

Xenometric

REST API

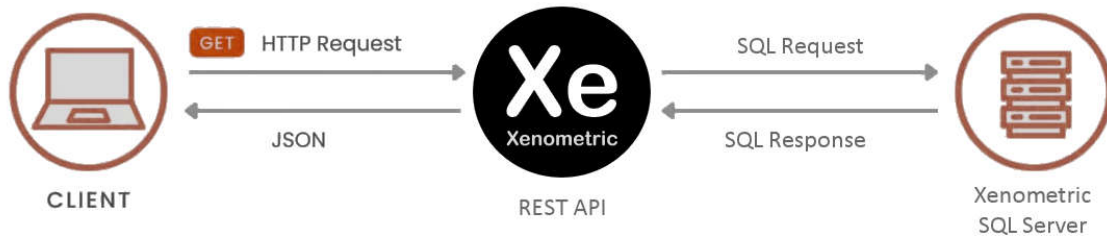
Contents

| | |
|-----------------------------|---|
| 1 Introduction | 3 |
| 2 REST Commands | 4 |
| 2.1 Locations Command | 5 |
| 2.2 History Command | 6 |
| 2.3 Live Command | 8 |
| 3 Swagger | 9 |

1 Introduction

A REST API is an application programming interface (API) that conforms to the design principles of the representational state transfer (REST) architectural style, a style used to connect distributed systems. REST APIs are sometimes referred to as RESTful APIs.



A REST API allows the user (client) to talk to an information provider (server) in a secure and predictable way. Xenometric's API allows customers to request data from our database, without the need to connect directly to the database. This provides a safe and standard way for customers to access their data with technology that they may already employ.





2 REST Commands

We only support GET commands, as our API is designed to allow access to your data. We do not accept POST, DELETE or PUT commands.



We support three different GET commands. The first is *Locations*. This command allows you to list all the locations that you can access with your username/password or API key.

GET `/xenorestapi/locations` Location Method  

The second is *History*, which allows you to collect all your count data for all periods since we started collecting your sensors' data.

GET `/xenorestapi/history` History Method  

Finally, there is the *Live* command. This allows you to see the real-time visitor counts and derived metrics, such as occupancy.

GET `/xenorestapi/live` Live Method  

2.1 Locations Command

The Locations command has one parameter.

| Parameter | Description | Example |
|-----------------|---|--|
| location | A location code that describes the parent of the locations you wish to list. The location can be a Group, Site or Zone (G = group, S = site and Z = entrance) specified by a letter followed by number. | G19 would be a valid location for group number 19. S25 would be a valid location for site number 25. |

Here is a curl example of the Location command.

```
curl "https://url_of_rest_api:port/xenorestapi/locations?location=G49" -H "Authorization: Basic VGhpcyBpcyB0aGUgYmFzZTY0IGVuY29kZWQgQVBJEtleQ==" -k
```

url_of_rest_api This is the IP address or URL of the server hosting the REST API.

port This is the port that the REST API is listening on.

G49 This is the code for the group number 49.

VGhpcyBpcyB0aGUgYmFzZTY0IGVuY29kZWQgQVBJEtleQ== This is the base 64 encoded API Key.

Here is an example of the JSON that is returned.

```
[
  {
    "location": "Demo Shopping Mall Group",
    "locationtype": "group",
    "locationcode": "g49"
  },
  {
    "location": "Demo Mall Dubai",
    "localtime": "2026-04-17T18:56:01+04:00",
    "locationtype": "site",
    "locationcode": "s246"
  },
  {
    "location": "First Floor Entrance Female",
    "localtime": "2026-04-17T18:56:01+04:00",
    "locationtype": "entrance",
    "locationcode": "e516",
    "entrancetype": "external"
  },
  {
    "location": "First Floor Entrance Male",
    "localtime": "2026-04-17T18:56:01+04:00",
    "locationtype": "entrance",
    "locationcode": "e515",
    "entrancetype": "external"
  }
]
```

