

World Wide Web and Mobile Applications

CSC 130 - Lab B01 and B02

Teaching Team

Department of Computer Science
University of Victoria

Lab 08 and Lab 09



Computer Science

Table of Contents

Admin

Lab 08 - Restful API

- Part 01 - A RESTFUL Web Service GET Request
- Part 02 - Handling the JSON Returned
- Part 03 - Creating Images Out of the Movie Rating
- Bonus - Add ALT Text to MPAA Standard Rating Image

Lab 09 - JSON and Babylon.js

- Part 01 - Data Models and JSON
- Part 02 - Babylon.js
- Part 03 - GitHub
- Bonus - Upload Project 01 and/or Project 02 Code to GitHub

Finished

Table of Contents

Admin

Lab 08 - Restful API

- Part 01 - A RESTFUL Web Service GET Request
- Part 02 - Handling the JSON Returned
- Part 03 - Creating Images Out of the Movie Rating
- Bonus - Add ALT Text to MPAA Standard Rating Image

Lab 09 - JSON and Babylon.js

- Part 01 - Data Models and JSON
- Part 02 - Babylon.js
- Part 03 - GitHub
- Bonus - Upload Project 01 and/or Project 02 Code to GitHub

Finished

All CSC 130 TA Lab Notes:

<https://notes.dominiquecharlebois.com/ta/csc130.html>

- Lab 08 and Lab 09 are due at the end of the lab today.
- **Lab Material:** Brightspace CSC 130 Course - Content - Day 10: Lab 8! and Lab 9!
- Checkpoints are *optional*, thus feel free to work at your own pace and demonstrate your work at the end.
- Ask Questions! Answer Questions! Collaborate with Peers!
- **Resource:** <https://developer.mozilla.org/en-US/>

Credit is awarded by demonstrating your work at the end of each lab:

- Students who complete all required material will obtain 3/4.
- Students who complete bonus material *AND* collaborate with peers will obtain 4/4.

You are expected to collaborate with peers by asking and answering questions.

Table of Contents

Admin

Lab 08 - Restful API

- Part 01 - A RESTFUL Web Service GET Request
- Part 02 - Handling the JSON Returned
- Part 03 - Creating Images Out of the Movie Rating
- Bonus - Add ALT Text to MPAA Standard Rating Image

Lab 09 - JSON and Babylon.js

- Part 01 - Data Models and JSON
- Part 02 - Babylon.js
- Part 03 - GitHub
- Bonus - Upload Project 01 and/or Project 02 Code to GitHub

Finished

Objectives

- Understand how to make a get request to a RESTFUL web services.
- More practice working with JSON objects References/Resources.
- Translation between JavaScript and jQuery.

Examples

- Open Movie Database API: -
`https://jsfiddle.net/ycoady/4en790k3/24/`

Resources

- `https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Working_with_Objects`
- `https://developer.mozilla.org/en-us/docs/Web/API/Document_Object_Model`
- `https://www.w3schools.com/jquery/jquery_get_started.asp`
- `https://www.w3schools.com/js/js_jquery_selectors.asp`

Starter Files

- Download the ZIP folder named **lab8-starter** within a folder named **csc130**.
- Unzip the folder named **lab8-starter**.

The folder named **lab8-starter** contains one HTML file, one CSS file, one JS file.

Check Point 01a

- Open up the lab8.html file in a text editor.
- Add the references to the JS and CSS files, and a link to jQuery into the HEAD section.
- Open up the lab8.html in a web browser. Verify that everything is connected correctly.
- Open up lab8.js in a text editor.

Resources:

<https://www.redhat.com/en/topics/api/what-is-a-rest-api>

Check Point 01b

- Create an account with Open Movie Database:
<http://www.omdbapi.com/apikey.aspx>.
- Use the RESTFUL API example to get the information from the user and then retrieve the data from Open Movie Database.

Hint: [http://www.omdbapi.com/?apikey=\[yourKey\]&t=\[movieTitle\]&y=\[movieYear\]](http://www.omdbapi.com/?apikey=[yourKey]&t=[movieTitle]&y=[movieYear])

where *[yourKey]* will be replaced by your key from the Open Movie Database (e.g., <http://www.omdbapi.com/?apikey=01234>).

Resources: <https://jsfiddle.net/ycoady/4en790k3/24/> and <http://www.omdbapi.com/>

Check Point 01c

- Complete the *submit()* function in the file named **lab8.js**.

Check Point 02

- Complete the *displayResults()* function in the file named **lab8.js**.
- Determine which attributes in the JSON data we need to access.
- Add the data dynamically to the HTML file with JavaScript.

Check Point 03

- Use the MPAA standard ratings with the corresponding MPAA standard image to add movie ratings to our displayed results (e.g., G, PG-13, R, M, etc.).
- Add the JavaScript variables (g, pg, pg13, r, and unrated) with the appropriate URL.
- Using the movie rating of the movie, display one of these MPAA standard rating images in addition to the text.

Hint: You will need to use if-statements to conditionally render the content based on the rating.

Bonus Check Point

- Add the *alt* attribute to the MPAA standard images.
- Set the *alt* text to correspond with the MPAA standard rating (i.e., A **G** rating will have **General Audience** as the *alt* text).

Table of Contents

Admin

Lab 08 - Restful API

- Part 01 - A RESTFUL Web Service GET Request
- Part 02 - Handling the JSON Returned
- Part 03 - Creating Images Out of the Movie Rating
- Bonus - Add ALT Text to MPAA Standard Rating Image

Lab 09 - JSON and Babylon.js

- Part 01 - Data Models and JSON
- Part 02 - Babylon.js
- Part 03 - GitHub
- Bonus - Upload Project 01 and/or Project 02 Code to GitHub

Finished

Objectives

- Practice thinking about JSON, arrays, and loops.
- Introduction to 3D using Babylon.js.
- Become familiar with GitHub and Repositories.

Check Point 01a

- Explore the type of *data model* you will use in Project 02 (i.e., how many objects in the array, stored values, names of attributes in the objects).

Resources: <https://www.imdb.com/interfaces/> or
https://bookshelf.erwin.com/bookshelf/public_html/2021R1/Content/References/Data%20Modeling%20overview/Data%20Model%20Example.html

Check Point 01b

- Create a design of your data model for your Project 2 (i.e., a hand drawn sketch, or list of formatted data elements).
- Demonstrate your **mocked up** JSON object.

Note: The JSON object can be from a server, but it does not have to be from a server.

Check Point 02

- Create your first Babylon.js app, as described in the tutorial.
- Modify *Playground 5* from the tutorial (e.g., modify the colour, texture, yeti, etc.).
- Demonstrate your modified *Playground 5*.

Tutorial: <https://doc.babylonjs.com/journey/theFirstStep#everyones-very-first-step>

Check Point 03

- Create the GitHub Repository for the Babylon.js Application.
- Upload the Babylon.js code to the GitHub Repository.

Bonus Check Point

- Upload Project 01 and/or Project 02 to GitHub.

Table of Contents

Admin

Lab 08 - Restful API

- Part 01 - A RESTFUL Web Service GET Request
- Part 02 - Handling the JSON Returned
- Part 03 - Creating Images Out of the Movie Rating
- Bonus - Add ALT Text to MPAA Standard Rating Image

Lab 09 - JSON and Babylon.js

- Part 01 - Data Models and JSON
- Part 02 - Babylon.js
- Part 03 - GitHub
- Bonus - Upload Project 01 and/or Project 02 Code to GitHub

Finished

Lab 08 and Lab 09 are Finished!

Final Steps

- Upload files to Brightspace.
- Add link to text field in Brightspace submission.