

AJAX and jQuery AJAX

AJAX Concepts, XMLHttpRequest,
jQuery AJAX



SoftUni Team
Technical Trainers



SoftUni
Foundation



Software University
<http://softuni.bg>

Table of Contents

1. AJAX Concepts
2. XMLHttpRequest
3. jQuery AJAX
 - `$.load`
 - `$.get() / $.post()`
 - `$.ajax()`
4. AJAX - Examples
 - Accessing Firebase with AJAX



Have a Question?

sli.do

#JS-CORE




AJAX

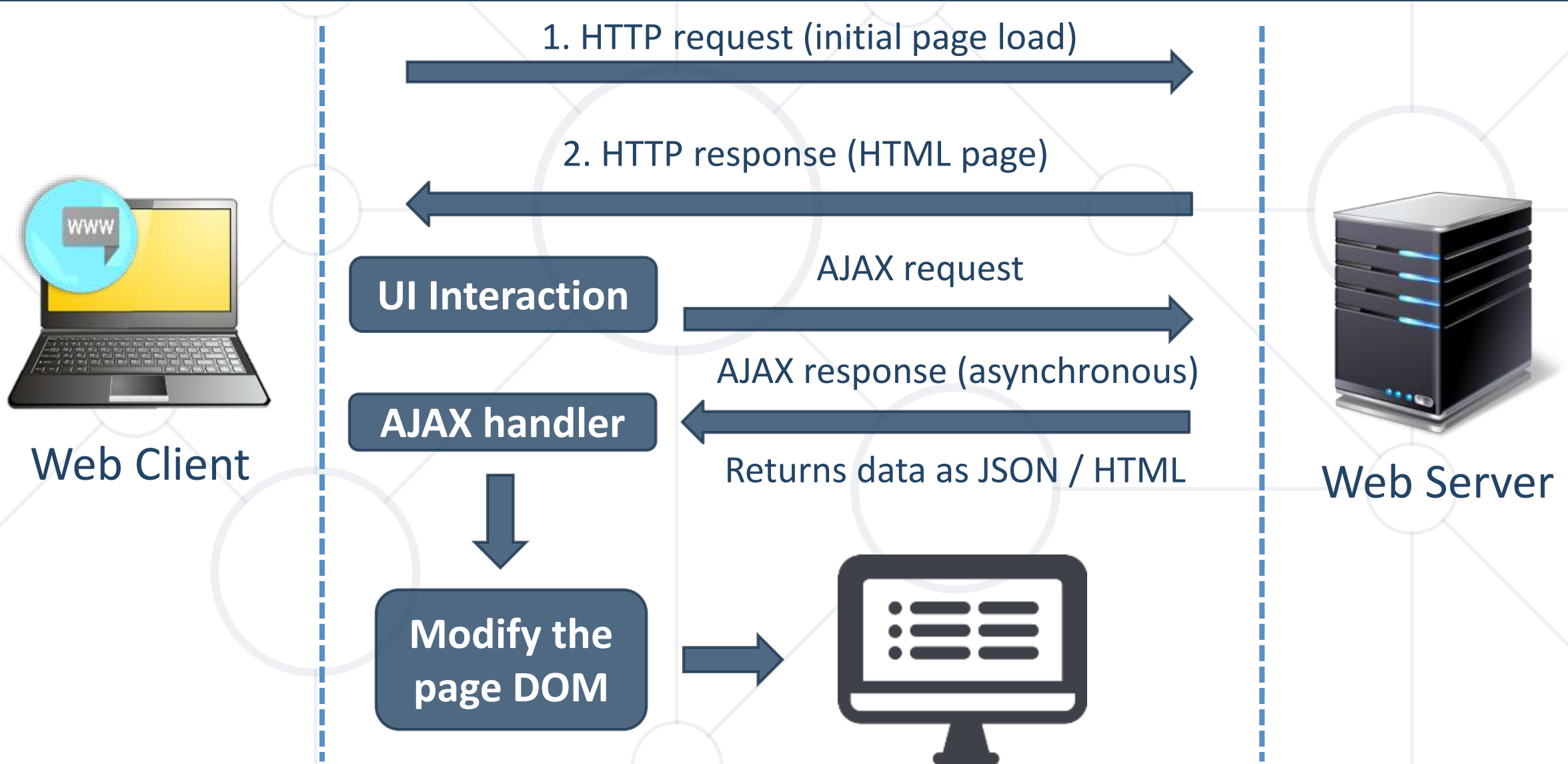
Asynchronous JavaScript and XML

What is AJAX?

