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

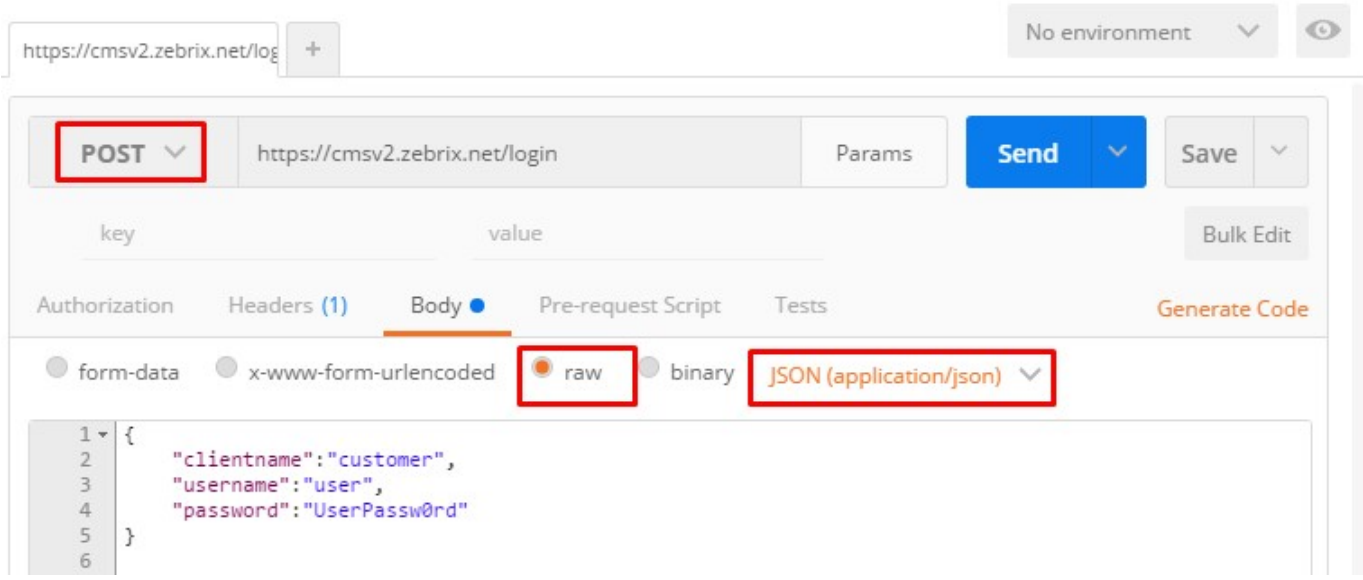
POST /login

Parameters has to be sent in the JSON format

