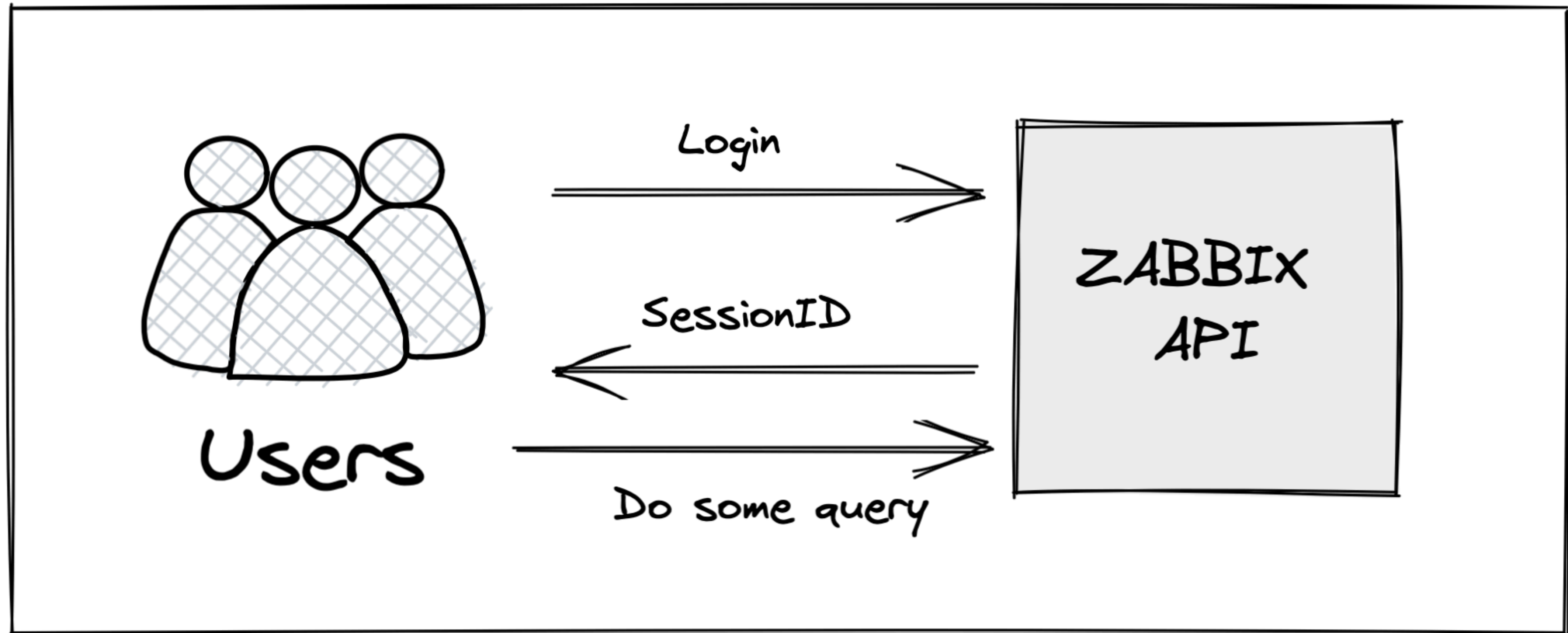




# ZABBIX API Workflow



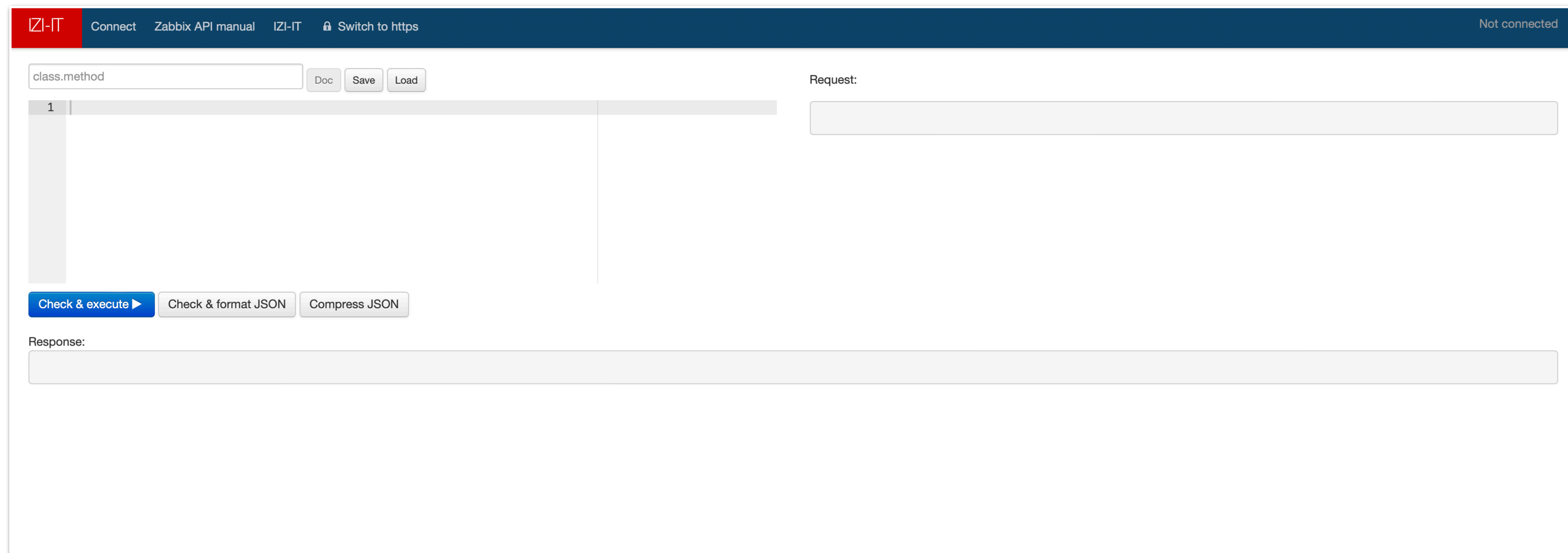
# API test

- <https://workshop.izi-it.io>
  - Connect with **user1** to **user39** with password **IZI-IT2022!**
- <http://lab.izi-it.io>



