Intended learning outcomes assessed by this work:

1. Develop a secure, open-standards-based API to support server-client communication.

2. Create modern web content involving asynchronous data retrieval, client-side DOM manipulation, standards adherence and user-user interaction

3. Manage data persistence across both server and client web-based solutions.

4. Design and implement an API and client based on given, non-trivial requirements resolving ethical issues surrounding user-data and user-generated content

Hand Out Date: 26 August 2019
Submission Date: 26 November 2019, by 11.59pm.
Assignment – 100% out of marks.

Task Description

Assignment Brief

You are required to use JavaScript to develop a full-stack cloud-based RESTful web API application using suitable third-party API(s). You should select a topic that interests you and ensure that there is data available for you to use via any public API(s) that you choose to use. For the higher grades you will need to build a simple client (web or smartphone) to show your API in action. After selecting a topic to develop, write a one paragraph proposal briefly describing the project and submit to the Lecturer for prior approval before commencing work on it.

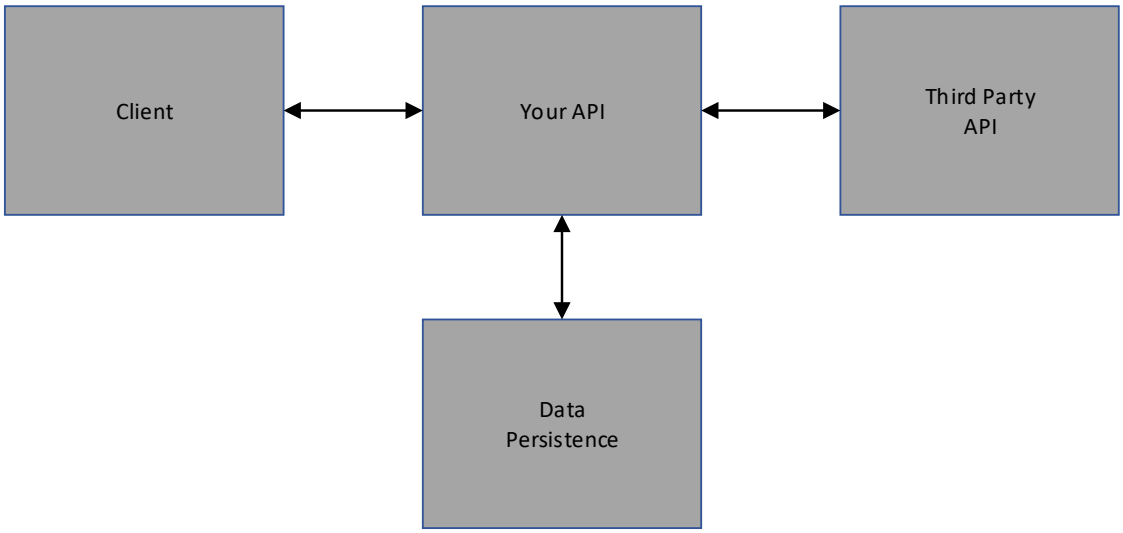
Here are some websites that lists publicly available APIs to get you started:

https://github.com/toddmotto/public-apis
https://apilist.fun/
https://any-api.com/
https://console.bluemix.net/developer/watson/documentation
https://developers.google.com/apis-explorer/#p/

However, you may also use any other API(s) not listed in any of the websites above.

Architecture

The diagram below may help you understand the requirements. Notice that any client will interact with your API and that this API should interact with both the third-party API(s) and with your data persistence solution.



Data

Your API will search online data sources such as API(s) or websites to obtain relevant data. However any results should be persisted within your API(s). How you persist this data is entirely up to you. However you will need to justify your choice(s) based on the features available. Options include persisting the data to the filesystem or using an appropriate database. You should carefully select which data you will be saving from the third-party API(s) and only display and persist this. You will lose marks if you persist everything you get back from the third-party API(s).

Features

The feature sets that is to be implemented is provided in the marking rubric. Each column in the marking rubric describes the minimum requirement to qualify for a particular grade. To achieve a high grade, you are expected to implement all the features with good programming competency, understanding and further extend on them.

Remember to use the Grading Rubric as your guide as you work on this assignment.

Programming Language

Use the Node.js framework and JavaScript on the server with any appropriate packages. Please note that you will get a zero grade if you choose other languages.