Asynchronous JavaScript And XML

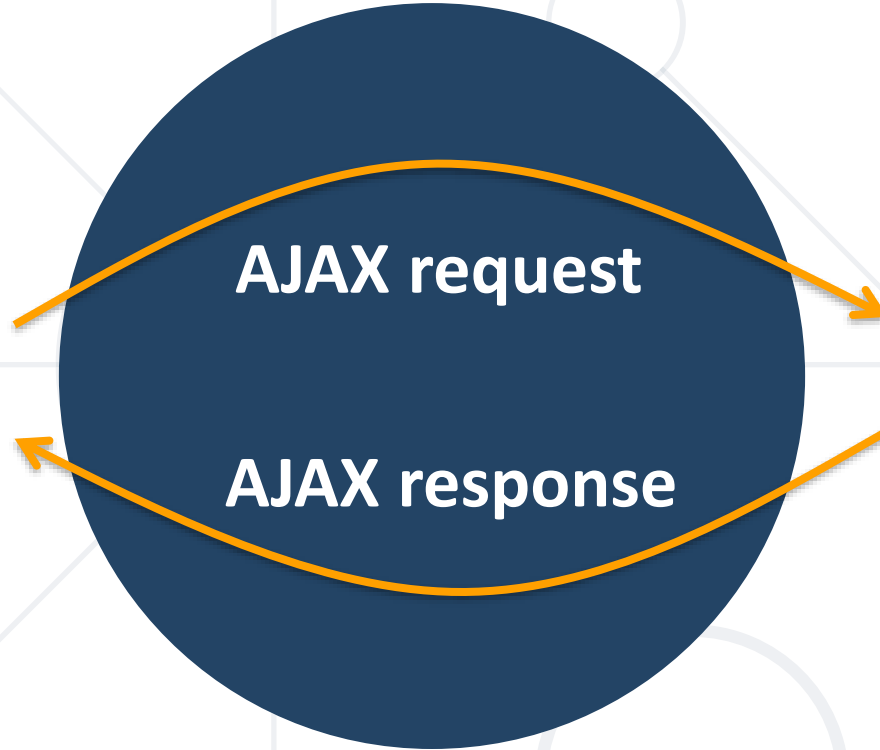
- 
- Background loading of **dynamic content/data**
 - Load data from the Web server and **render** it
 - Two types of AJAX
 - **Partial page rendering**
 - Load HTML fragment + show it in a **<div>**
 - **JSON service**
 - Load JSON object and display it with JS/jQuery