# API test

- <http://lab.izi-it.io>

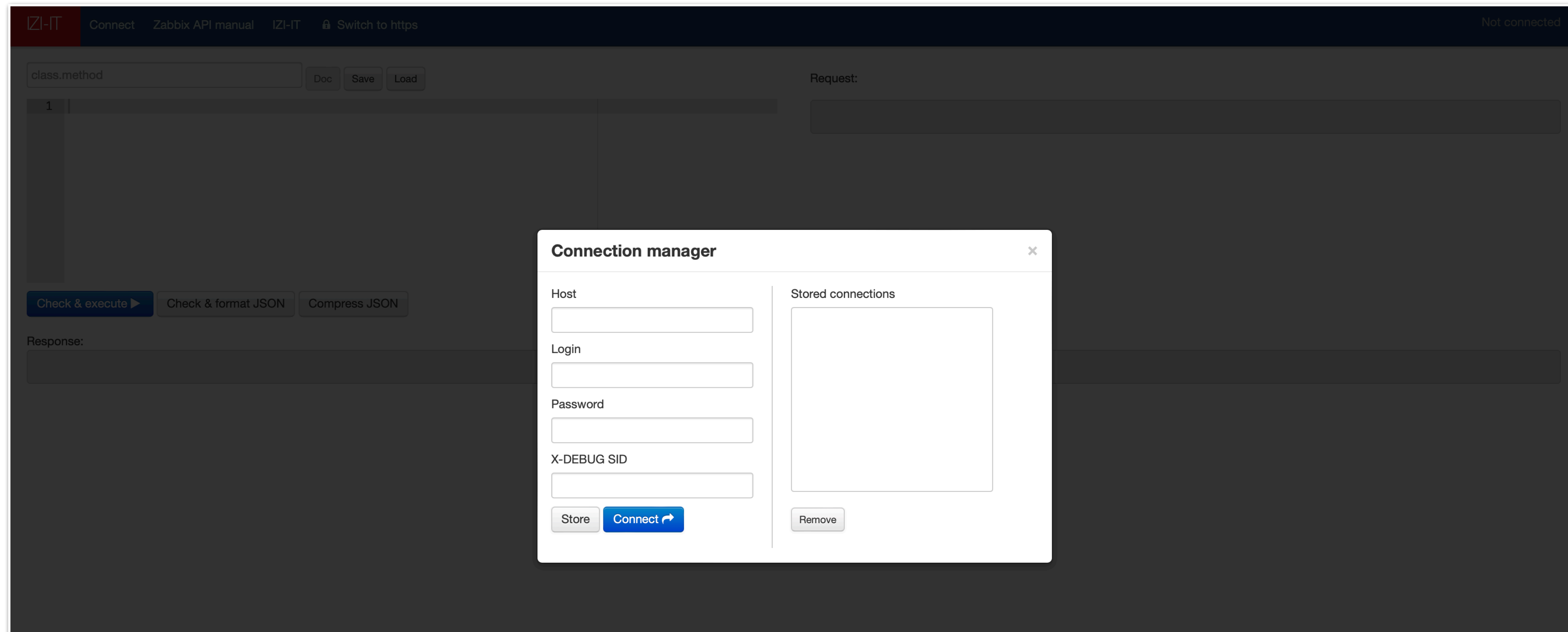


The screenshot shows the IZI-IT web interface for API testing. At the top, a dark blue header contains the IZI-IT logo, navigation links (Connect, Zabbix API manual, IZI-IT), a lock icon with 'Switch to https', and a 'Not connected' status indicator. The main workspace is divided into several sections. On the left, there is a text input field containing 'class.method', followed by 'Doc', 'Save', and 'Load' buttons. Below this is a code editor with a line number '1' and a vertical cursor. To the right of the editor is a 'Request:' label and a large text area for entering the request body. At the bottom left, there are three buttons: 'Check & execute' (highlighted in blue), 'Check & format JSON', and 'Compress JSON'. Below these buttons is a 'Response:' label and a large text area for displaying the response.



