UrbanPulseManagement API

Overview

API for the management of the configuration of [ui!] UrbanPulse modules

Version information

Version : 1.0

URI scheme

BasePath : /UrbanPulseManagement/api
Schemes : HTTPS

Tags

- catalogue
- category
- clustering
- connector
- event processor
- event type
- health status
- internal
- kpi
- login
- module setup
- permissions
- roles
- sensor
- statement
- status
- user
- version
- virtualsensor

Security

BASIC

Authenticate using basic authorization

Type : basic

Paths

retrieve root hypercat catalogue

```
GET /cat
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string

Responses

HTTP Code	Description	Schema
200	successful operation	HypercatCatalogueTO

Produces

- `application/json; charset=utf-8`

Tags

- catalogue

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**
Unknown	**connector**

register a new category

```
POST /categories
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string

Type	Name	Description	Schema
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Body	**body** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json; charset=utf-8`

Tags

- category

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**
Unknown	**connector**

retrieve registered categories with optional filter by name or sensor ID

```
GET /categories
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Query	**name** *optional*		string
Query	**onlyRoots** *optional*		boolean
Query	**resolveChildren** *optional*		boolean
Query	**sensor** *optional*		string

Responses

HTTP Code	Description	Schema
200	successful operation	< CategoryTO > array

Produces

- `application/json; charset=utf-8`

Tags

- category

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**
Unknown	**connector**

retrieve all root categories

```
GET /categories/root
```

Parameters

Type	Name	Description	Schema	Default
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(username):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string	

Type	Name	Description	Schema	Default
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string	
Query	**resolveChildren** *optional*		boolean	`"false"`

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Produces

- `application/json; charset=utf-8`

Tags

- category

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**
Unknown	**connector**

retrieves a category by id

```
GET /categories/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
200	successful operation	CategoryTO

Produces

- `application/json; charset=utf-8`

Tags

- category

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**
Unknown	**connector**

updates an already existing category

```
PUT /categories/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string
Body	**body** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json; charset=utf-8`

Tags

- category

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**
Unknown	**connector**

deletes a category and updates dependant relationships

```
DELETE /categories/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string

Type	Name	Description	Schema
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Tags

- category

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

retrieves hypercat catalogue

```
GET /categories/{id}/cat
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
200	successful operation	HypercatCatalogueTO

Produces

- `application/json; charset=utf-8`

Tags

- category

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**
Unknown	**connector**

retrieves all child categories of a given parent

```
GET /categories/{id}/children
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string
Query	**resolveChildren** *optional*		boolean

Responses

HTTP Code	Description	Schema
200	successful operation	< CategoryTO > array

Produces

- `application/json; charset=utf-8`

Tags

- category

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**
Unknown	**connector**

retrieves the parent of a given category

```
GET /categories/{id}/parent
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
200	successful operation	CategoryTO

Produces

- `application/json; charset=utf-8`

Tags

- category

Security

Type	Name
basic	**BASIC**

Type	Name
Unknown	**HMAC**
Unknown	**connector**

retrieves the parent of a given category

```
GET /categories/{id}/sensors
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
200	successful operation	< SensorTO > array

Produces

- `application/json; charset=utf-8`

Tags

- category

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**
Unknown	**connector**

register a new connector

```
POST /connectors
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string

Type	Name	Description	Schema
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Body	**body** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json`

Tags

- connector

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

retrieve all registered connectors

```
GET /connectors
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string

Responses

HTTP Code	Description	Schema
200	successful operation	< ConnectorTO > array

Produces

- `application/json; charset=utf-8`

Tags

- connector

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

retrieve a registered connector specified by its id

```
GET /connectors/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
200	successful operation	ConnectorTO

Produces

- `application/json; charset=utf-8`

Tags

- connector

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**
Unknown	**connector**

update a registered connector specified by its id

```
PUT /connectors/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string

Type	Name	Description	Schema
Path	**id** *required*		string
Body	**body** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json`

Tags

- connector

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**
Unknown	**connector**

delete a registered connector specified by its id

```
DELETE /connectors/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Tags

- connector

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Retrieve all registered sensors for the connector with the given ID.