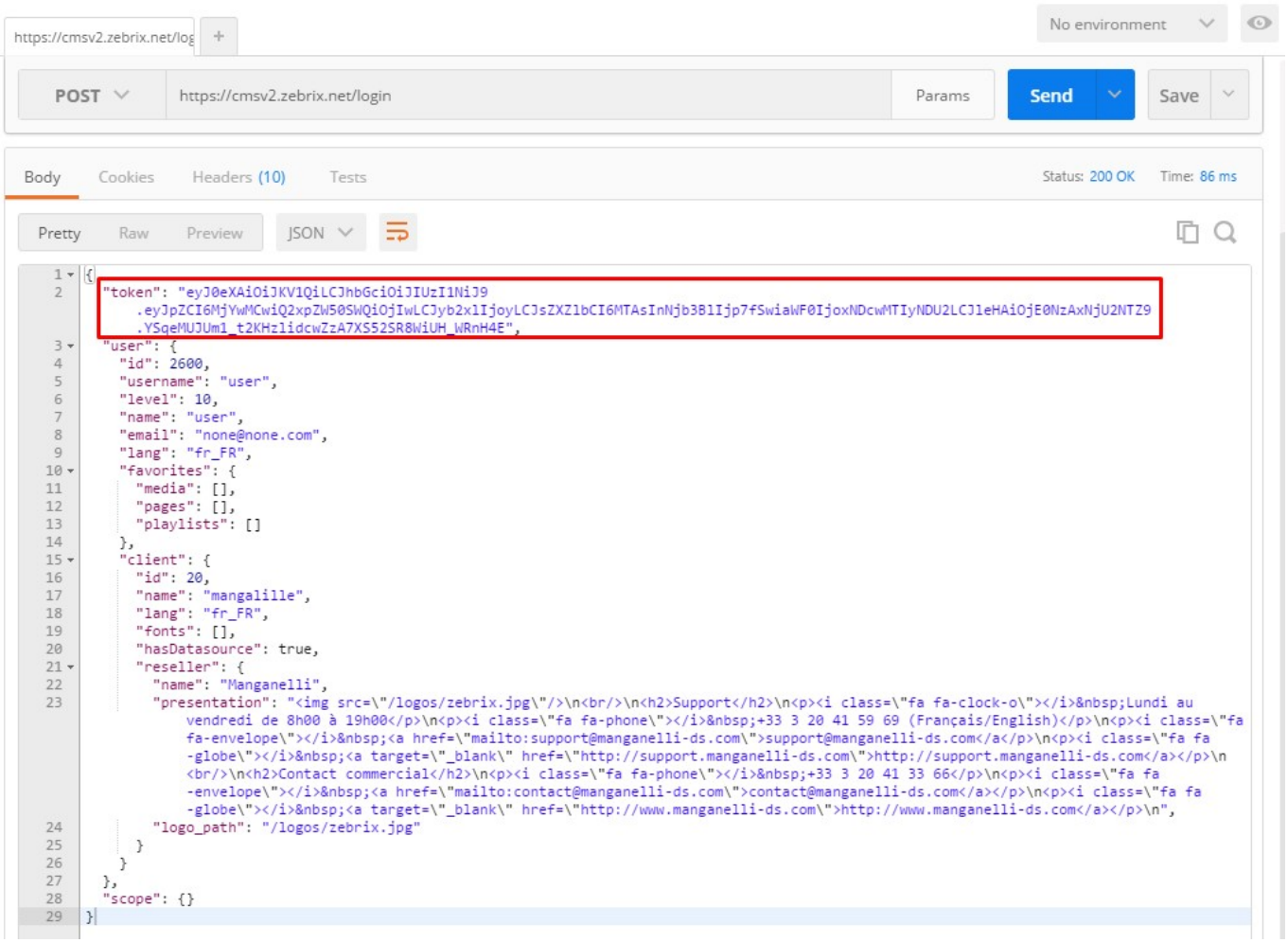
Parameters	
clientname	the customer name
username	the username
password	the user password

login demo with "POSTMAN" (Google Chrome Extension)