AJAX: Workflow





Web Client



Web Server

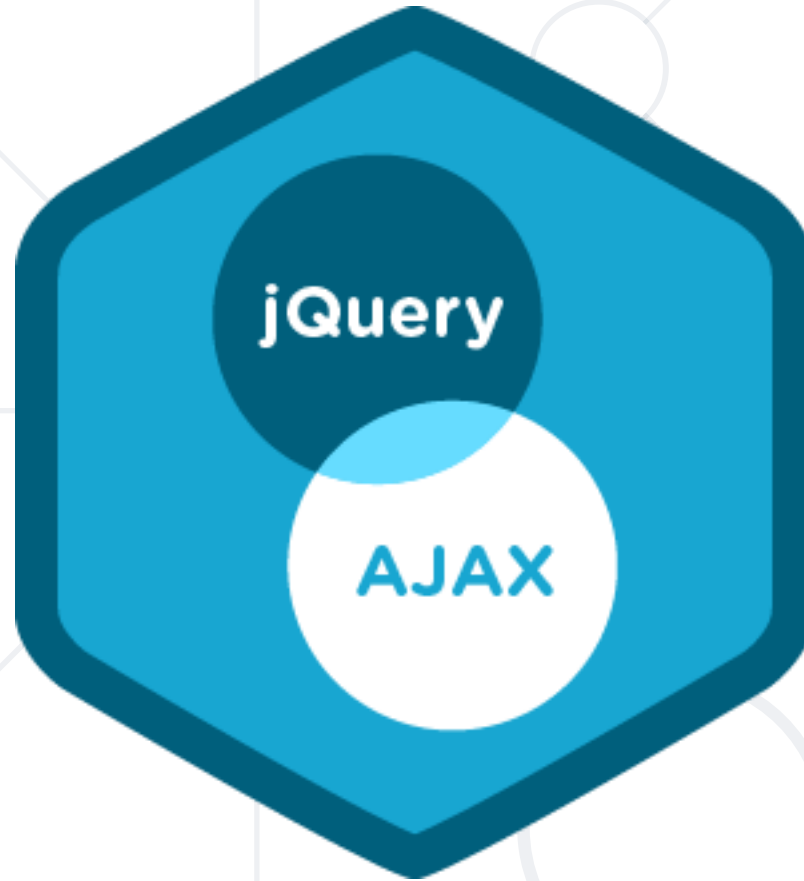
Using the XMLHttpRequest Object

XMLHttpRequest - Standard API for AJAX

```
<button onclick="loadRepos()">Load Repos</button>  
<div id="res"></div>
```

```
function loadRepos() {  
  let req = new XMLHttpRequest();  
  req.onreadystatechange = function() {  
    if (this.readyState == 4 && this.status == 200)  
      document.getElementById("res").textContent =  
        this.responseText;  
  };  
  req.open("GET",  
    "https://api.github.com/users/testnakov/repos", true);  
  req.send();  
}
```

Check your code here: <https://judge.softuni.bg/Contests/1568/Lab-AJAX>

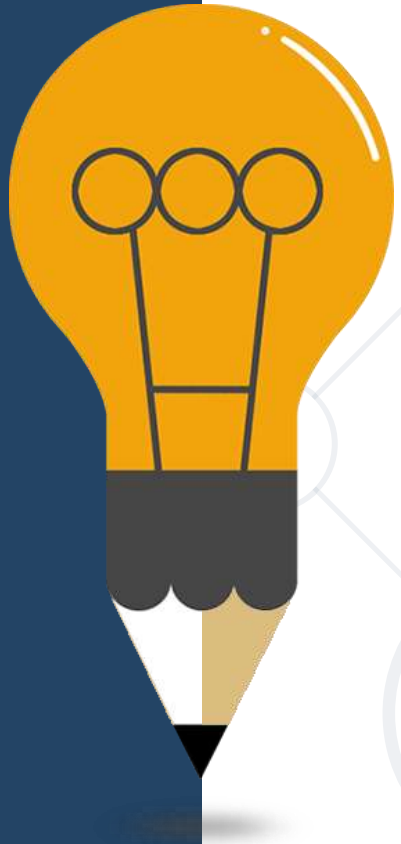


jQuery AJAX

Simplified AJAX Calls with jQuery

jQuery AJAX

Simplifies AJAX calls



- **Load** any **static/dynamic** data
- Pass data to the **server** using **data** parameter

jQuery methods can be called during the life cycle of **AJAX call** progress

- **ajaxSuccess** (callback) - callback for whenever an AJAX request completes **successfully**
- **ajaxStop**(callback) - callback for whenever **all AJAX requests** have **ended**

Problem: AJAX Load

Create a **page** holding a **button**. Clicking the button should **load** a **html fragment** and display it inside a **div**.

AJAX jQuery.load()

Load Title



Voilla!