```
GET /connectors/{id}/sensors
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
200	successful operation	< SensorTO > array

Produces

- `application/json; charset=utf-8`

Tags

- connector

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**
Unknown	**connector**

Get the status of the event processor.

```
GET /eprstatus
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Query	**key** *optional*		string

Responses

HTTP Code	Description	Schema
200	successful operation	< string, JsonValue > map

Produces

- `application/json; charset=utf-8`

Tags

- event processor
- status

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

register new event type

```
POST /eventtypes
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Body	**body** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json`

Tags

- event type

Security

Type	Name
basic	**BASIC**

Type	Name
Unknown	**HMAC**
Unknown	**connector**

retrieve all registered event types

```
GET /eventtypes
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string

Responses

HTTP Code	Description	Schema
200	successful operation	< EventTypeTO > array

Produces

- `application/json; charset=utf-8`

Tags

- event type

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**
Unknown	**connector**

retrieve registered event type with given ID

```
GET /eventtypes/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string

Type	Name	Description	Schema
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
200	successful operation	EventTypeTO

Produces

- `application/json; charset=utf-8`

Tags

- event type

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**
Unknown	**connector**

update already registered event type with given ID

```
PUT /eventtypes/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string
Body	**body** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json`

Tags

- event type

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**
Unknown	**connector**

delete event type with given ID

```
DELETE /eventtypes/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Produces

- `application/json; charset=utf-8`

Tags

- event type

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Get the Authorization and UrbanPulse-Timestamp header for further custom use.

```
POST /hasher
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string

Type	Name	Description	Schema
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Body	**body** *optional*		HasherInputTO

Responses

HTTP Code	Description	Schema
200	successful operation	HasherOutputTO

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

query health status of UrbanPulse

```
GET /kpi
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Query	**refresh** *optional*		integer (int32)

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Produces

- `application/json; charset=utf-8`

Tags

- health status
- kpi

Security

Type	Name
basic	**BASIC**

Type	Name
Unknown	**HMAC**

check that the user is existing in Keycloak

```
GET /login
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Produces

- `application/json; charset=utf-8`

Tags

- login

- user

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

shutdown module with given ID

```
POST /moduleSetup/exitModule/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json`

Tags

- clustering
- internal
- module setup

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

get all registered module instances

```
GET /moduleSetup/registrations
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string

Type	Name	Description	Schema
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string

Responses

HTTP Code	Description	Schema
200	successful operation	< UPModuleEntity > array

Produces

- `application/json`

Tags

- clustering
- internal
- module setup

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

get all registered module instances of a certain module type

```
GET /moduleSetup/registrations/{moduleType}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**moduleType** *required*		string

Responses

HTTP Code	Description	Schema
200	successful operation	< UPModuleEntity > array

Produces

- `application/json`

Tags

- clustering
- internal
- module setup

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

clear all module transactions / connections / registrations

```
POST /moduleSetup/reset
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Tags

- clustering

- internal
- module setup

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

clear transactions / connections / registrations for module with given ID

```
POST /moduleSetup/resetModule/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json`

Tags

- clustering
- internal
- module setup

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

send command to module with given ID

```
POST /moduleSetup/sendModuleCommand/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string

Type	Name	Description	Schema
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string
Body	**body** *optional*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json`

Tags

- clustering
- internal
- module setup

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Create a new permission.

```
POST /permissions
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Body	**body** *required*		PermissionTO

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json; charset=utf-8`

Tags

- permissions

Security

Type	Name
basic	**BASIC**

Type	Name
Unknown	**HMAC**

Retrieve all permissions.

```
GET /permissions
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string

Responses

HTTP Code	Description	Schema
200	successful operation	< PermissionTO > array

Produces

- `application/json; charset=utf-8`

Tags

- permissions

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Retrieve the user with the given ID.

```
GET /permissions/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
200	successful operation	PermissionTO

Produces

- `application/json; charset=utf-8`

Tags

- permissions

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Update the permission with the given ID.

```
PUT /permissions/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string

Type	Name	Description	Schema
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string
Body	**body** *required*		PermissionTO

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json`

Tags

- permissions

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Delete the user with the given ID.

```
DELETE /permissions/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Tags

- permissions

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Create a new role.

```
POST /roles
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Body	**body** *required*		RoleWithIds

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json; charset=utf-8`

Tags

- roles

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Retrieve all roles.

