

Lab: HTTP and REST

Problems for in-class lab for the ["JS Front-End" course @ SoftUni](#).

1. REST Countries

NOTE: Install "[Postman](#)" REST Client to **ease** your tasks.

Your first task is to get detailed information about Bulgaria.

- Send a "**GET**" request to the link given below.
- **Copy** the response in JSON format.

REQUEST:

<https://restcountries.com/v2/name/Bulgaria>

GET

https://restcountries.com/v2/name/Bulgaria

Send

RESPONSE:

GET

https://restcountries.com/v2/name/Bulgaria

Send

ParamsAuthorizationHeaders (7)BodyPre-request ScriptTestsSettings

BodyCookiesHeaders (7)Test Results

Status: 200 OKTime: 416 msSize: 856 BSave Response

PrettyRawPreviewVisualizeJSON

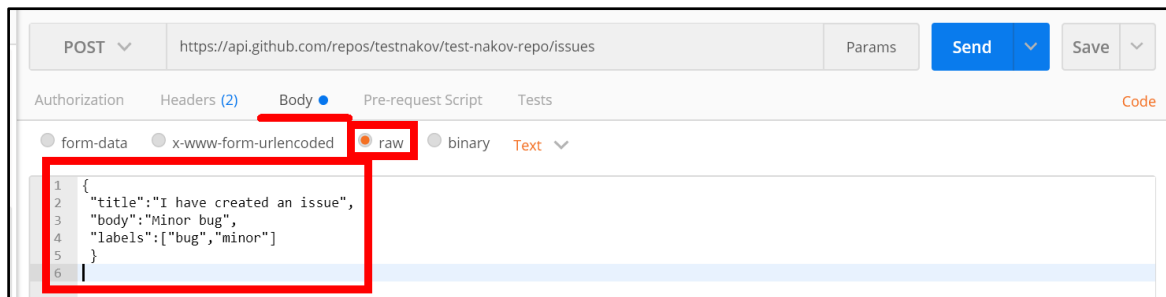
```
1 [
2   {
3     "name": "Bulgaria",
4     "topLevelDomain": [
5       ".bg"
6     ],
7     "alpha2Code": "BG",
8     "alpha3Code": "BGR",
9     "callingCodes": [
10      "359"
11    ],
12    "capital": "Sofia",
13    "altSpellings": [
14      "BG",
15      "Republic of Bulgaria",
16      "Република България"
17    ],
18    "subregion": "Eastern Europe",
19    "region": "Europe",
20    "population": 6927288,
21    "latlng": [
22      43.0,
23      25.0
```

2. GitHub: Labels Issue

Get the **first** issue from repository with name "test-nakov-repo". Send a GET request to <https://api.github.com/repos/testnakov/test-nakov-repo/issues/:id>, where **:id** is the issue.

3. Github: Create Issue

This time we have to **create** an issue (data should be **send** to the server). Send a **"POST"** request to the server with the following JSON as **body** (send it as **application/json**):



```
1 {
2   "title": "I have created an issue",
3   "body": "Minor bug",
4   "labels": ["bug", "minor"]
5 }
6
```

You need to use your GitHub **account credentials** to submit issues. Under the Authorization tab, select Basic and enter your username and password. Send the request to the URI from the previous task, but without the **:id**.

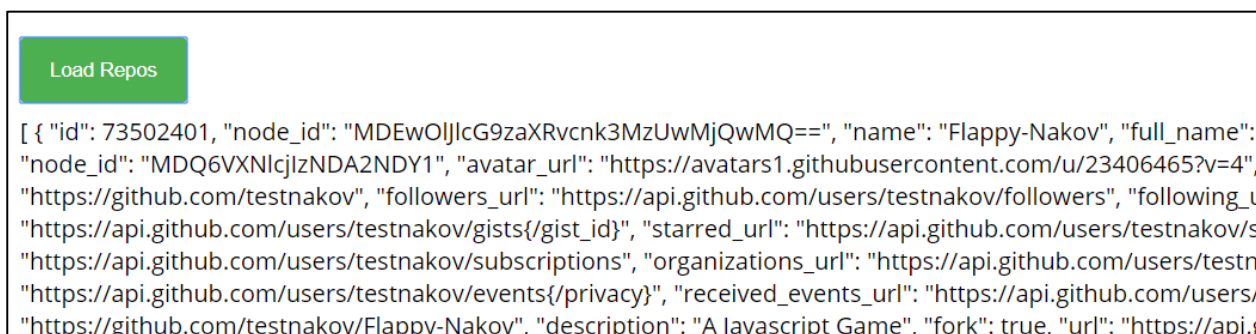
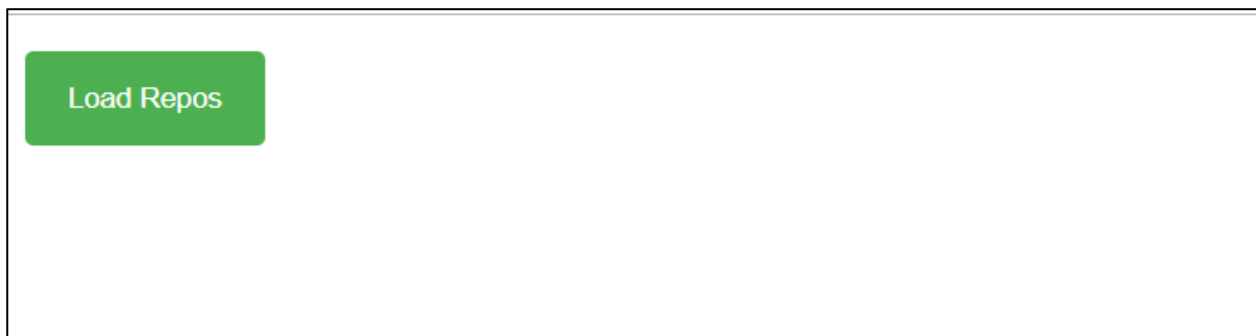
4. GitHub Repos

