

Lab: Asynchronous Programming

1. Github Commits

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

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

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> (<error.statusText>)"`

Screenshots:

The screenshot shows a web form with two input fields: "GitHub username:" containing "nakov" and "Repo:" containing "nakov.io.cin". A green "Load Commits" button is to the right of the Repo field. Below the inputs, a bulleted list displays four commit messages from Svetlin Nakov: "Delete Console.Cin.v11.suo", "Create LICENSE", "Update README.md", and "Added better documentation".

The screenshot shows the same web form but with "GitHub username:" set to "Innos" and "Repo:" set to "Examss". The "Load Commits" button is green. Below the inputs, a bulleted list shows "Error: 404 (Not Found)". To the right, a browser's developer console is open, showing a red error message: "GET https://api.github.com/repos/Innos/Examss/commits 404 (Not Found)".

2. Blog

Write a program for reading blog content. It needs to make **requests** to the **server** and display **all blog posts** and their **comments**.

Firebase URL - <https://blog-apps-c12bf.firebaseio.com/>

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

The button with ID **"btnLoadPosts"** should make a **GET** request to **"/posts"**. The **response** from the **server** will be an **Object of objects**.

```

▼ { -LhdbZ31ND3Rhw41UGmN: {...}, -Lhdc-Ttz9-KiW9uvh6W: {...}, -LhdcLmyARLEB1bsSvjZ: {...}, -LhdccRyWr_7UCPtclmM: {...} }
  ► -LhdbZ31ND3Rhw41UGmN: {body: "An asynchronous model allows multiple things to happen...the result (for example, the data read from disk).", id: "rrt87..."}
  ► -Lhdc-Ttz9-KiW9uvh6W: {body: "In a synchronous programming model, things happen...stops your program for the time the action takes.", id: "rrt87..."}
  ► -LhdcLmyARLEB1bsSvjZ: {body: "One approach to asynchronous programming is to make...the callback function is called with the result.", id: "rrt87..."}
  ► -LhdccRyWr_7UCPtclmM: {body: "Working with abstract concepts is often easier when...turn an object that represents this future event.", id: "rrt87..."}
  ► __proto__: Object

```

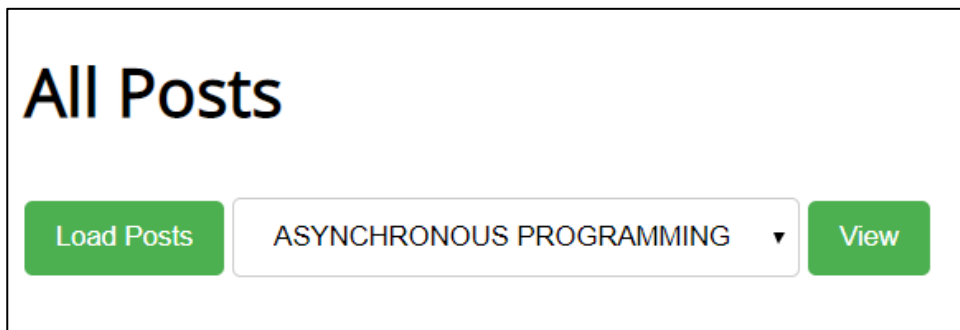
Each object will be in the following format:

```

{
  body: {postBody},
  id: {postId},
  title: {postTitle}
}

```

Create an **<option>** for each post using its **object key** as value and **current object title property** as text inside the node with ID **"posts"**.



```

▼ <select id="posts">
  <option value="-LhdbZ31ND3Rhw41UGmN">ASYNCHRONOUS PROGRAMMING</option>
  <option value="-Lhdc-Ttz9-KiW9uvh6W">SYNCHRONOUS PROGRAMMING</option>
  <option value="-LhdcLmyARLEB1bsSvjZ">CALLBACKS</option>
  <option value="-LhdccRyWr_7UCPtclmM">PROMISES</option>
</select>

```

When the button with ID **"btnViewPost"** is clicked, a **GET** request should be made to:

- **"/posts/{postId}"** to obtain the selected post (from the dropdown menu with ID **"posts"**) - The following **request** will return a **single object** as described above.
- **"/comments"** - to obtain all comments. The request will **return a Object of objects**.

```

▼ { -Lhdewt02LJrzuThWlmj: {...}, -LhdfHFg8dNxK-qUaukL: {...}, -LhdfVg4JDka0Cft-dQZ: {...}, -LhdfuAXo1mPycgRRf-3: {...}, -Lhdg0x8QG-j2vnNUhL5: {...}, ... }
  ► -Lhdewt02LJrzuThWlmj: {id: "rrt8713kx1jda5r", postId: "rrt875tgjxlingqb", text: "So good article. Nice!"}
  ► -LhdfHFg8dNxK-qUaukL: {id: "rrt878p0jx1jdgze", postId: "rrt875tgjxlingqb", text: "Rly helpful. Thanks!"}
  ► -LhdfVg4JDka0Cft-dQZ: {id: "rrt879ccjx1jdo03", postId: "rrt879rkjxlimol2", text: "Now I understand it... Thanks!"}
  ► -LhdfuAXo1mPycgRRf-3: {id: "rrz123cjxhhfdoti443", postId: "rrt87twjxlimswr", text: "Amazing article! Good job!"}
  ► -Lhdg0x8QG-j2vnNUhL5: {id: "rrz123smshhfdoti543", postId: "rrt87twjxlimswr", text: "You are the best! +1 For this Article!"}
  ► -LhdgPKif5sYtjYn61SQ: {id: "rrz35smshhfdoti543", postId: "rrt87btjxlimxui", text: "Good job my man! You are the best!"}
  ► -LhdgZvm5UCF6eo5vU6g: {id: "rrz35ssshhfdoti444", postId: "rrt87btjxlimxui", text: "AMAZING ARTICLE! It's was pleasure to read it! Thanks bro!"}
  ► -Lhdgh3EH01FrB09CCp: {id: "rrz404smshhfdoti404", postId: "rrt87btjxlimxui", text: "It was ok, next time you will crush them!"}
  ► __proto__: Object

```

Each object will be in the following format:

```
{  
  id: {commentId},  
  postId: {postId},  
  text: {commentText}  
}
```

You have to find this comments that are for the current post (check the **postId** property)

Display the post title inside **h1** with ID "**post-title**" and the post content inside **ul** with ID "**post-body**". Display **each comment** as a **** inside **ul** with ID "**post-comments**". Do not forget to clear its content beforehand.

ASYNCHRONOUS PROGRAMMING

An asynchronous model allows multiple things to happen at the same time. When you start an action, your program continues to run. When the action finishes, the program is informed and gets access to the result (for example, the data read from disk).

Comments

- So good article. Nice!
- Rly helpful. Thanks!

```
<h1 id="post-title">ASYNCHRONOUS PROGRAMMING</h1>  
▼<p id="post-body">  
  "An asynchronous model allows multiple things to happen at the same time. When you start an action, your program  
  continues to run. When the action finishes, the program is informed and gets access to the result (for example, the  
  data read from disk)."  
</p>  
<h2>Comments</h2>  
▼<ul id="post-comments">  
  <li id="rrt8713kx1jda5r">So good article. Nice!</li>  
  <li id="rrt878p0jx1jdgze">Rly helpful. Thanks!</li>  
</ul>
```