Call



Return



Example of GET API on screen

GET /api/screen/:id

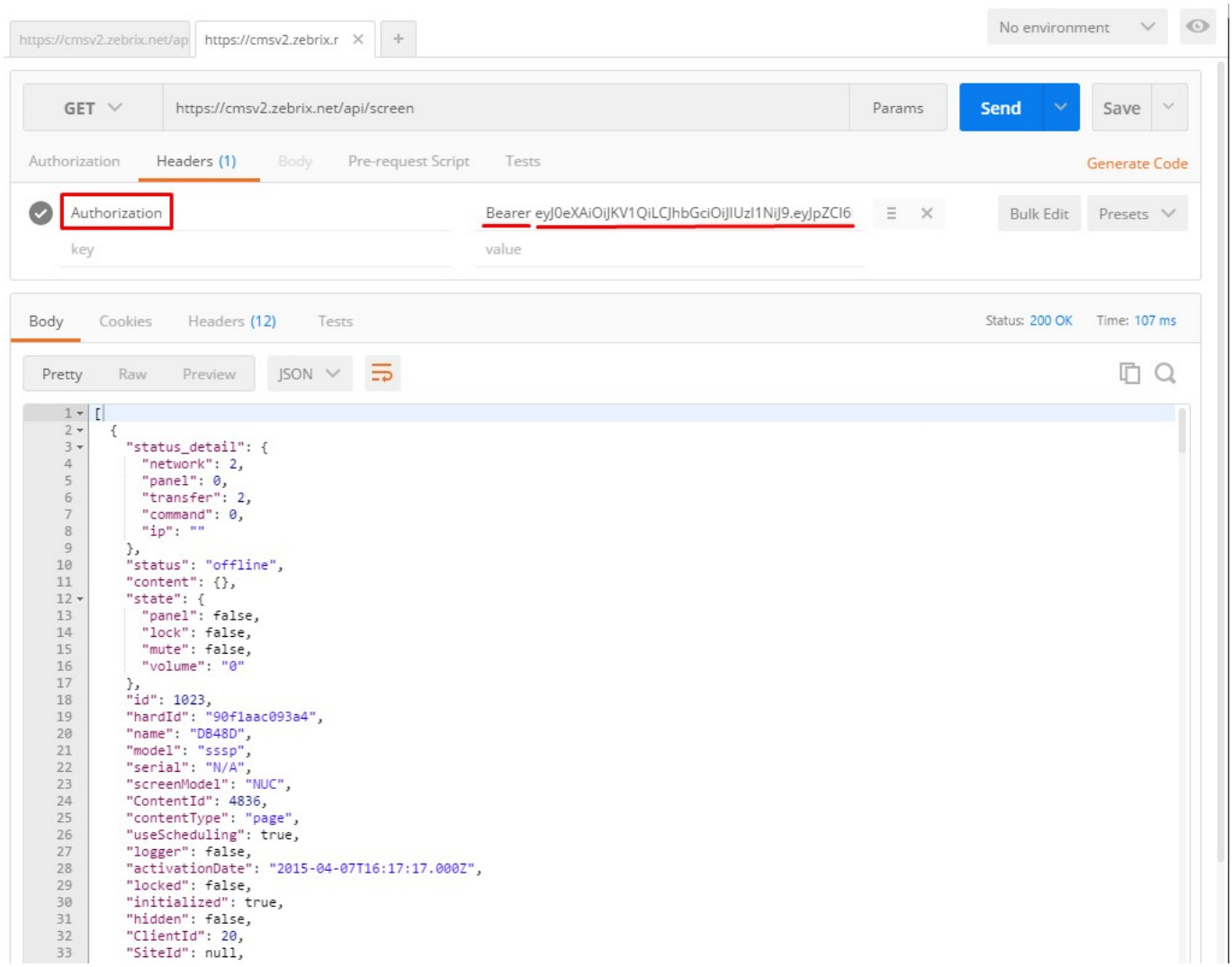
This example will demonstrate how to request information with a GET call to the API. We will use the screen api (to get the list of all screens with their properties). It will work the same for all other objects in zebrix (media, pages, playlist, etc.)

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.