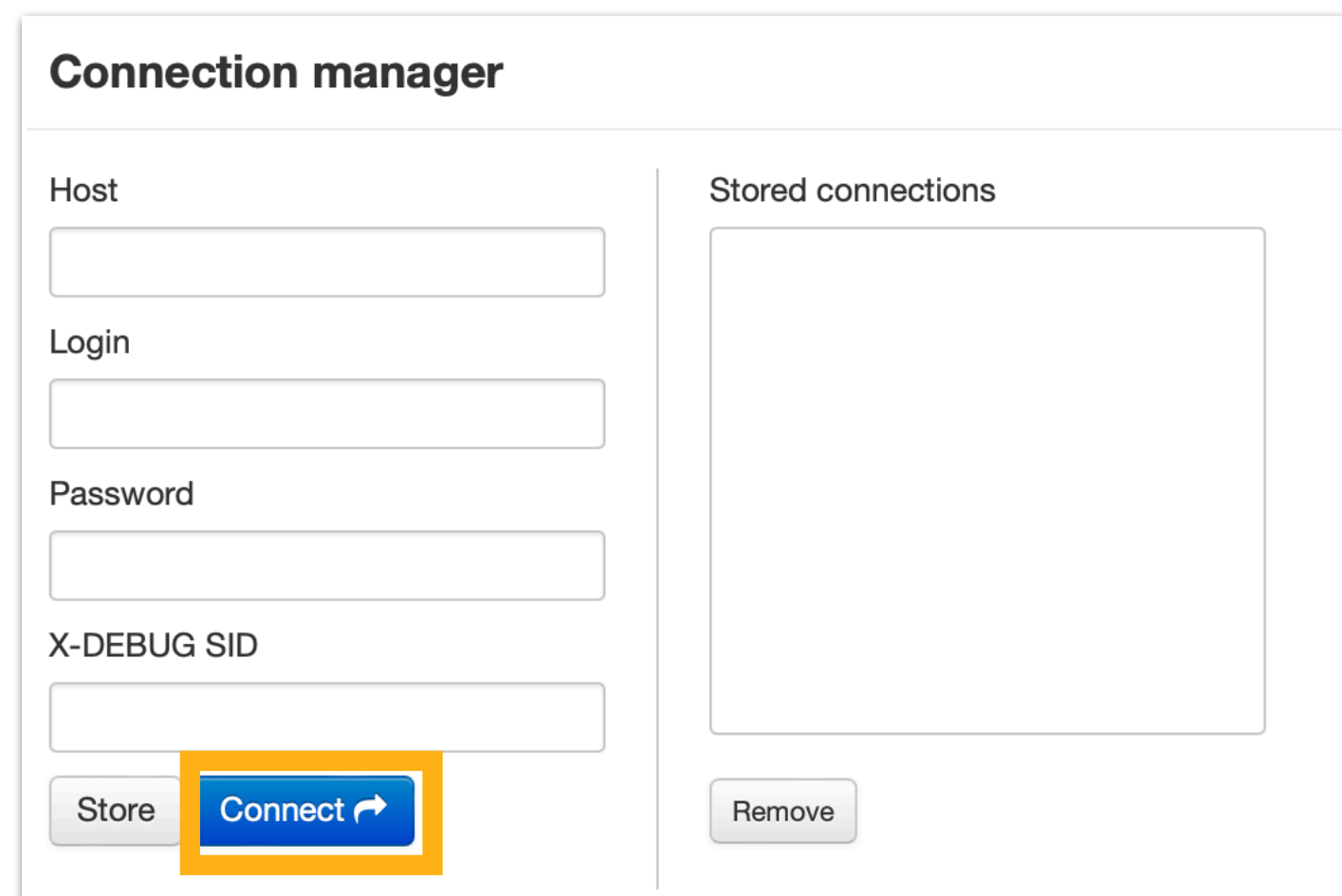
# API test

- <http://lab.izi-it.io>



# API test

- <http://lab.izi-it.io>



The screenshot shows a web interface titled "Connection manager". On the left, there are four input fields labeled "Host", "Login", "Password", and "X-DEBUG SID". Below these fields are two buttons: "Store" and "Connect". The "Connect" button is highlighted with a yellow border and contains a right-pointing arrow icon. On the right side of the interface, there is a section titled "Stored connections" which contains a large, empty rectangular box. Below this box is a "Remove" button.

- Host: <https://workshop.izi-it.io>
- Login: user1 to user39
- Password: IZI-IT2022!

And then **Connect**



# API test

- <http://lab.izi-it.io>

The screenshot displays the IZI-IT web interface for API testing. The top navigation bar includes the IZI-IT logo, a 'Connect' button, links to 'Zabbix API manual' and 'IZI-IT', a 'Switch to https' button with a lock icon, and a status indicator 'Connected to workshop.izi-it.io'.

The main workspace is divided into two sections. On the left, there is a text input field labeled 'class.method' containing the value '1'. Above this field are three buttons: 'Doc', 'Save', and 'Load'. Below the input field is a large, empty text area for editing the request body.

On the right, the 'Request:' section shows a JSON-RPC request body: `{"jsonrpc": "2.0", "method": "apiinfo.version", "id": 1, "params": []}`.

