

zebrix REST api starter guide

Introduction

Every zebrix REST API call required a token that has to be included in the request header. The token can be obtained with a login procedure.

login api

POST /login

Parameters has to be sent in the JSON format

| Parameters | |
|------------|-------------------|
| clientname | the customer name |
| username | the username |
| password | the user password |

login demo with "POSTMAN" (Google Chrome Extension)

Call

The screenshot shows the Postman interface with a POST request to `https://cmsv2.zebrix.net/login`. The Body tab is selected, and the raw JSON data is:

```
1 {  
2   "clientname": "customer",  
3   "username": "user",  
4   "password": "UserPassw0rd"  
5 }  
6
```

Return

```

https://cmsv2.zebrix.net/login
POST https://cmsv2.zebrix.net/login
Params
Send Save
Body Cookies Headers (10) Tests
Status: 200 OK Time: 86 ms
Pretty Raw Preview JSON
[{"token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9 .eyJpZCI6MjYwMCwiQ2xpZW50SWQiOjIwLCJyb2xIjoyLCJsZXZlbCI6MTAsInNjb3B1Ijp7fSwiaWF0IjoxNDcwMTIyNDU2LCJleHAIoJE0NzAxNjU2NTZ9.YSqeMUJUmlt2KHzlidcwZzA7XS52SR8WiUHWnH4E", "user": { "id": 2600, "username": "user", "level": 10, "name": "user", "email": "none@none.com", "lang": "fr_FR", "favorites": { "media": [], "pages": [], "playlists": [] }, "client": { "id": 20, "name": "manganelli", "lang": "fr_FR", "fonts": [], "hasDatasource": true, "reseller": { "name": "Manganelli", "presentation": "<img src=\"/logos/zebrix.jpg\"/>\n<br>\n<h2>Support</h2>\n<p><i class=\"fa fa-clock-o\"\></i>&ampnbspLundi au vendredi de 8h00 à 19h00</p>\n<p><i class=\"fa fa-phone\"\></i>&ampnbsp+33 3 20 41 59 69 (Français/English)</p>\n<p><i class=\"fa fa-envelope\"\></i>&ampnbsp<a href=\"mailto:support@manganelli-ds.com\">support@manganelli-ds.com</a></p>\n<p><i class=\"fa fa-globe\"\></i>&ampnbsp<a target=\"_blank\" href=\"http://support.manganelli-ds.com\">http://support.manganelli-ds.com</a></p>\n<br>\n<h2>Contact commercial</h2>\n<p><i class=\"fa fa-phone\"\></i>&ampnbsp+33 3 20 41 33 66</p>\n<p><i class=\"fa fa-envelope\"\></i>&ampnbsp<a href=\"mailto:contact@manganelli-ds.com\">contact@manganelli-ds.com</a></p>\n<p><i class=\"fa fa-globe\"\></i>&ampnbsp<a target=\"_blank\" href=\"http://www.manganelli-ds.com\">http://www.manganelli-ds.com</a></p>\n", "logo_path": "/logos/zebrix.jpg" } }, "scope": {} }

```

Example of GET API on screen

GET /api/screen/:id

This example will demonstrate how to request information with a GET call to the API. We will use the screen api (to get the list of all screens with their properties). It will work the same for all other objects in zebrix (media, pages, playlist, etc.)

In this request we will need to use the previously requested token in the request header of the API will return a 401 (not authorized) error.

The token must be prefixed with the word “Bearer” and a space char that way :

Bearer

eyJ0eXAiOiJKV1QiLCJhbGciOikLIUzI1NiJ9.eyJpZCI6MjYwMCwiQ2xpZW50SWQiOjIwLCJyb2xIjoyLCJsZXZlbCI6MTAsInNjb3B1Ijp7fSwiaWF0IjoxNDcwMTIyNDU2LCJleHAIoJE0NzAxNjU2NTZ9.YSqeMUJUmlt2KHzlidcwZzA7XS52SR8WiUHWnH4E

The call of the API will return a JSON array of all screens :

The screenshot shows a POSTMAN interface with the following details:

