# Exercise: REST Services and AJAX

Problems for exercises and homework for the ["JavaScript Applications" course @ SoftUni](https://softuni.bg/trainings/3847/js-applications-october-2022).

**Working with Remote Data**

For the solution of some of the following tasks, you will need to use an up-to-date version of the **local REST service**, provided in the lesson’s resources archive. You can [read the documentation here](https://github.com/softuni-practice-server/softuni-practice-server).

1. **REST Countries**

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

* Manually compile an HTTP (as text) request for retrieving information about Bulgaria;
* Use Postman to make the same request;
* Make a request that retrieving only the fields name, capital, region, population for the country Italy.
* Make a request that takes all German-speaking countries.

Your first task is to get detailed information about Bulgaria.

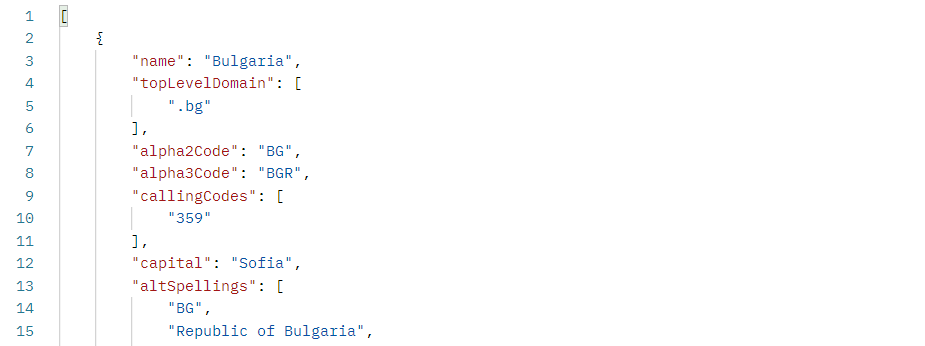
* Send a "**GET**" request to the link given below.

**REQUEST**:

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



**RESPONSE**:



Each API has documentation, where you can see how to use the API. You can find the documentation of this API here: <https://restcountries.com/>

* Now try to filter only specific fields of the information about Italy. Send a GET request with the needed parameter to get a response only with this information about the country:
* name, capital, region and population;



* There is a way to get a response holding all the countries, which citizens speak the German language. Send a GET request to become the information for these countries (Austria, Belgium, Germany, Holy See, Liechtenstein, Luxembourg, Switzerland), but filter the response to have only their names and region.



**Solution:**

**🡪** filter only specific field of the information about Italy - - name, capital, region and population;

^https://restcountries.com/v2/name/Bulgaria?fields=name,population,capital,region

**🡪** all the countries, which citizens speak the German language

^https://restcountries.com/v3.1/lang/german?fields=name,region

## Bus Stop

Perform an HTTP request that displays arrival times for all buses by a given bus stop ID.

* GET: <http://localhost:3030/jsonstore/bus/businfo/:busId>

You will receive a JSON object in the format:

stopID: {

name: stopName,

buses: { busId: time, … }

}





### Hints

The webhost will respond with valid data to IDs 1287, 1308, 1327 and 2334.



Solution

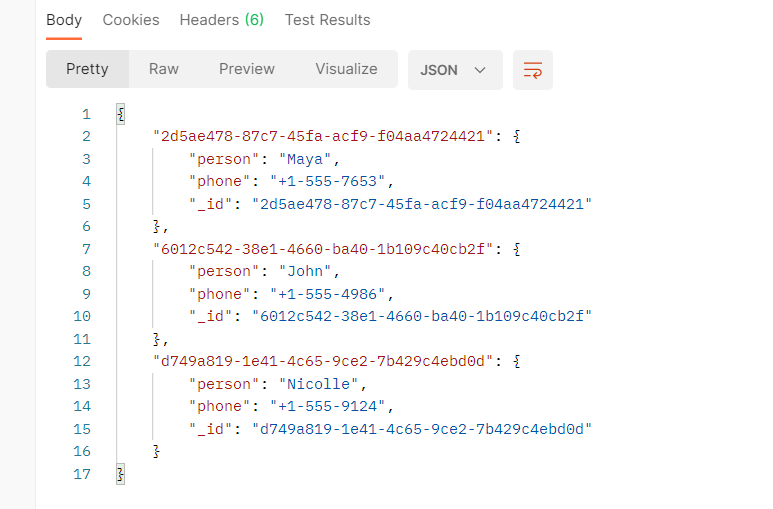
--- <http://localhost:3030/jsonstore/bus/businfo/1287>