I am a text loaded with AJAX request

Solution: AJAX Load

ajax-load.html

```
<div id="text">
  <h1>AJAX jQuery.load()</h1>
  <button onclick="loadTitle()">Load Title</button>
</div>
```

ajax-load.js

```
function loadTitle() {
  $('#text').load("text.html");
}
```

text.html

```
<h1>Voilla!</h1>
<p>I am a text loaded
with AJAX request</p>
```

Check your solution here: <https://judge.softuni.bg/Contests/1568/Lab-AJAX>

Handling Errors

Scripts can execute AJAX requests

- To their **origin** (same-origin policy)
- When the remote server explicitly allows AJAX calls via **CORS**

Uses a special HTTP header:
Access-Control-Allow-Origin

```
$(document).ajaxError(function(event, req, settings) {  
    $('#text').text(`Error loading data: ${req.status}  
    (${req.statusText})`);  
});  
function loadTitle() {  
    $('#text').load("http://dir.bg");  
}
```

This cross-origin AJAX request will fail due to missing CORS headers



Problem: Load GitHub Repos with AJAX

GitHub username:

```
<input type="text" id="username" value="k1r1L" />  
<button onclick="loadRepos()">Load Repos</button>
```

```
<ul id="repos"></ul>
```

```
function loadRepos() {  
    // AJAX call ...  
}
```

GitHub username:

- [k1r1L/Angular-2-Demos](#)
- [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](#)

Solution: Load GitHub Repos with AJAX

```
function loadRepos() {  
    $("#repos").empty();  
    let url = "https://api.github.com/users/" +  
        $("#username").val() + "/repos";  
    $.ajax({ url,  
        success: displayRepos,  
        error: displayError  
    });  
    function displayRepos(respos) { ... }  
    function displayError(err) { ... }  
}
```

Solution: Load GitHub Repos with AJAX (2)

```
function displayRepos(respos) {  
  for (let repo of respos) {  
    let link = $('<a>').text(repo.full_name);  
    link.attr('href', repo.html_url);  
    $("#repos").append($('<li>').append(link));  
  }  
}  
  
function displayError(err) {  
  $("#repos").append("<li>Error</li>");  
}
```

Check your solution here: <https://judge.softuni.bg/Contests/1568/Lab-AJAX>



AJAX - Examples

Using jQuery to Access REST APIs

Problem: Phonebook App in Firebase

Create a mini **phonebook app** that hold your data in **Firestore**. Disable the authentication to simplify your work. Implement "**list phones**", "**add phone**", "**delete phone**".

Phonebook

- Kiril: +359 234737343 Delete
- Maria: +359 545472651 Delete
- Todor: +359 452787917 Delete

Load

Create Contact

Person:

Phone:

Solution: Setup a Firebase DB

Create a new project at [Firebase](#) and enable public read/write access

Add a project

Project name

phonebook-nakov

Project ID

phonebook-nakov-f2782

Analytics location

Bulgaria

Cloud Firestore location

nam5 (us-central)

☒ Use the default settings for sharing Google Analytics for Firebase data

☒ Share your Analytics data with all Firebase features

☒ Share your Analytics data with Google to improve Google Products and Services

☒ Share your Analytics data with Google to enable technical support

☒ Share your Analytics data with Google to enable Benchmarking

☒ Share your Analytics data with Google Account Specialists

☒ I accept the [controller-controller terms](#). This is required when sharing Analytics data to improve Google Products and Services. [Learn more](#)