2.2 History Command

The History command has six parameters.

| Parameter | Description | Example |
|----------------------|---|--|
| location | A location code that describes the parent of the locations you wish to list. The location can be a Group, Site or Zone (G = group, S = site, Z = zone and E = entrance) specified by a letter followed by a number. | G19 would be a valid location for group number 19. S25 would be a valid location for site number 25. |
| datatype | Valid datatypes are vis, occ and occvis. vis is the visitation data (in counts). occ is the occupancy data. occvis is both occupancy and visitation data. | vis would provide just the visitor counts. |
| locationtypes | Locations types to be shown in the results set. You can specify group (g), sites (s), zones (z) and entrances (e). To include sites use s. To include entrances, use e. To include sites and entrances, use se | g, s, z, e or any combination. E.g. gse |
| timeinterval | Breakdown of the reported time interval in the returned JSON object. d = daily breakdown, h = hourly breakdown. | d or h |
| start | Start time of the range in ISO8601 format YYYY-MM-DDThh:mm:ss | 2026-04-09T00:00:00 |
| end | End time of the range in ISO8601 format YYYY-MM-DDThh:mm:ss | 2026-04-10T00:00:00 |

Here is a curl example of the Location command.

```
curl
"https://url_of_rest_api:port/xenorestapi/history?location=G49&datatype=occvis&locationtypes=gse&timeinterval=d&start=2026-04-09T00:00:00&end=2026-04-10T00:00:00" -H "Authorization: Basic
VGhpcyBpcyB0aGUgYmFzZTY0IGVuY29kZWQgQVBJIETleQ==" -k
```

url_of_rest_api This is the IP address or URL of the server hosting the REST API.
port This is the port that the REST API is listening on.
G49 This is the code for the group number 49.
occvis This code instructs the API to return visitor and occupancy data.
gse A location types of gse returns the data for the group, sites and entrances.
d The results will contain the data by day.
2026-04-09T00:00:00 The start date of midnight on the 9th April 2026.
2026-04-10T00:00:00 The end date of midnight on the 10th April 2026.
VGhpcyBpcyB0aGUgYmFzZTY0IGVuY29kZWQgQVBJIETleQ== This is the base 64 encoded API Key.

Here is an example of the JSON that is returned.

```
[
{
  "location": "Demo Shopping Mall Group",
  "locationtype": "group",
  "locationcode": "g49",
  "counts": [
    {
      "localtime": "2026-04-09",
      "visitors_in": "0"
    }
  ]
}
```

```
},
{
  "location": "Demo Mall Dubai",
  "locationtype": "site",
  "locationcode": "s246",
  "counts": [
    {
      "localtime": "2026-04-09",
      "visitors_in": "0",
      "occupancy": "0"
    }
  ]
},
{
  "location": "Main Entrance Female",
  "locationtype": "entrance",
  "locationcode": "e513",
  "entrancetype": "external",
  "counts": [
    {
      "localtime": "2026-04-09",
      "visitors_in": "0"
    }
  ]
},
{
  "location": "Main Entrance Male",
  "locationtype": "entrance",
  "locationcode": "e511",
  "entrancetype": "external",
  "counts": [
    {
      "localtime": "2026-04-09",
      "visitors_in": "0"
    }
  ]
}
]
```

2.3 Live Command

The History command has six parameters.

| Parameter | Description | Example |
|------------------|---|---|
| location | A location code that describes the parent of the locations you wish to list. The location can be a Group, Site or Zone (G = group, S = site, Z = zone and E = entrance) specified by a letter followed by a number. | S246 would be a valid location for site number 246. |
| datatype | Valid datatypes are vis, occ and occvis. vis is the visitation data (in counts). occ is the occupancy data. occvis is both occupancy and visitation data. | vis would provide just the visitor counts. |
| divisions | Locations types to be shown in the results set. You can specify group (g), sites (s), zones (z) and entrances (e). To include sites use s. To include entrances, use e. To include sites and entrances, use se | g, s, z, e or any combination. E.g. gse |