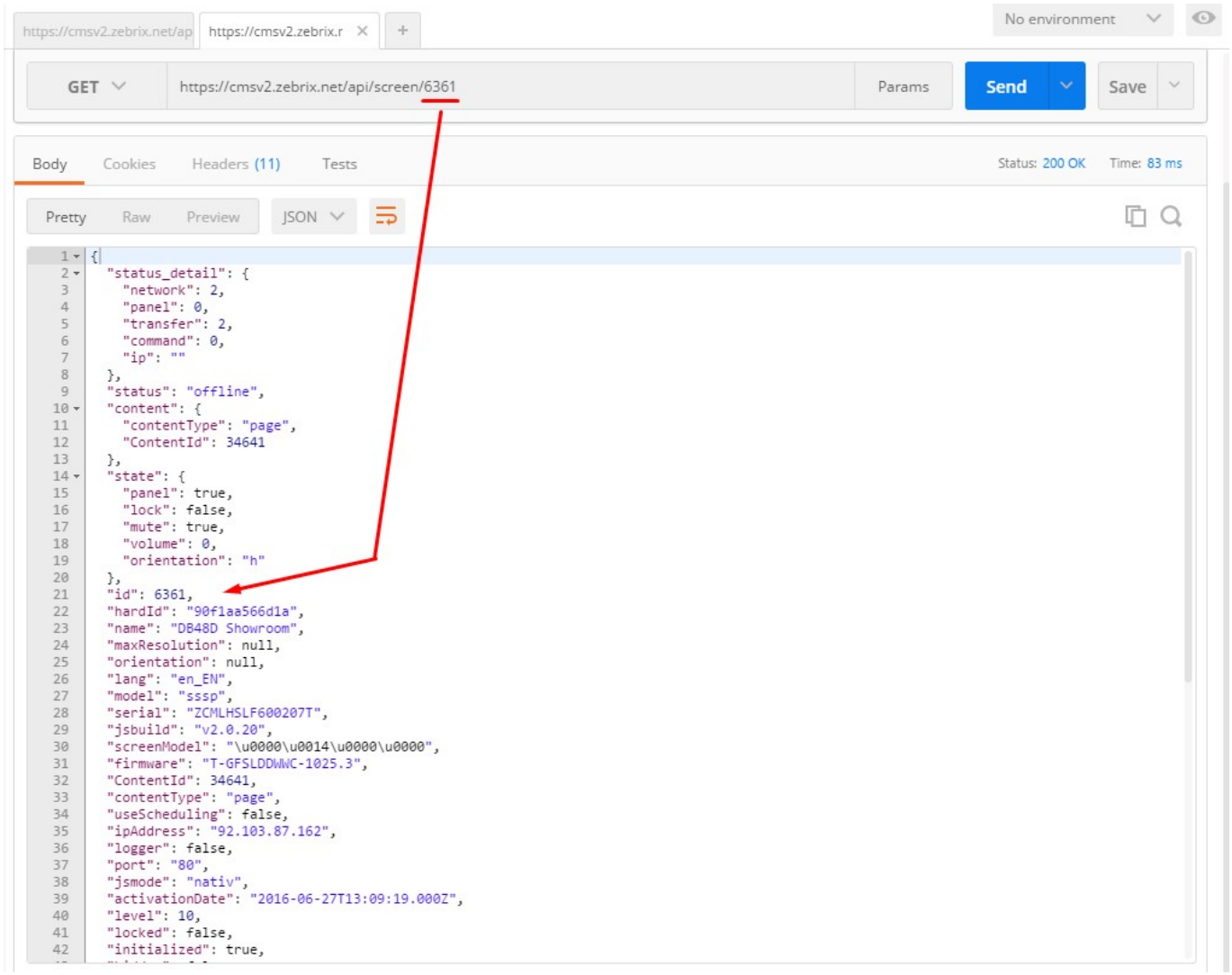
The token must be prefixed with the word “Bearer” and a space char that way :

Bearer eyJ0eXAiOiJKV1QiLCJhbGci0kLIUzI1NiJ9.eyJpZCI6MjYwMCwiQ2xpZW50SWQiOjIwLCJyb2xlIjoyLCJsZXZlbCI6MTAsInNjb3BlIjp7fSwiaWF0IjoxNDcwMTIyNDU2LCJleHAiOjE0NzAxNjU2NTZ9.YSqeMUJUm1*t2JullidcwZzA7XS52SR8WiUH*WRnH4E

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



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 :



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	HASABEAMER
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)

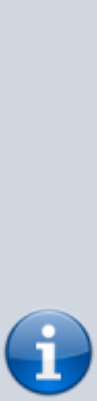
Using our meeting rooms example, here is the result you will get :


```
{
  "defaults": {
    "ROOM": "Black Room",
    "VISITOR": "Audi",
    "SEATS": "10",
    "HASABEAMER": "true"
  },
  "id": 1010,
  "name": "test14",
  "createdAt": "2016-04-29T14:53:26.000Z",
  "updatedAt": "2016-04-29T14:57:11.000Z",
  "ClientId": 20,
  "TaxoId": null,
  "taxo": null
},
{
  "defaults": {
    "ROOM": "Yellow room",
    "VISITOR": "MORA",
    "SEATS": "6",
    "HASABEAMER": "true"
  },
  "id": 1011,
  "name": "test21",
  "createdAt": "2016-04-29T14:59:58.000Z",
  "updatedAt": "2016-04-29T14:59:58.000Z",
  "ClientId": 20,
  "TaxoId": null,
  "taxo": null
},
```


Theses datasource 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 uderscope + page + id of the concerned page.



