

Lab: REST Services and AJAX

1. Darth Vader Response

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

Your first task is to get detailed information about Darth Vader.

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

REQUEST:

`https://swapi.co/api/people/4`



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**.

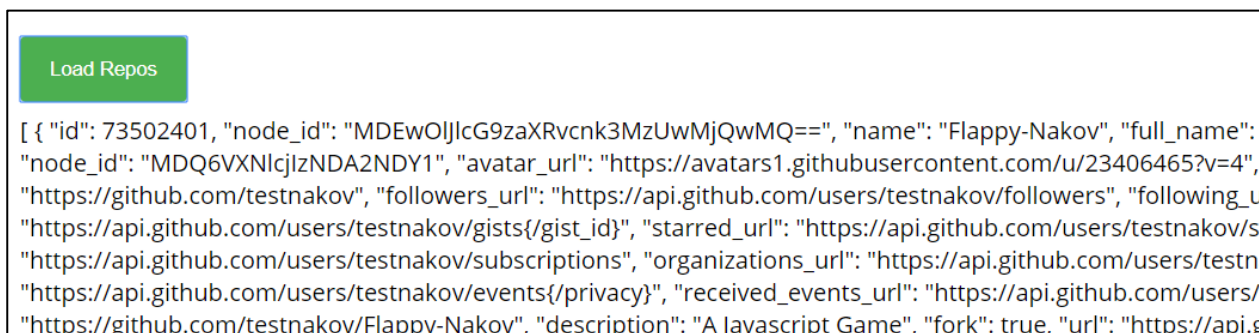
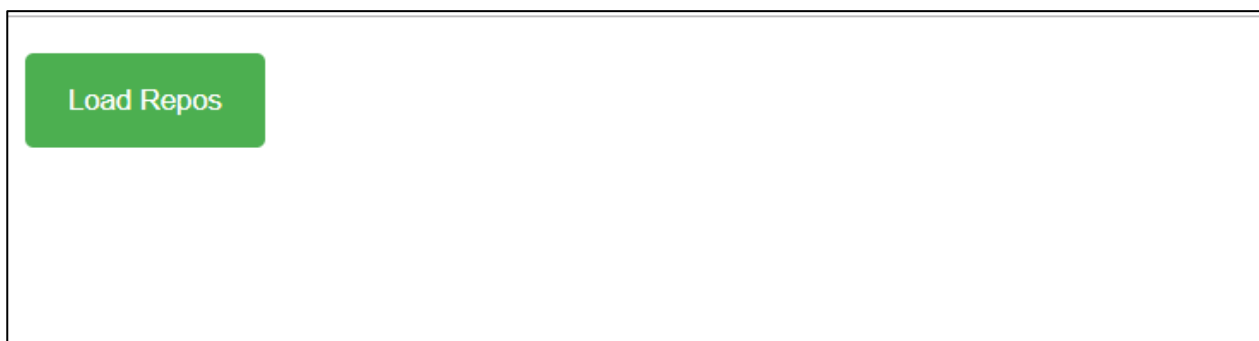
4. XHR (XmlHttpRequest)

Your task is to **write** a JS function that **loads** a github repository **asynchronously with AJAX**. You should **create** an instance of **XmlHttpRequest** and attach an **onreadystatechange** event to it. (An EventHandler that is called whenever the readyState attribute changes). In your event handler, when the **readyState** attribute reaches a value of **4** (it is ready), replace the text content of a **div** element with **id "res"** with the value of the **responseText** property of the request. **Do not format** the response in any way.

[More on XmlHttpRequest.open\(\)](#)

Skeleton will be provided in the **Resources** folder.

Examples



5. Github Repos

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.

Skeleton will be provided in the **Resources** folder.

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](#)