Cancel

Create project

phonebook-nakov

Database

Realtime Database

DataRulesBackupsUsage

1

2

3

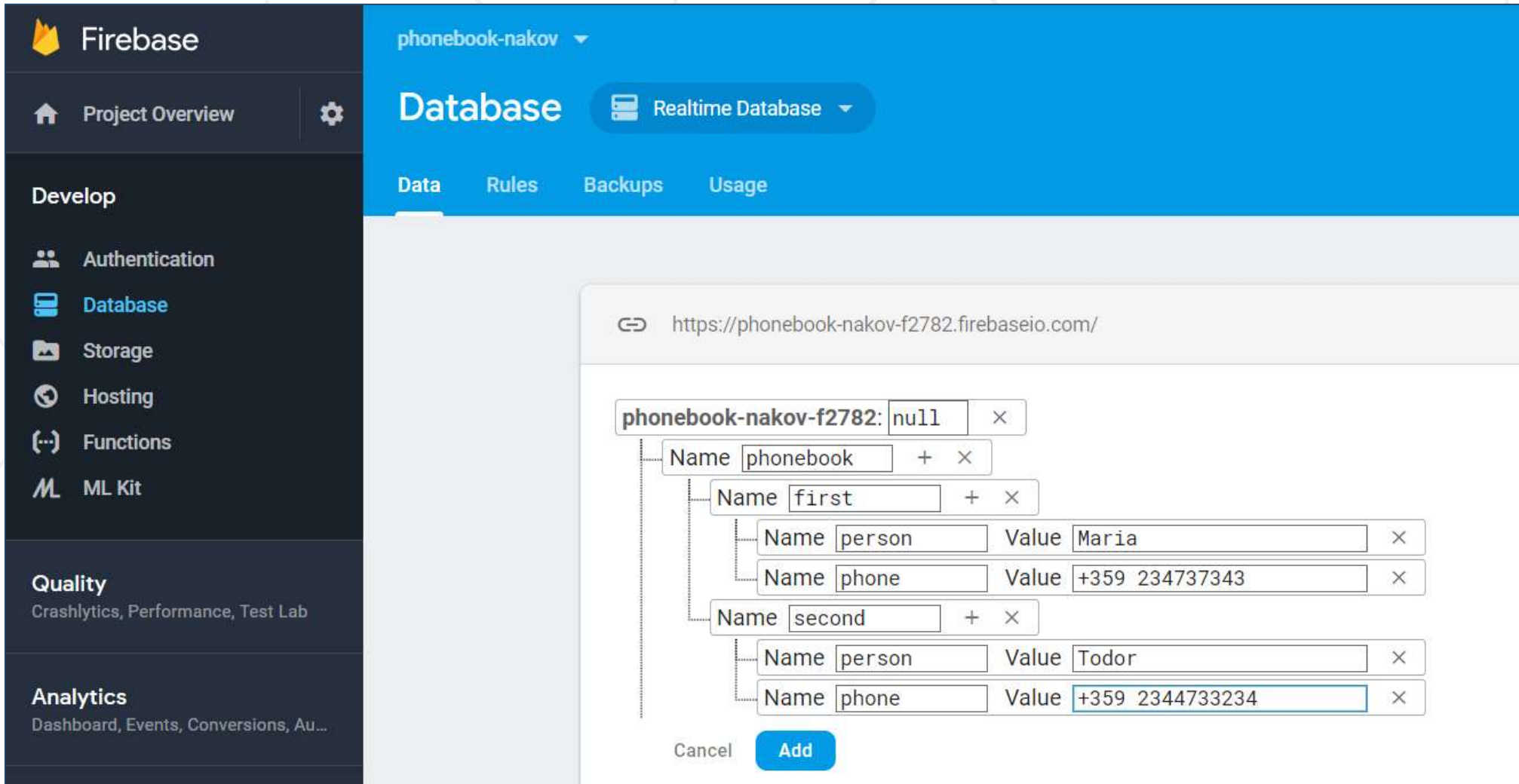
4

5

6

```
{  
  "rules": {  
    ".read": true,  
    ".write": true  
  }  
}
```

Solution: Add Sample Data in Firebase



The screenshot displays the Firebase console for a project named 'phonebook-nakov'. The left sidebar shows the 'Database' section selected under 'Develop'. The main area shows the 'Realtime Database' interface with tabs for 'Data', 'Rules', 'Backups', and 'Usage'. A data tree is visible, showing a root node 'phonebook-nakov-f2782' with a value of 'null'. Below this, there are two child nodes: 'first' and 'second'. Each of these nodes has a 'person' child node with a 'Value' field. The 'first' node's 'person' child has a 'Value' of 'Maria', and the 'second' node's 'person' child has a 'Value' of 'Todor'. At the bottom of the data tree, there is an 'Add' button and a 'Cancel' button.