--- <http://localhost:3030/jsonstore/bus/businfo/1308>

## Phonebook GET

Perform an HTTP request that show people’s names, their phones and \_id.

GET requests: <http://localhost:3030/jsonstore/phonebook>



## Phonebook POST

Create two different requests with **POST** request in Postman. The data sent in a POST request should be a valid JSON object, containing properties person and phone. Example format:   
{

"person": "<person>",

"phone": "<phone>"  
}

**POST:** [**http://localhost:3030/jsonstore/phonebook**](http://localhost:3030/jsonstore/phonebook)

**Solution:**

**-> create person -> with POST -> body -> row -> JSON -> {"person": "Ivaylo",**

**"phone": "+359123"}**

## Employees

Use the link below to get, post, put and delete request in Postman. Send five types of requests supported by the API.

* <http://dummy.restapiexample.com/>
* http://dummy.restapiexample.com/api/v1/employees
* **GET REQUEST**

Use the link in Full Route section to get all employee data and single employee data.





* **POST REQUEST**

Create new record in database



* **PUT REQUEST**

Update an employee record



* **DELETE REQUEST**

Delete an employee record



Solution:

--- Get lists with employees -> http://dummy.restapiexample.com/api/v1/employees

--- Get only one employee с GET -> http://dummy.restapiexample.com/api/v1/employee/5

---CREATE employee -> https://dummy.restapiexample.com/api/v1/create

--- UPDATE employee с PUT -> http://dummy.restapiexample.com/api/v1/update/5 ->

{"employee\_name": "Joro",

"employee\_salary": 162700,

"employee\_age": 23}

--- DELETE employee -> http://dummy.restapiexample.com/api/v1/delete/5

## Firebase App

Create **Firebase** application to configure a collection. In that case your collection will be with books and authors.

Use postman to create, read, delete entries from the collection.

|  |  |  |
| --- | --- | --- |
| Books Genre | Title | Author |
| Non-Fiction | [Get Out of Your Head: Stopping the Spiral of Toxic Thoughts](https://console.firebase.google.com/project/project-2533123672001324079/database/project-2533123672001324079-default-rtdb/data/Books/Non-Fiction/Get%20Out%20of%20Your%20Head%3A%20Stopping%20the%20Spiral%20of%20Toxic%20Thoughts): | Jennie Allen |
| Trust: America’s Best Chance | Pete Buttigieg |
| When Breath Becomes Air | Paul Kalanithi |
| Romance | It Ends With Us | Colleen Hoover |
| The Proposal | Jasmine Guillory |
| Thrillers | Gone Girl | Gillian Flynn |



* Use **JS-Applications-Remote-Databases-Guide**.

## Backendless App

Create Backendless music application. It has to contain title and singer.

Use Postman to create, read, delete entries from collection.

|  |  |
| --- | --- |
| Singer | Title |
| Eminem | Cinderella Man |
| Alan Walker | Faded |
| Dove Cameron | We Belong |



* Use **JS-Applications-Remote-Databases-Guide**.

## Back4App App

Create Back4App App for students. It has to contain first name, last name and average grade.

Use Postman to create, read, delete entries from collection.

|  |  |  |
| --- | --- | --- |
| First Name | Last Name | Average Grade |
| James | Christopher | 5.70 |
| Barbara | Smith | 4.90 |
| Sarah | Johnson | 5.90 |
| William | Miller | 3.30 |
| Matthew | Taylor | 4.10 |



* In Postman you must add headers.



* Use **JS-Applications-Remote-Databases-Guide**.