    },
    ],
    "requestBody": {
        "$ref": "#/components/requestBodies/StrategyTrigger"
    },
    "responses": {
        "200": {
            "description": "The trigger update was accepted"
        },
        "403": {
            "description": "The trigger was not accepted. Authorisation may
have failed, or the strategy provider may have understood the request but
decided not to accept it. The reason will be elaborated in the response body.",
            "content": {
                "application/json": {
                    "schema": {
                        "$ref": "#/components/schemas/TriggerUpdateFeedback"
                    }
                }
            }
        }
    }
},
"/status/{serviceImplementer}/{serviceRequester}": {
    "get": {
        "tags": [
            "status"
        ],
        "summary": "Finds status of Strategies that can be accessed by specified
requester identification",
        "operationId": "getStatus",
        "parameters": [
            {
                "name": "serviceImplementer",
                "in": "path",
                "description": "the name of the service implementer",
                "required": true,
                "style": "simple",
                "explode": false,
                "schema": {
                    "type": "string"
                }
            },
            {
                "name": "serviceRequester",
                "in": "path",
                "description": "unique id of the requester",
                "required": true,
                "style": "simple",
                "explode": false,
                "schema": {
                    "type": "string"
                }
            }
        ]
    },
    "responses": {
        "200": {

```

```

        "description": "successful operation",
        "content": {
            "application/json": {
                "schema": {
                    "$ref": "#/components/schemas/StrategyStatusPublication"
                }
            }
        },
        "403": {
            "description": "Invalid request details"
        }
    }
}
},
"components": {
    "schemas": {
        "String": {
            "type": "string"
        },
        "DateTime": {
            "type": "string",
            "format": "date-time"
        },
        "Double": {
            "type": "number"
        },
        "Language": {
            "type": "string"
        },
        "_ExtensionType": {
            "title": "_ExtensionType",
            "type": "object"
        },
        "CountryCode": {
            "$ref": "#/components/schemas/String"
        },
        "InternationalIdentifier": {
            "type": "object",
            "properties": {
                "country": {
                    "$ref": "#/components/schemas/CountryCode"
                },
                "nationalIdentifier": {
                    "$ref": "#/components/schemas/String"
                },
                "_internationalIdentifierExtension": {
                    "$ref": "#/components/schemas/_ExtensionType"
                }
            }
        },
        "required": [
            "country",
            "nationalIdentifier"
        ]
    },
    "MultilingualString": {
        "type": "object",
        "additionalProperties": false,
        "properties": {
            "values": {

```

```

        "type": "array",
        "items": {
            "$ref": "#/components/schemas/MultiLingualStringValue"
        }
    }
},
"MultiLingualStringValue": {
    "type": "object",
    "additionalProperties": false,
    "properties": {
        "lang": {
            "type": "string"
        },
        "value": {
            "type": "string"
        }
    },
    "required": [
        "lang",
        "value"
    ]
},
"PayloadPublication": {
    "type": "object",
    "properties": {
        "lang": {
            "$ref": "#/components/schemas/Language"
        },
        "feedDescription": {
            "$ref": "#/components/schemas/MultilingualString"
        },
        "feedType": {
            "$ref": "#/components/schemas/String"
        },
        "publicationTime": {
            "$ref": "#/components/schemas/DateTime"
        },
        "publicationCreator": {
            "$ref": "#/components/schemas/InternationalIdentifier"
        },
        "_payloadPublicationExtension": {
            "$ref": "#/components/schemas/_ExtensionType"
        }
    },
    "required": [
        "lang",
        "publicationTime",
        "publicationCreator"
    ]
},
"_StrategyStatusEnum": {
    "type": "object",
    "additionalProperties": false,
    "properties": {
        "value": {
            "$ref": "#/components/schemas/StrategyStatusEnum"
        },
        "_extendedValue": {
            "type": "string"
        }
    }
}

```