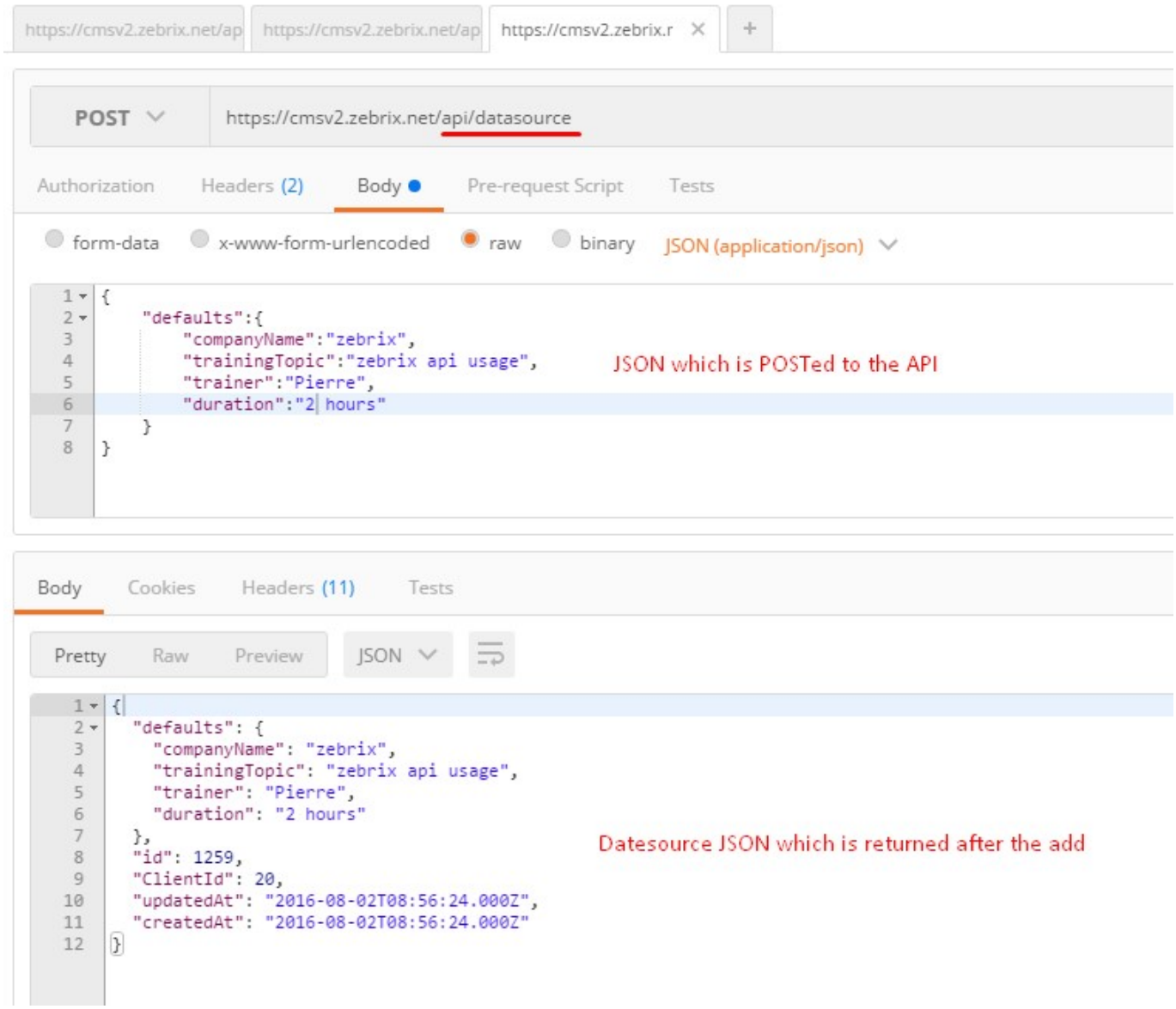
How to add a new datasource to zebrix

POST /api/datasource/

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

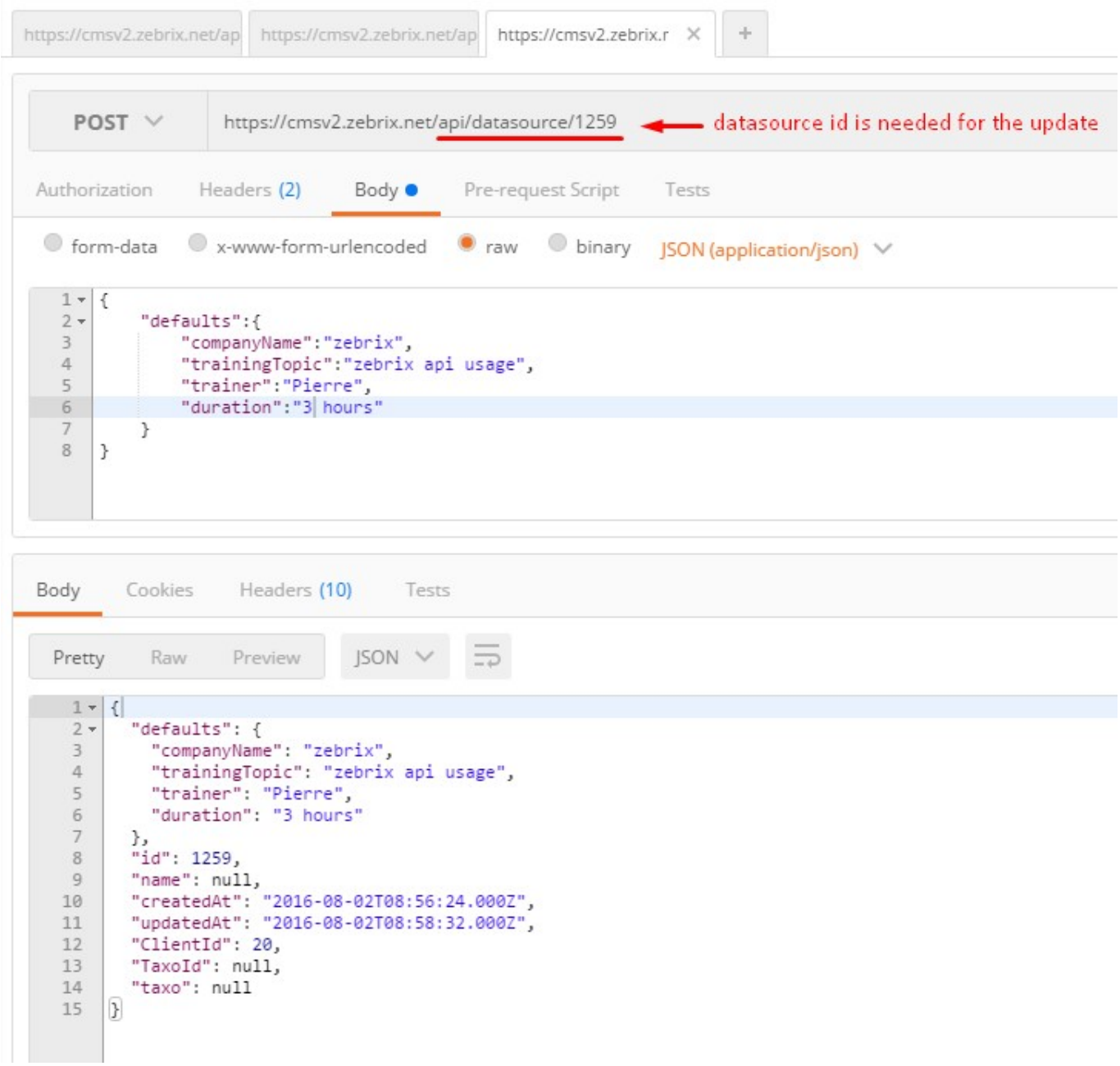
```
{
    "defaults":{
        "companyName":"zebrix",
        "trainingTopic":"zebrix api usage",
        "trainer":"Pierre",
        "duration":"3 hours",
        "misc":"test"
    }
}
```

Example of usage :



How to update an existing datasource

POST /api/datasource/:id



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. Is 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.