Your task is to **write** a JS function that **loads** a github repository **asynchronously with AJAX**. You should **use the Fetch API**. Obtain the data by making a **GET** request to the following URL:

"https://api.github.com/users/testnakov/repos"

Transform the **body** to **text** with **res.text()** and in the second **then()** block of the **Promise** replace the text content of a **div** element with **id "res"** with the value from the response. **Do not format** the response in any way.

Examples



```
[{"id": 73502401, "node_id": "MDEwOlJlcG9zaXRvcnk3MzUwMjQwMQ==", "name": "Flappy-Nakov", "full_name": "MDQ6VXNlcjZlNDA2NDY1", "avatar_url": "https://avatars1.githubusercontent.com/u/23406465?v=4", "https://github.com/testnakov", "followers_url": "https://api.github.com/users/testnakov/followers", "following_u": "https://api.github.com/users/testnakov/gists{/gist_id}", "starred_url": "https://api.github.com/users/testnakov/s", "https://api.github.com/users/testnakov/subscriptions", "organizations_url": "https://api.github.com/users/testn", "https://api.github.com/users/testnakov/events{/privacy}", "received_events_url": "https://api.github.com/users/", "https://github.com/testnakov/Flappy-Nakov", "description": "A Javascript Game", "fork": true, "url": "https://api.g
```

5. Github Repos By Username

Your task is to **write** a JS function that **executes** an **AJAX** request with **Fetch API** and loads all user **github repositories** by a given username (taken from an input field with id "username") into a **list** (each repository as a **list-item**) with id "repos". Use the properties **full_name** and **html_url** of the returned objects to create a link to each repo's GitHub page. If an **error** occurs (like 404 "Not Found"), **append** to the list a list-item with **text** the current instead. Clear the contents of the list before any new content is appended. See the **highlighted lines** of the skeleton for formatting details of each list item.

Examples

GitHub username: Load Repos

- [{repo.full_name}](#)

GitHub username: Load Repos

- [k1r1L/Angular-2-Demos](#)
- [k1r1L/Angular-Sli.do](#)
- [k1r1L/awesome-interview-questions](#)
- [k1r1L/CSharp-Web-Development-Basics](#)
- [k1r1L/CSharp-Web-MVC-Frameworks-ASP.NET](#)
- [k1r1L/Databases-Advanced-Entity-Framework](#)
- [k1r1L/Databases-MS-SQL-Server-Exercises](#)
- [k1r1L/Express-Demo-Server](#)
- [k1r1L/express-js-exercises](#)
- [k1r1L/Front-End-Web-FMI-Project](#)
- [k1r1L/Fundamental-Level](#)
- [k1r1L/JavaScript-Advanced](#)
- [k1r1L/JavaScript-Applications](#)
- [k1r1L/JavaScript-Fundamentals](#)
- [k1r1L/React-Project](#)
- [k1r1L/SoftUni-ExpressJS-Fundamentals](#)
- [k1r1L/Softuni-Memes](#)
- [k1r1L/Tetris-JavaFundamentals-Teamwork](#)
- [k1r1L/TicTacToe](#)
- [k1r1L/University-Information-System](#)

6. Github Commits

Write a JS program that loads all commit messages and their authors from a github repository using a given HTML.

The `loadCommits()` function should get the **username** and **repository** from the HTML textboxes with IDs **"username"** and **"repo"** and make a **GET** request to the **Github API**:

`https://api.github.com/repos/<username>/<repository>/commits`

Swap **<username>** and **<repository>** with the ones from the HTML:

- In case of **success**, for **each** entry add a **list item** (``) in the **unordered list** (``) with **id** **"commits"** with text in the following format:
"<commit.author.name>: <commit.message>"
- In case of an **error**, add a single **list item** (``) with text in the following format:
"Error: <error.status> (Not Found)"

Screenshots:

GitHub username:

Repo:

- Svetlin Nakov: Delete Console.Cin.v11.suo
- Svetlin Nakov: Create LICENSE
- Svetlin Nakov: Update README.md
- Svetlin Nakov: Added better documentation

GitHub username:

Repo:

- Error: 404 (Not Found)