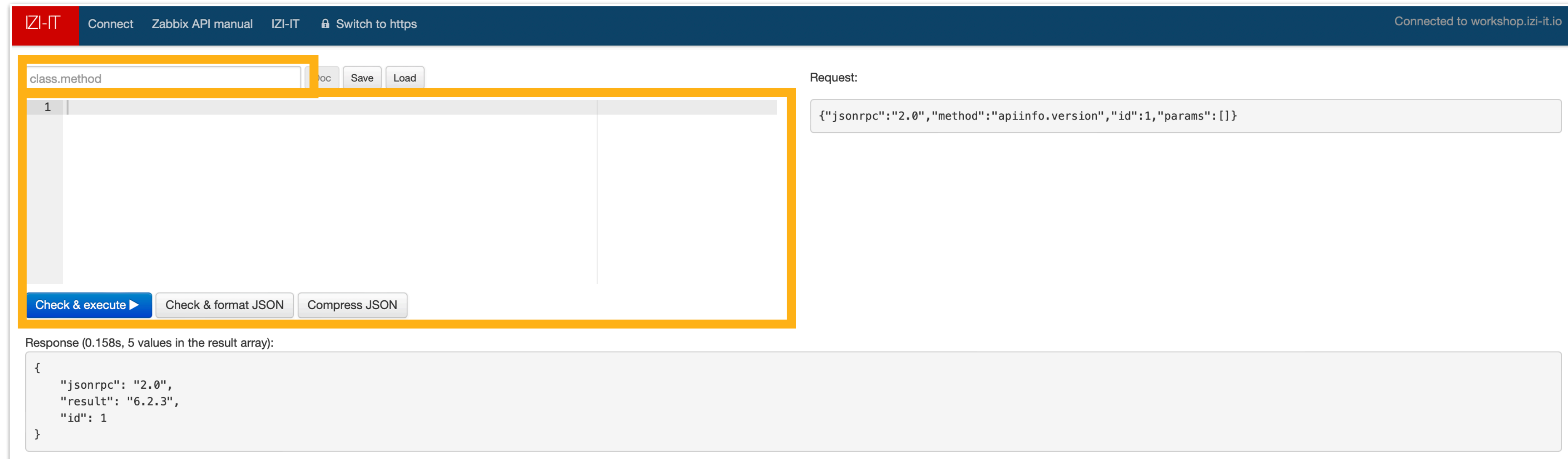
Below the request input, there are three buttons: 'Check & execute' (highlighted in blue), 'Check & format JSON', and 'Compress JSON'.

The 'Response' section at the bottom shows the result of the execution: 'Response (0.158s, 5 values in the result array):'. The response body is a JSON object: `{ "jsonrpc": "2.0", "result": "6.2.3", "id": 1 }`.



# API test

- What is the class.method field ?
- What is the params field ?



The screenshot displays the IZI-IT web interface for API testing. The top navigation bar includes links for 'Connect', 'Zabbix API manual', 'IZI-IT', and a 'Switch to https' button. The main workspace is divided into several sections:

- Class/Method Input:** A text field labeled 'class.method' with a value of 'apiinfo.version' is highlighted by an orange box. Below it, a list of methods is shown, with '1' selected.
- Request:** A text area contains the JSON-RPC request: 

```
{"jsonrpc": "2.0", "method": "apiinfo.version", "id": 1, "params": []}
```
- Response:** A text area shows the response: 

```
{  "jsonrpc": "2.0",  "result": "6.2.3",  "id": 1}
```

Buttons for 'Check & execute', 'Check & format JSON', and 'Compress JSON' are located below the request and response areas. The status bar at the bottom indicates 'Connected to workshop.izi-it.io'.





# API test

class.method

Save Load

1

Check & execute ▶ Check & format JSON Compress JSON

Request:

```
{ "jsonrpc": "2.0", "method": "apiinfo.version", "id": 1, "params": [] }
```

Response (0.158s, 5 values in the result array):

```
{  "jsonrpc": "2.0",  "result": "6.2.3",  "id": 1 }
```

- class.method: the method you will use
- params field: the params of your method