Here is a curl example of the Location command.

```
curl "https://url_of_rest_api:port/xenorestapi/live?location=S246&datatype=occvis&divisions=s" -H "Authorization: Basic VGhpcyBpcyB0aGUgYmFzZTY0IGVuY29kZWQgQVBJIEtleQ==" -k
```

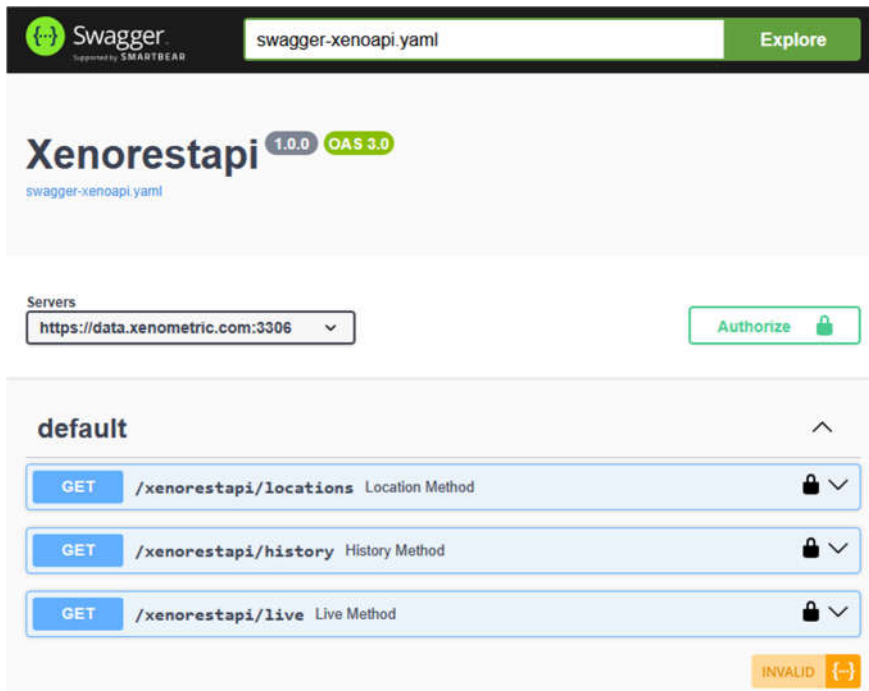
url_of_rest_api This is the IP address or URL of the server hosting the REST API.
port This is the port that the REST API is listening on.
S246 This is the code for the site number 246.
occvis This code instructs the API to return visitor and occupancy data.
s A location types of s returns the data for the site.
d The results will contain the data by day.
VGhpcyBpcyB0aGUgYmFzZTY0IGVuY29kZWQgQVBJIEtleQ== This is the base 64 encoded API Key.

Here is an example of the JSON that is returned.

```
[
  {
    "location": "Demo Mall Dubai",
    "locationtype": "site",
    "locationcode": "s246",
    "localtime": "2026-04-17T19:51:35+04:00",
    "occupancy": "0",
    "occupancylimit": "100",
    "visitors": "0"
  }
]
```

3 Swagger

To get to our Swagger page, login to your Xenometric reporting with your username and password. There is a link to Swagger at bottom of the page.



Within the Swagger page you can test the Xenometric REST API.

Firstly, you need to login by clicking on the Authorize button. Use your reporting username and password and click Authorize.