```
GET /roles
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string

Responses

HTTP Code	Description	Schema
200	successful operation	< RoleTO > array

Produces

- `application/json; charset=utf-8`

Tags

- roles

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Retrieve the user with the given ID.

```
GET /roles/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
200	successful operation	RoleTO

Produces

- `application/json; charset=utf-8`

Tags

- roles

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Update the role with the given ID.

```
PUT /roles/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string

Type	Name	Description	Schema
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string
Body	**body** *required*		RoleWithIds

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json`

Tags

- roles

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Delete the user with the given ID.

```
DELETE /roles/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Tags

- roles

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Link a permission (if not exists) to a role

```
POST /roles/{id}/permissions/sensors/{SID}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**SID** *required*		string
Path	**id** *required*		string
Body	**body** *required*		ScopesWithOperations

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json; charset=utf-8`

Tags

- roles

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Get all the permission which linked to the role and contains the SID

```
GET /roles/{id}/permissions/sensors/{SID}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string

Type	Name	Description	Schema
Path	**SID** *required*		string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Produces

- `application/json; charset=utf-8`

Tags

- roles

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Delete the given permission link from the role's permission list

```
DELETE /roles/{id}/permissions/{permissionId}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string
Path	**permissionId** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Tags

- roles

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

register a new sensor

```
POST /sensors
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Body	**body** *optional*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json`

Tags

- sensor

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**
Unknown	**connector**

retrieve all registered sensors filtered with the category id

```
GET /sensors
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string

Type	Name	Description	Schema
Query	**category** *optional*		string
Query	**sids** *optional*		string

Responses

HTTP Code	Description	Schema
200	successful operation	< SensorTO > array

Produces

- `application/json; charset=utf-8`

Tags

- sensor

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**
Unknown	**connector**

get a registered sensor specified by its id

```
GET /sensors/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Produces

- `application/json; charset=utf-8`

Tags

- sensor

Security

Type	Name
basic	**BASIC**

Type	Name
Unknown	**HMAC**
Unknown	**connector**

updates an already existing sensor

```
PUT /sensors/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key. If HMAC is used with connector authentication, everything is the same as above; however, the connectors key and its ID is used	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string
Body	**body** *optional*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json`

Tags

- sensor

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**
Unknown	**connector**

delete a registered sensor specified by its id

```
DELETE /sensors/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string

Type	Name	Description	Schema
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Tags

- sensor

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Register new statement

```
POST /statements
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Body	**body** *required*		StatementTO

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json; charset=utf-8`

Tags

- statement

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Retrieve all registered statements

```
GET /statements
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string

Responses

HTTP Code	Description	Schema
200	successful operation	< StatementTO > array

Produces

- `application/json; charset=utf-8`

Tags

- statement

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Retrieve registered statement with given ID

```
GET /statements/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
200	successful operation	StatementTO

Produces

- `application/json; charset=utf-8`

Tags

- statement

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Remove registered statement with given ID. Will only work if every update listener for the given statement has been removed first.

```
DELETE /statements/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string

Type	Name	Description	Schema
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Tags

- statement

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Register new update listener for statement with given ID and authentication information

```
POST /statements/{id}/update-listeners
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string

Type	Name	Description	Schema
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string
Body	**body** *required*	UpdateListeners should contain an "authJson" object instead of the deprecated hmac key. The authJson object looks like this: {"authMethod": "BASIC", "user": "foo", "password": "bar"}	UpdateListenerTO

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json; charset=utf-8`

Tags

- statement

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Retrieve all registered update listeners for statement with given ID

```
GET /statements/{id}/update-listeners
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
200	successful operation	< UpdateListenerTO > array

Produces

- `application/json; charset=utf-8`

Tags

- statement

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Retrieve registered update listener with given ID for statement with given ID

```
GET /statements/{statementId}/update-listeners/{listenerId}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**listenerId** *required*		string
Path	**statementId** *required*		string

Responses

HTTP Code	Description	Schema
200	successful operation	UpdateListenerTO

Produces

- `application/json; charset=utf-8`

Tags

- statement

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Remove registered update listener with given ID for statement with given ID

```
DELETE /statements/{statementId}/update-listeners/{listenerId}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string

Type	Name	Description	Schema
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**listenerId** *required*		string
Path	**statementId** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Tags

- statement

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Create a new user.