- Method:** GET
- URL:** https://cmsv2.zebrix.net/api/screen
- Headers (1):** Authorization (set to Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6.)
- Body:** (Pretty, Raw, Preview, JSON)
- Response:** Status: 200 OK, Time: 107 ms

```
1 [ { 2   "status_detail": { 3     "network": 2, 4     "panel": 0, 5     "transfer": 2, 6     "command": 0, 7     "ip": "" 8   }, 9   "status": "offline", 10  "content": {}, 11  "state": { 12    "panel": false, 13    "lock": false, 14    "mute": false, 15    "volume": "0" 16  }, 17  "id": 1023, 18  "hardId": "90f1aac093a4", 19  "name": "DB48D", 20  "model": "sssp", 21  "serial": "N/A", 22  "screenModel": "NUC", 23  "ContentId": 4836, 24  "contentType": "page", 25  "useScheduling": true, 26  "logger": false, 27  "activationDate": "2015-04-07T16:17:17.000Z", 28  "locked": false, 29  "initialized": true, 30  "hidden": false, 31  "ClientId": 20, 32  "SiteId": null, 33 }
```

To get information for a specific screen, its unique id can be added in the URL. For example, we want to get information about the screen with the id 6361 :

```

1 [
2   "status_detail": {
3     "network": 2,
4     "panel": 0,
5     "transfer": 2,
6     "command": 0,
7     "ip": ""
8   },
9   "status": "offline",
10  "content": {
11    "contentType": "page",
12    "ContentId": 34641
13  },
14  "state": {
15    "panel": true,
16    "lock": false,
17    "mute": true,
18    "volume": 0,
19    "orientation": "h"
20  },
21  "id": 6361, -----^
22  "hardId": "90f1aa566d1a",
23  "name": "D848D Showroom",
24  "maxResolution": null,
25  "orientation": null,
26  "lang": "en_EN",
27  "model": "sssp",
28  "serial": "ZCMLHSLF600207T",
29  "jsbuild": "v2.0.20",
30  "screenModel": "\u0000\u0014\u0000\u0000",
31  "firmware": "T-GFSLDDWNC-1025.3",
32  "ContentId": 34641,
33  "contentType": "page",
34  "useScheduling": false,
35  "ipAddress": "92.103.87.162",
36  "logger": false,
37  "port": "80",
38  "jsmode": "nativ",
39  "activationDate": "2016-06-27T13:09:19.000Z",
40  "level": 10,
41  "locked": false,
42  "initialized": true,

```

How to set content on a screen?

Basically, you will have to post the following JSON

```
{'contentType': 'page', 'ContentId': contentId, 'useScheduling': False}
```

* **contentType (string)**: possible value « page » and « playlist »

- **contentId (integer)** : id of the content you want to set
- **useScheduling (boolean)** : false is mandatory when setting an arbitrary content.

To go back to schedule mode, you only need to POST the following JSON

```
{'useScheduling': True}
```

to the following URL:

```
POST : https://cmsv2.zebrix.net/api/screen/{screenID}/setContent
```

To get ID of pages and playlist, you can make a GET request to following URLs:

- /api/screen
- /api/page
- /api/playlist

How to use API for datasource

Introduction to datasource

In zebrix a **datasource** is basically one row from a database.

As an example let take a table from DB containing information about 4 meeting rooms. Each meeting room has 4 properties :

- ROOM (room name)
- VISITOR (current visitor who uses the meeting room)
- SEATS
- HASBEAMER

| | A | B | C | D |
|---|-------------|---------|-------|-----------|
| 1 | ROOM | VISITOR | SEATS | HASBEAMER |
| 2 | Yellow room | MORA | 6 | true |
| 3 | Dark Room | BARCO | 0 | false |
| 4 | White Room | Zebrix | 8 | false |
| 5 | Black Room | Audi | 10 | true |
| 6 | | | | |

In zebrix this table of 4 meeting rooms will requires 4 datasources (i.e. meeting room 1, meeting room 2, meeting room 3, meeting room 4). Each line/datasource will contains all columns / fields in a JSON.