Available authorizations

basicAuth (http, Basic)

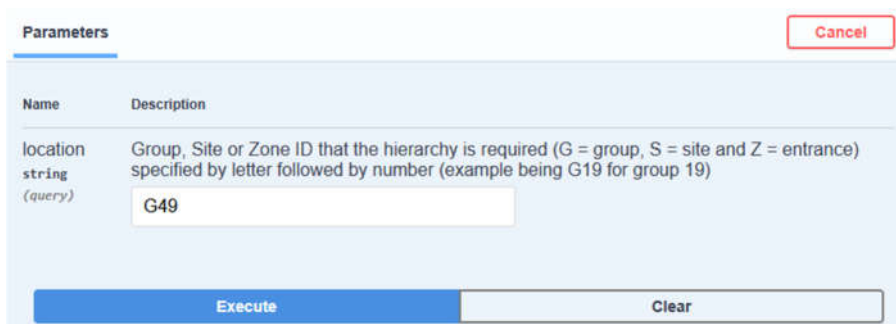
Username:

Password:

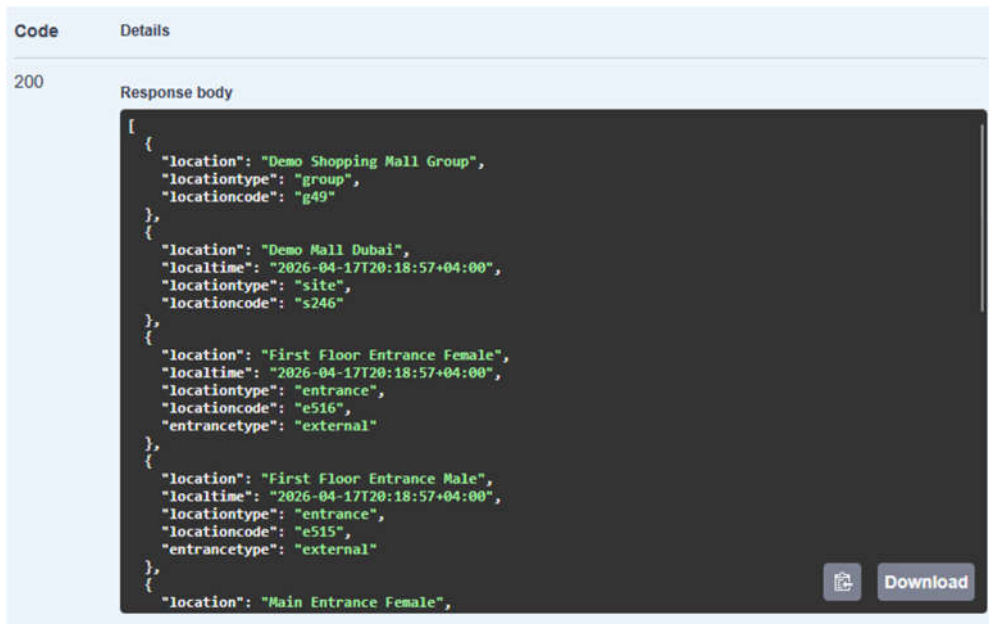
Authorize

Close

You can now run any of the API commands. Choose a command from Locations, History or Live and then click the Try it out button. Provide the required parameters, such as G49 for the Locations command. And click Execute/



You should see the results returned within a few seconds.



The screenshot shows a REST client interface with a 'Code' tab selected. The status is '200' and the 'Response body' is displayed as a JSON array. The array contains five objects, each representing a location with fields for location, localtime, locationtype, and locationcode. The last object in the array is partially visible.

```
[
  {
    "location": "Demo Shopping Mall Group",
    "locationtype": "group",
    "locationcode": "g49"
  },
  {
    "location": "Demo Mall Dubai",
    "localtime": "2026-04-17T20:18:57+04:00",
    "locationtype": "site",
    "locationcode": "s246"
  },
  {
    "location": "First Floor Entrance Female",
    "localtime": "2026-04-17T20:18:57+04:00",
    "locationtype": "entrance",
    "locationcode": "e516",
    "entrancetype": "external"
  },
  {
    "location": "First Floor Entrance Male",
    "localtime": "2026-04-17T20:18:57+04:00",
    "locationtype": "entrance",
    "locationcode": "e515",
    "entrancetype": "external"
  },
  {
    "location": "Main Entrance Female",
```

If there is a problem, you will see a response code and error message. An example of an error would be 403 Unauthorized. This indicates that your user account does not have access to the requested location.