```
POST /users
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Body	**body** *required*		UserWithIds

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json; charset=utf-8`

Produces

- `application/json; charset=utf-8`

Tags

- user

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Retrieve all users.

```
GET /users
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string

Responses

HTTP Code	Description	Schema
200	successful operation	< UserTO > array

Produces

- `application/json; charset=utf-8`

Tags

- user

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Retrieve the user with the given ID.

```
GET /users/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
200	successful operation	UserTO

Produces

- `application/json; charset=utf-8`

Tags

- user

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Update the user with the given ID.

```
PUT /users/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string

Type	Name	Description	Schema
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string
Body	**body** *required*		UserWithIds

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json`

Produces

- `application/json; charset=utf-8`

Tags

- user

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Delete the user with the given ID.

```
DELETE /users/{id}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Tags

- user

Security

Type	Name
basic	**BASIC**

Type	Name
Unknown	**HMAC**

Assign a permission (if not exists) to a user to access the given sensor data

```
POST /users/{id}/permissions/sensors/{SID}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**SID** *required*		string
Path	**id** *required*		string
Body	**body** *required*		ScopesWithOperations

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json; charset=utf-8`

Tags

- user

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Get all the permission which linked to the user and contains the SID

```
GET /users/{id}/permissions/sensors/{SID}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string

Type	Name	Description	Schema
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**SID** *required*		string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Produces

- `application/json; charset=utf-8`

Tags

- user

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Delete the given permission link from the user’s permission list

```
DELETE /users/{id}/permissions/{permissionId}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string
Path	**permissionId** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Tags

- user

Security

Type	Name
basic	**BASIC**

Type	Name
Unknown	**HMAC**

Reset the token of your own user.

```
POST /users/{id}/resetKey
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**id** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json; charset=utf-8`

Produces

- `application/json; charset=utf-8`

Tags

- user

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Retrieve version of UrbanPulseManagement

```
GET /version
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string

Responses

HTTP Code	Description	Schema
200	successful operation	string

Produces

- `text/plain`

Tags

- version

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

create a virtual sensor

```
POST /virtualsensors
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string

Type	Name	Description	Schema
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Body	**body** *optional*		VirtualSensorExtendedTo

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json`

Tags

- virtualsensor

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

retrieve all registered virtual sensors by category id and statement name

```
GET /virtualsensors
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user's secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Query	**category** *optional*		string
Query	**resultStatementName** *optional*		string

Responses

HTTP Code	Description	Schema
200	successful operation	< VirtualSensorTO > array

Produces

- `application/json; charset=utf-8`

Tags

- virtualsensor

Security

Type	Name
basic	BASIC
Unknown	HMAC

retrieve a registered virtual sensor by its id

```
GET /virtualsensors/{sid}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**sid** *required*		string

Responses

HTTP Code	Description	Schema
200	successful operation	VirtualSensorTO

Produces

- `application/json; charset=utf-8`

Tags

- virtualsensor

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

delete a registered virtual sensor by id

```
DELETE /virtualsensors/{sid}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd’T’HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**sid** *required*		string

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Produces

- `application/json; charset=utf-8`

Tags

- virtualsensor

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

update a virtual sensor’s targets array