Get all datasources

GET /api/datasource/:id

All datasources will be returned with all their properties (including their content)

Using our meeting rooms example, here is the result you will get :

```
111 },
112   "defaults": {
113     "ROOM": "Black Room",
114     "VISITOR": "Audi",
115     "SEATS": "10",
116     "HASABEAMER": "true"
117   },
118   "id": 1010,
119   "name": "test14",
120   "createdAt": "2016-04-29T14:53:26.000Z",
121   "updatedAt": "2016-04-29T14:57:11.000Z",
122   "ClientId": 20,
123   "TaxoId": null,
124   "taxo": null
125 },
126 {
127   "defaults": {
128     "ROOM": "Yellow room",
129     "VISITOR": "MORA",
130     "SEATS": "6",
131     "HASABEAMER": "true"
132   },
133   "id": 1011,
134   "name": "test21",
135   "createdAt": "2016-04-29T14:59:58.000Z",
136   "updatedAt": "2016-04-29T14:59:58.000Z",
137   "ClientId": 20,
138   "TaxoId": null,
139   "taxo": null
140 }
```

These datasources can be used by users in their page by following instructions on that page (section 5) ([Using datasources in Zebrix](#))

Please notice that some datasources are auto-generated by zebrix when the functionality of “zone with variable content” is used. These datasources can be easily recognized with their name which always begin with double underscore + page + id of the concerned page.





How to add a new datasource to zebrix

POST /api/datasource/

The content of the datasource has to be JSON encoded and put as an object in a defaults variable.

```

1 { 
2   "defaults":{ 
3     "companyName":"zebrix",
4     "trainingTopic":"zebrix api usage",
5     "trainer":"Pierre",
6     "duration":"3 hours",
7     "misc":"test"
8   }
9 }
```

Example of usage :

The screenshot shows the Postman interface with a POST request to <https://cmcsv2.zebrix.net/api/datasource>. The Body tab is selected, showing a raw JSON payload:

```

1 {  
2   "defaults": {  
3     "companyName": "zebrix",  
4     "trainingTopic": "zebrix api usage",  
5     "trainer": "Pierre",  
6     "duration": "2 hours"  
7   }  
8 }

```

A red annotation next to the JSON payload says: "JSON which is POSTed to the API".

Below the request, the response tab is selected, showing the returned JSON object:

```

1 {  
2   "defaults": {  
3     "companyName": "zebrix",  
4     "trainingTopic": "zebrix api usage",  
5     "trainer": "Pierre",  
6     "duration": "2 hours"  
7   },  
8   "id": 1259,  
9   "ClientId": 20,  
10  "updatedAt": "2016-08-02T08:56:24.000Z",  
11  "createdAt": "2016-08-02T08:56:24.000Z"  
12 }

```

A red annotation next to the JSON object says: "Datasource JSON which is returned after the add".

How to update an existing datasource

POST /api/datasource/:id

The screenshot shows the Postman interface with a POST request to <https://cmsv2.zebrix.net/api/datasource/1259>. A red arrow points to the URL with the text "datasource id is needed for the update". The "Body" tab is selected, showing JSON data:

```

1 {  
2   "defaults": {  
3     "companyName": "zebrix",  
4     "trainingTopic": "zebrix api usage",  
5     "trainer": "Pierre",  
6     "duration": "3 hours"  
7   }  
8 }

```

Below, the "Body" tab is selected again, showing the same JSON data in Pretty, Raw, Preview, and JSON formats.

```

1 {  
2   "defaults": {  
3     "companyName": "zebrix",  
4     "trainingTopic": "zebrix api usage",  
5     "trainer": "Pierre",  
6     "duration": "3 hours"  
7   },  
8   "id": 1259,  
9   "name": null,  
10  "createdAt": "2016-08-02T08:56:24.000Z",  
11  "updatedAt": "2016-08-02T08:58:32.000Z",  
12  "ClientId": 20,  
13  "TaxoId": null,  
14  "taxo": null  
15 }

```

The update process works the same as a creation but the id of the datasource need to be specified.

How to force screens to update

POST /api/screens/updateds

When a datasource is updated screens that display this datasource will not be update until the updateds is called. This API is usually called after update of a datasource. If displayed pages contain multiple datasources it is adviced to call the updateds method at the end of the update process of all datasources instead of after every datasource update to limit ressources and bandwidth usage.

From:

<https://documentation.zebrix.net/> - **zebrix documentation**



Permanent link:

<https://documentation.zebrix.net/doku.php?id=en:zebrixrestapi&rev=1542380667>

Last update: **2020/06/22 11:53**