```

    },
    "required": [
      "value"
    ]
  },
  "StrategyStatusEnum": {
    "type": "string",
    "enum": [
      "active",
      "inactive",
      "_extended"
    ]
  },
  "_TriggerStateEnum": {
    "type": "object",
    "additionalProperties": false,
    "properties": {
      "value": {
        "$ref": "#/components/schemas/TriggerStateEnum"
      },
      "_extendedValue": {
        "type": "string"
      }
    }
  },
  "required": [
    "value"
  ]
},
"TriggerStateEnum": {
  "type": "string",
  "enum": [
    "disabled",
    "enabled",
    "_extended"
  ]
},
"_TriggerUpdateErrorEnum": {
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "value": {
      "$ref": "#/components/schemas/TriggerUpdateErrorEnum"
    },
    "_extendedValue": {
      "type": "string"
    }
  }
},
"required": [
  "value"
]
},
"TriggerUpdateErrorEnum": {
  "type": "string",
  "enum": [
    "accessDenied",
    "notAuthenticated",
    "other",
    "strategyIdDoesNotExist",
    "_extended"
  ]
},

```

```

"StrategyTrigger": {
  "type": "object",
  "properties": {
    "triggerState": {
      "$ref": "#/components/schemas/_TriggerStateEnum"
    },
    "serviceRequester": {
      "$ref": "#/components/schemas/String"
    },
    "_strategyTriggerExtension": {
      "$ref": "#/components/schemas/_ExtensionType"
    }
  },
  "required": [
    "strategyId",
    "triggerState",
    "triggerRequester"
  ]
},
"TriggerUpdateFeedback": {
  "type": "object",
  "properties": {
    "triggerUpdateError": {
      "$ref": "#/components/schemas/_TriggerUpdateErrorEnum"
    },
    "triggerUpdateRejectionReason": {
      "$ref": "#/components/schemas/String"
    },
    "_triggerUpdateFeedbackExtension": {
      "$ref": "#/components/schemas/_ExtensionType"
    }
  },
  "required": [
    "triggerUpdateStatus"
  ]
},
"StrategyStatusPublication": {
  "type": "object",
  "properties": {
    "strategyStatuses": {
      "type": "array",
      "items": {
        "$ref": "#/components/schemas/StrategyStatus"
      }
    },
    "_strategyStatusPublicationExtension": {
      "$ref": "#/components/schemas/_ExtensionType"
    }
  },
  "required": [
    "strategyStatuses"
  ],
  "allOf": [
    {
      "$ref": "#/components/schemas/PayloadPublication"
    }
  ]
},
"StrategyDefinition": {
  "type": "object",
  "properties": {

```

```

    "strategyId": {
      "$ref": "#/components/schemas/String"
    },
    "strategyName": {
      "$ref": "#/components/schemas/String"
    },
    "strategyDescription": {
      "$ref": "#/components/schemas/String"
    },
    "easting": {
      "$ref": "#/components/schemas/Double"
    },
    "northing": {
      "$ref": "#/components/schemas/Double"
    },
    "_strategyDefinitionExtension": {
      "$ref": "#/components/schemas/_ExtensionType"
    }
  },
  "required": [
    "strategyId",
    "strategyName"
  ]
},
"StrategyStatus": {
  "type": "object",
  "properties": {
    "strategyStatus": {
      "$ref": "#/components/schemas/_StrategyStatusEnum"
    },
    "statusMessage": {
      "$ref": "#/components/schemas/String"
    },
    "errorMessage": {
      "$ref": "#/components/schemas/String"
    },
    "strategyChangeStateTime": {
      "$ref": "#/components/schemas/DateTime"
    },
    "strategy": {
      "$ref": "#/components/schemas/StrategyDefinition"
    },
    "_strategyStatusExtension": {
      "$ref": "#/components/schemas/_ExtensionType"
    }
  },
  "required": [
    "strategyChangeStateTime",
    "strategyStatus",
    "strategy"
  ]
}
},
"requestBodies": {
  "StrategyTrigger": {
    "description": "Object that can be used to update a Strategy Trigger",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/StrategyTrigger"
        }
      }
    }
  }
}

```

```
    }
  },
  "required": true
}
},
"securitySchemes": {
  "basicAuth": {
    "type": "http",
    "scheme": "basic"
  }
}
}
}
```