```
PATCH /virtualsensors/{sid}
```

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	UrbanPulse authentication header can have multiple modes. If Basic Auth is used, the value should be in the following format: Basic <Base64-encoded username:password>. If HMAC is used with user authentication, the value should be in the following format: UP base64(user name):hmac256(hash). The hash is calculated over the timestamp + request body (for POST/PUT) or timestamp + request path (for GET/DELETE) using the user’s secret key.	string

Type	Name	Description	Schema
Header	**UrbanPulse-Timestamp** *optional*	The UrbanPulse-Timestamp should be defined in case of HMAC is used as the authorization mode. It has to be provided in the following format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ" (e.g. "2015-05-28T23:54:02.123+0000"). The time zone to use is UTC and the value must not differ more than 15 minutes from the current server time.	string
Path	**sid** *required*		string
Body	**body** *optional*		< string, JsonValue > map

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json`

Tags

- virtualsensor

Security

Type	Name
basic	**BASIC**
Unknown	**HMAC**

Definitions

AuthJsonTO

Name	Schema
authMethod *optional*	string
password *optional*	string
user *optional*	string

CategoryTO

Name	Schema
childCategories *optional*	< string > array
description *optional*	string
id *optional*	string
metadata *optional*	< string > array
name *optional*	string
parentCategory *optional*	string
sensors *optional*	< string > array

ConnectorTO

Name	Schema
backchannelEndpoint *optional*	string

Name	Schema
backchannelKey *optional*	string
description *optional*	string
id *optional*	string
key *optional*	string
sensors *optional*	< string > array

EventTypeExtendedTO

Name	Schema
config *optional*	< string, object > map
description *optional*	< string, object > map
id *optional*	string
name *optional*	string
sensors *optional*	< string > array

EventTypeTO

Name	Schema
config *optional*	string

Name	Schema
description *optional*	string
id *optional*	string
name *optional*	string
sensors *optional*	< string > array

HasherInputTO

Name	Schema
body *optional*	string
method *optional*	string
path *optional*	string
secretKey *optional*	string

HasherOutputTO

Name	Schema
authorizationHeader *optional*	string
timestampHeader *optional*	string

HypercatCatalogueTO

Name	Schema
catalogue-metadata *optional*	< HypercatMetadataTO > array
items *optional*	< HypercatItemTO > array

HypercatItemTO

Name	Schema
href *optional*	string
item-metadata *optional*	< HypercatMetadataTO > array

HypercatMetadataTO

Type : object

JsonValue

Name	Schema
valueType *optional*	enum (ARRAY, OBJECT, STRING, NUMBER, TRUE, FALSE, NULL)

PermissionTO

Name	Description	Schema
id *optional*	UUID - optional on POST and PUT requests; must match the path parameter if given in PUT request, must not exist yet if given in POST request	string
name *required*		string

RoleTO

Name	Description	Schema
id *optional*	UUID - optional on POST and PUT requests; must match the path parameter if given in PUT request, must not exist yet if given in POST request	string
name *required*		string
permissions *optional*		< PermissionTO > array

RoleWithIds

Name	Schema
name *required*	string
permissions *optional*	< string > array

ScopesWithOperations

Name	Schema
operation *optional*	< string > array
scope *optional*	< string > array

SensorTO

Name	Schema
categories *optional*	< string > array
description *optional*	string
eventType *optional*	string

Name	Schema
id *optional*	string
location *optional*	string
senderid *optional*	string

StatementTO

Name	Schema
comment *optional*	string
id *optional*	string
name *optional*	string
query *optional*	string

UPModuleEntity

Name	Description	Schema
id *optional*		string
lastHeartbeat *optional*		string (date-time)
mailSent *optional*	**Default** : `false`	boolean
moduleState *optional*		enum (HEALTHY, UNSTABLE, UNHEALTHY, UNKNOWN)

Name	Description	Schema
moduleType *required*		string

UpdateListenerTO

Name	Schema
authJson *optional*	AuthJsonTO
id *optional*	string
key *optional*	string
statementId *optional*	string
target *optional*	string

UserTO

Name	Description	Schema
id *optional*	UUID - optional on POST and PUT requests; must match the path parameter if given in PUT request, must not exist yet if given in POST request	string
name *optional*	User name - optional on PUT, but not on POST	string
password *optional*	Password - write only (will not be returned in GET requests)	string
permissions *optional*	User permissions in addition to the ones added by the user's roles; full JSON objects (including ID and name)	< PermissionTO > array
roles *optional*	User roles as full JSON objects (including ID and name)	< RoleTO > array

Name	Description	Schema
secretKey *optional*	SecretKey - hidden field	string

UserWithIds

Name	Schema
name *optional*	string
password *optional*	string
permissions *optional*	< string > array
roles *optional*	< string > array

VirtualSensorExtendedTo

Name	Schema
category *optional*	string
description *optional*	< string, object > map
eventTypes *optional*	< EventTypeExtendedTO > array
resultEventType *optional*	EventTypeExtendedTO
statements *optional*	< StatementTO > array
targets *optional*	< string > array

VirtualSensorTO

Name	Schema
categoryId *optional*	string
description *optional*	string
id *optional*	string
resultEventTypeId *optional*	string
resultStatementId *optional*	string
sid *optional*	string
targets *optional*	string