# Lab: HTTP and REST

Problems for in-class lab for the ["JS Front-End" course @ SoftUni](https://softuni.bg/trainings/3976/js-front-end-february-2023).

**1. REST Countries**

**NOTE: Install** "[Postman](https://www.getpostman.com/)" REST Client to **ease** your tasks.

Your first task is to get detailed information about Bulgaria.

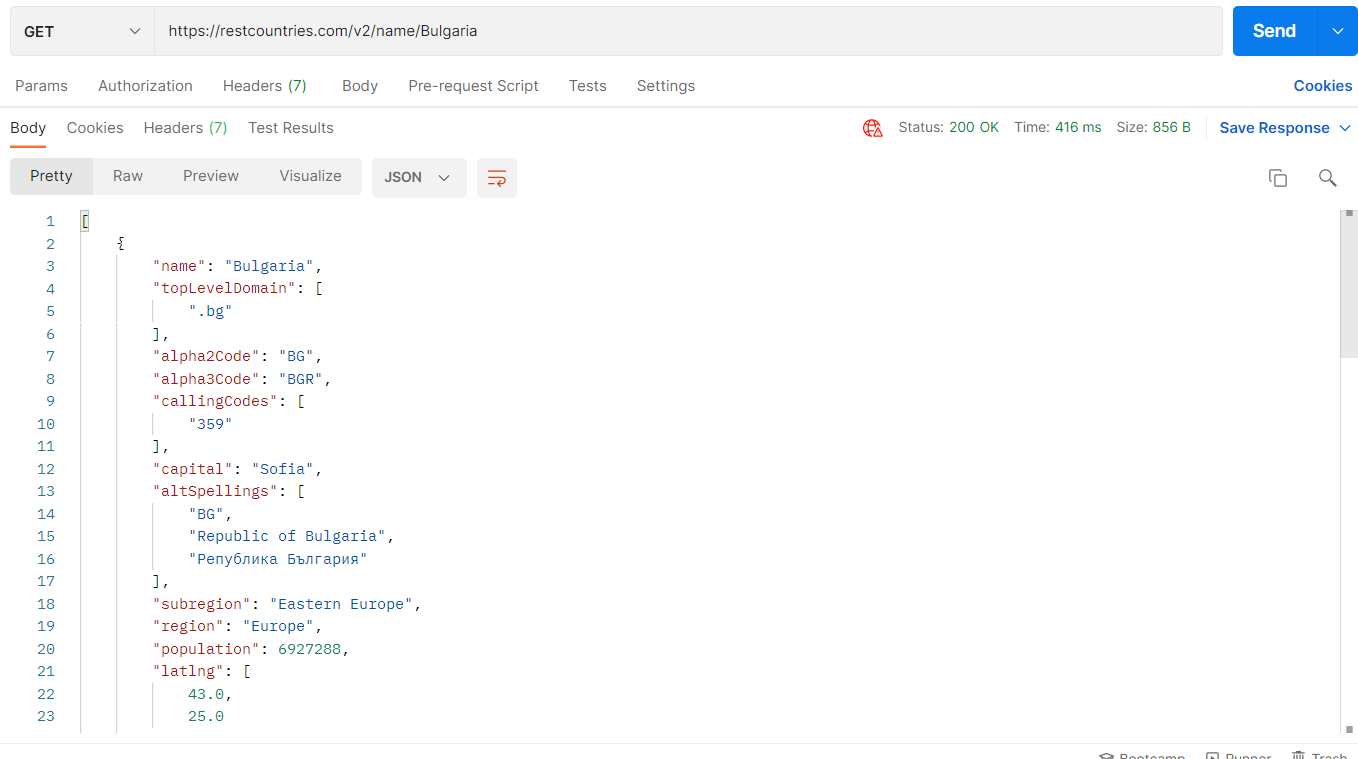
* Send a "**GET**" request to the link given below.
* **Copy** the response in JSON format.

**REQUEST**:

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



**RESPONSE**:



**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):



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

Shape

Description automatically generated

Text

Description automatically generated

## 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

Graphical user interface, text, application

Description automatically generated

A picture containing text

Description automatically generated

## 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 (<li>) in the unordered list (<ul>) with id "commits" with text in the following format:

"<commit.author.name>: <commit.message>"

* In case of an **error**, add a single list item (<li>) with text in the following format:  
  "Error: <error.status> (Not Found)"

### Screenshots:

Graphical user interface, text, application, chat or text message

Description automatically generated

Graphical user interface, text, application

Description automatically generated