# API test

- <https://www.zabbix.com/documentation/current/en/manual/api/reference>



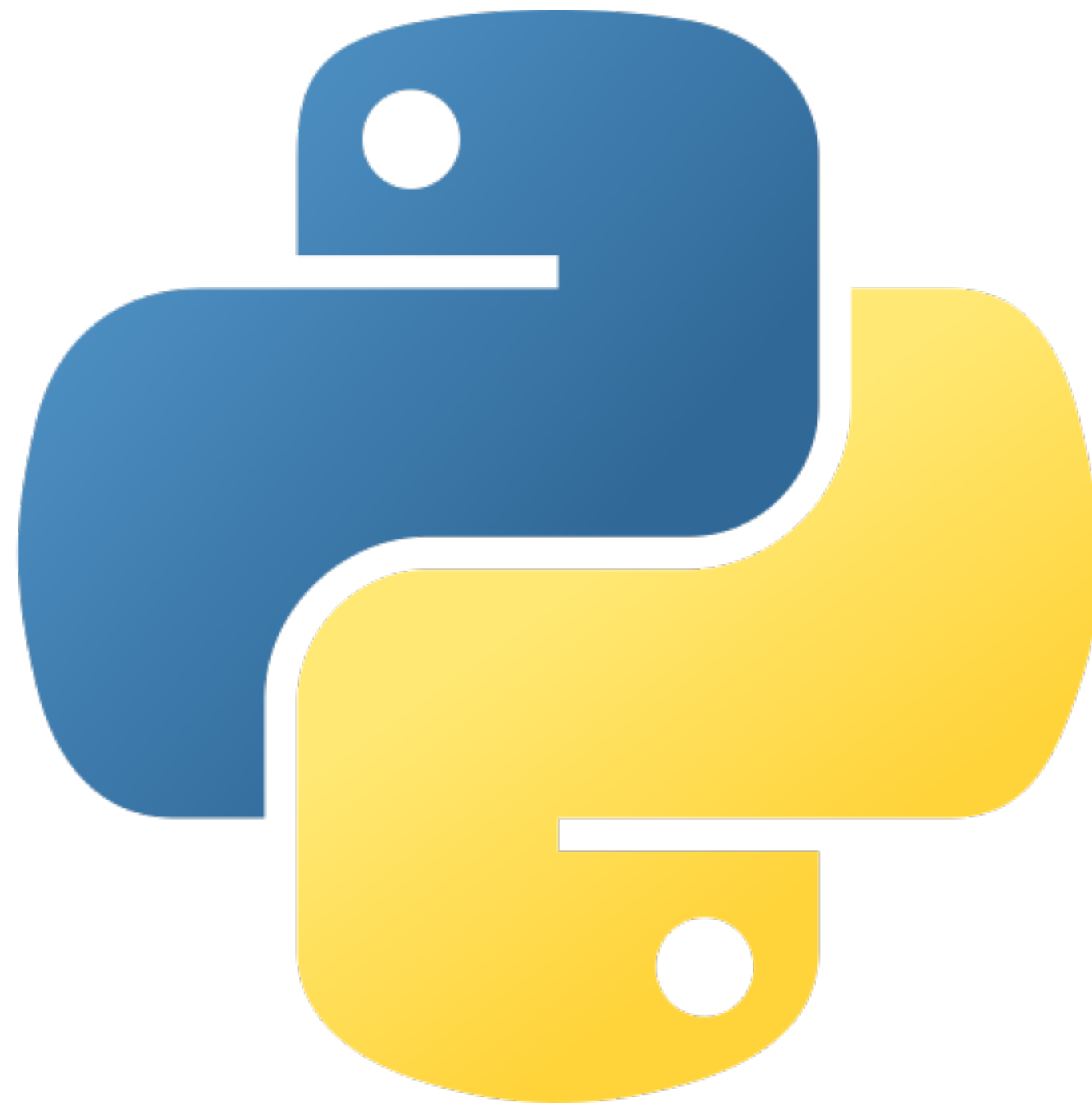
# API test

- Try to create a single host with the documentation
  - What kind of methods will we use ?



# ZABBIX API with Python

- Now let's try with Python



# ZABBIX API with Python

- First: we need pyzabbix to be installed
  - pip install pyzabbix
  - <https://github.com/lukecyca/pyzabbix>



# ZABBIX API with Python

```
from pyzabbix import ZabbixAPI

zapi = ZabbixAPI("http://zabbixserver.example.com")

zapi.login("zabbix user", "zabbix pass")

print("Connected to Zabbix API Version %s" % zapi.api_version())
```



# ZABBIX API with Python

- Now let's do the same and create a host with Python

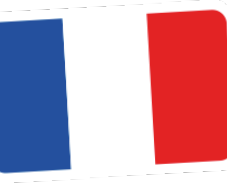


# ZABBIX API with Python

- So that's it
- Maybe you want to discuss about the API or the fact that France will be one more time World Champion of Football at the end of the year







# Enjoy and don't forget

## Ba-na-na