phonebook-nakov

Database

Realtime Database

Data Rules Backups Usage

https://phonebook-nakov-f2782.firebaseio.com/

phonebook-nakov-f2782: null

Name phonebook

Name first

Name person Value Maria

Name phone Value +359 234737343

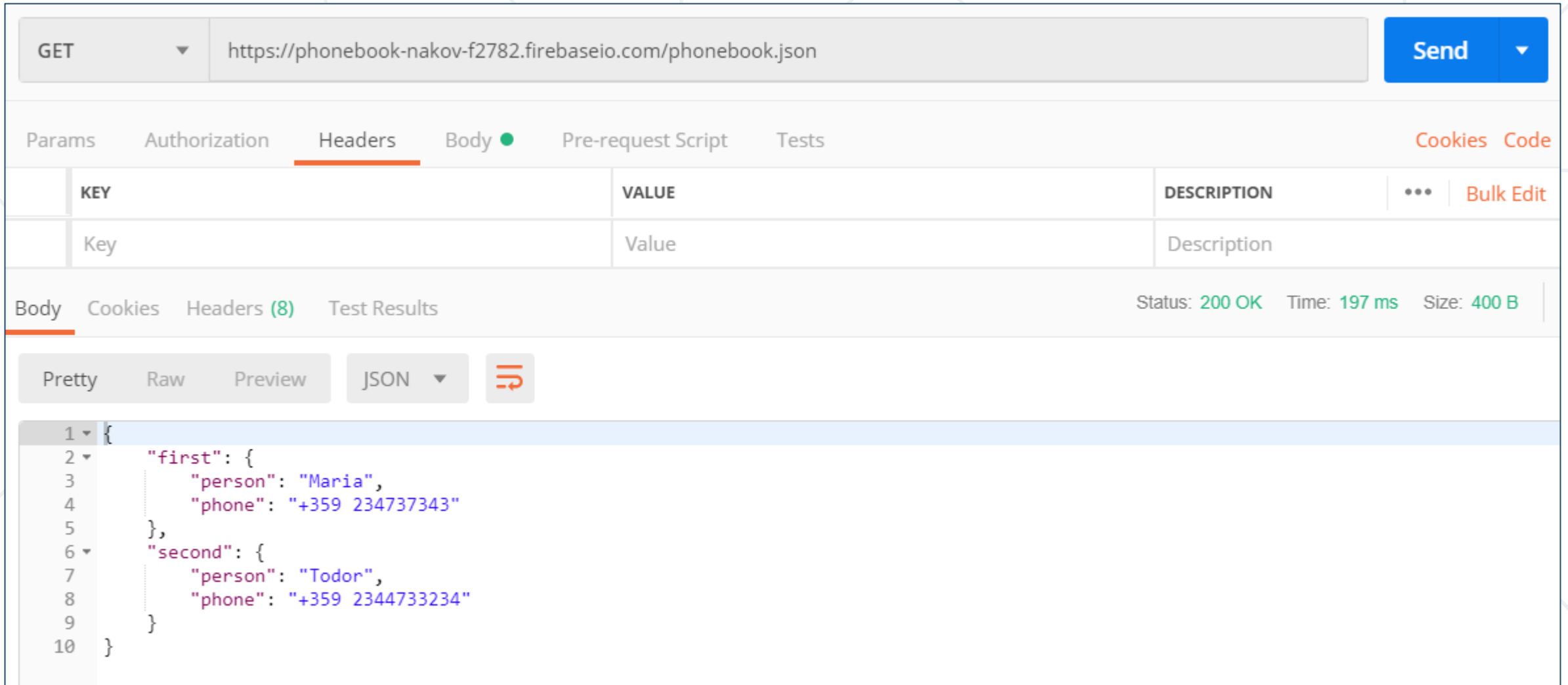
Name second

Name person Value Todor

Name phone Value +359 2344733234

Cancel Add

Solution: Test Your REST Service



The screenshot displays a REST client interface with the following components:

- Request Method and URL:** A dropdown menu shows "GET", and the URL bar contains "https://phonebook-nakov-f2782.firebaseio.com/phonebook.json". A blue "Send" button is located to the right.
- Request Headers:** A tabbed interface shows "Headers" selected. It contains a table with columns "KEY", "VALUE", and "DESCRIPTION". The table has one row with "Key", "Value", and "Description". A "Bulk Edit" link is visible on the right.
- Response Body:** A tabbed interface shows "Body" selected. The response is displayed in "JSON" format, showing a status of "200 OK", a time of "197 ms", and a size of "400 B". The JSON data is as follows:

```
1 {
2   "first": {
3     "person": "Maria",
4     "phone": "+359 234737343"
5   },
6   "second": {
7     "person": "Todor",
8     "phone": "+359 2344733234"
9   }
10 }
```

