Lab 9 – Interacting with REST APIs using JS

1 Introduction

Welcome to Week 9! …we're almost there!

This lab is extremely important for your coursework. Over the previous weeks you have developed a good understanding of web fundamentals. You can now: create HTML pages, style them with CSS, make them work across devices with Flexbox, and now you can validate form input with Regular Expressions. The last piece of the puzzle is interacting with servers (I created for you, and you are learning server-side programming next year) and other devices through APIs. Further to this, you can make API calls within your web pages to create an element of interactivity. This is how all modern websites work and is the difference between a static website and a rich web application.

This lab is the follow-up to the last lab. So, if you haven't finished API with postman, you need to do that first. If you missed this week's lecture, you also need to read it first.

2 Learning Objectives

- To practice interacting with REST APIs through JavaScript

3 REST APIs in JS

3.1 Exercise 1: Create a HTML form

The HTML form can be similar to that in Slide 11 of Lecture 9.

The next set of exercises replicate the code that was demonstrated in the lecture.

Start by creating a form for adding a Movie to the API. The form should have three inputs and a submit button. Make sure to add all required ID's so that can be referenced in JS.

3.2 Exercise 2: Validate the form

Add an event listener which is triggered when the forms submit button is clicked. The function should validate the input using what you learnt in the lab last week. Your validation should check the following:

1. That all fields have been entered
2. All fields should be the right data type
3. The year should be greater than 1900, and less that or equal to the current year (check W3Schools for finding the current year)
4. The movie name and director should be less than or equal to 32 characters.

3.3 Exercise 3: Send a POST request to the server

If the form passes your validation checks, you can send to the server. Use the lecture slides to help you. Test that your application is working using Postman.

3.4 Exercise 4: Send a GET request to the server

Using the lecture slides to help you, add a table to your HTML page that is automatically populated with the MovieDB list of movies when the page loads. Edit your POST request so that the table is updated when a new movie is added.

3.5 Exercise 5: Pretty error handling

Add a stylesheet to your web page and create two class styles: one for errors and one for success. Add an empty tag to your web page for providing feedback to users. Using the lecture slides to help you, can you provide feedback to the user on whether their form submission was successful or not. Use the lecture slides to add your class styles depending on the message displayed.

3.6 Exercise 6: PATCH and DELETE

The previous exercises were a lot of copying and pasting from the lecture slides. This exercise is a true test of understanding. Can you add functionality into your application for updating and deleting records in your table and the MovieDB API? (do the delete first as it is easier)

3.7 Exercise 7: Your assignment

If you can complete all exercises above, you will be able to do the coursework!
If there is time, start implementing the API calls within your assignment project.