Solution: Phonebook in Firebase - HTML

```
<h1>Phonebook</h1>
<ul id="phonebook"></ul>
<button id="btnLoad">Load</button>
```

```
<h2>Create Contact</h2>
Person: <input type="text" id="person" />
<br>
Phone: <input type="text" id="phone" />
<br>
<button id="btnCreate">Create</button>
```

Phonebook

Load

Create Contact

Person:

Phone:

Create

Solution: Phonebook in Firebase - JS Code

```
$(function () {  
  $('#btnLoad').click(loadContacts);  
  $('#btnCreate').click(createContact);  
  let baseServiceUrl =  
    'https://phonebook-nakov.firebaseio.com/phonebook';  
  function loadContacts() { ... }  
  function displayContacts(contacts) { ... }  
  function displayError(err) { ... }  
  function createContact() { ... }  
  function deleteContact(key) { ... }  
});
```

Solution: Phonebook in Firebase - JS Code

```
function loadContacts() {  
    $("#phonebook").empty();  
    $.get(baseServiceUrl + '.json')  
        .then(displayContacts)  
        .catch(displayError);  
}  
  
function displayError(err) {  
    $("#phonebook").append($("<li>Error</li>"));  
}
```


Solution: Phonebook in Firebase - JS Code

```
function displayContacts(contacts) {  
  for (let key in contacts) {  
    let person = contacts[key]['person'];  
    let phone = contacts[key]['phone'];  
    let li = $("- ");  
    li.text(person + ': ' + phone + ' ');  
    $("#phonebook").append(li);  
    li.append($("<button>Delete</button>"))  
      .click(deleteContact.bind(this, key));  
  }  
}

```

Bind the event handler with
the **current key**

Solution: Phonebook in Firebase - JS Code

```
function createContact() {  
    let newContactJSON = JSON.stringify({  
        person: $('#person').val(),  
        phone: $('#phone').val()  
    });  
    $.post(baseServiceUrl + '.json', newContactJSON)  
        .then(loadContacts)  
        .catch(displayError);  
    $('#person').val('');  
    $('#phone').val('');  
}
```

Solution: Phonebook in Firebase - JS Code

```
function deleteContact(key) {  
  let request = {  
    method: 'DELETE',  
    url: baseUrl + '/' + key + '.json'  
  };  
  $.ajax(request)  
    .then(loadContacts)  
    .catch(displayError);  
}
```

The correct contact **key** will come as parameter (due to binding)

Check your solution here: <https://judge.softuni.bg/Contests/1568/Lab-AJAX>



Practice: jQuery AJAX

Live Exercises in Class (Lab)

- **AJAX** == Asynchronous JavaScript and XML
- AJAX sends **asynchronous HTTP** requests
- Scripts can **execute** AJAX requests
- jQuery **simplifies** AJAX calls
- The **\$.ajax()** function underlies all AJAX requests sent by jQuery



Questions?



SoftUni



**Software
University**



**SoftUni
Svetlina**



**SoftUni
Creative**



**SoftUni
Digital**



**SoftUni
Foundation**



**SoftUni
Kids**

SoftUni Diamond Partners



XSsoftware



SBTech
we know sports



telenor



SoftwareGroup
doing it right

NETPEAK



SmartIT



Postbank

Решения за твоето утре

**SUPER
HOSTING
.BG**

INDEAVR

Serving the high achievers



INFRAGISTICS®

LIEBHERR



aeternity



SoftUni Organizational Partners



OneBit
SOFTWARE



WORLD
OF
MYTHS

Trainings @ Software University (SoftUni)

- Software University - High-Quality Education and Employment Opportunities
 - softuni.bg
- Software University Foundation
 - <http://softuni.foundation/>
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University Forums
 - forum.softuni.bg



- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license