Justify your choices of Node.js packages modules in the reflective video.

Submission

This assignment will be due at 11:59pm on the Due date mentioned in page 1. You must submit your assignment earlier.

You are required to submit three components:

1. a link to your live API
2. links to your Git remote (hosted on GitHub)
3. a link to your reflective video (hosted on YouTube)
    1. The length of the video MUST be 5 – 10 minutes only.
    2. The video should contain the demonstration of the features of your API, showing the front end client, plus your reflection

Live API

You need to deploy your latest API(s) release on a cloud service such as Heroku so your lecturers can test its functionality. Make sure it is deployed and functional before the assignment deadline and that the deployed version matches the source code you submit.

Source Code

You are required to track your API code and your client code in Git repositories. You will be required to submit links to your Git hosted on GitHub and should be public so that your lecturers can access them. Both repositories should include full documentation available through the home page. Run ESLint to demonstrate code quality.

Front End Client

The front end client can be a Website or Smart Phone Apps that can compose Web API request and present Web API response. You can use any tools to build the Website or Smart Phone Apps.

For a higher marks it should be able to

1. Do visual presentation of the Web API response,
2. Do asynchronous requests and dynamically updates client UI.
3. Responsiveness to error messages, states and conditional response from Web API.
4. Present mashup response from multiple Web APIs.

Please refer to the marking rubric for the mark details.

Video & Report

Once you have completed the API and the client you need to explain and critically analyse (critique) how your API and client work by recording a screencast and upload to YouTube. This should cover all the points in the grading criteria and demonstrate your skills and knowledge of the subject.

After uploading you must change the video permissions from private to unlisted. This will allow the lecturers to access the video via the link but prevent it being publicly searchable. Make sure you separately demonstrate the API (using Postman) and the client and ensure you justify your choices of language constructs and architecture.

1. Demonstrate your API using Postman, showing the requests and responses (headers/body). For the highest grades you will also be expected to demonstrate your API integrated into a client such as a website or smartphone app.

2. Demonstrate the back-end persistence showing how the data is stored.

3. Demonstrate the front end client.

4. Explain your code with sufficient details.

Presentation:

Present your system in class. You may play your uploaded YouTube video and take questions from the floor, or, use PowerPoint slides, or both. Be prepared to demonstrate your live API during presentation.

304CEM Grading Rubric

(Criteria) Marks	(I) 0	(II) 1 – 5	(III) 6 – 10	(IV) 11 – 15	(V) 16 - 20	(VI) 21 – 25
Web API 25%	No API demonstrated or not working as expected.	Simple functional API demonstrating a basic understanding of REST principles. Implemented GET request for resources and collections.	Meeting criteria II. GET request that includes conditional, filtering and sorting.	Overall good work. Meeting criteria II to III. The API demonstrates a good understanding of REST-ful Web API with DELETE and POST request.	An overall outstanding work. Meeting criteria II to IV. Used linter. API is fully REST compliant. Provides feedback for invalid requests through appropriate response codes and messages.	An overall exemplary work. Meeting criteria II to V. API demonstrates user registration and authentication.
Interactive Client 25%	No interactive client to present Web API response.	Simple UI to help users compose Web API request.	Meeting criteria II. Simple visual presentation of the Web API response.	Overall good work. Meeting criteria II & III. Asynchronous requests and dynamically updates client UI.	An overall outstanding work. Meeting criteria II to IV. Responsiveness to error messages, states and conditional response from Web API.	An overall exemplary work. Meeting criteria II to V. Present mashup response from multiple Web APIs. .
Data Persistence 25%	No data persistence layer.	Process responses from third party Web API and persist them.	Meeting criteria II. Query the persisted data and return the results to the Web client.	Overall good work. Meeting criteria II & III. Can update and delete the data in the persistence layer.	An overall outstanding work. Meeting criteria II to IV. Storing user information and configurations on client side, or, server side.	An overall exemplary work. Meeting criteria II to V. Well designed data persistence layer. Fully complies to requirements and standards.
Reflective Video 25%	No video link submitted. No video done.	Little critical discussion and incomplete. Some attempt at explanation. But no understanding of subject matter. Presentation unclear.	Meeting criteria II. Some critical discussion but incomplete. Limited understanding of subject. Presentation skills are ok but could be better.	Overall acceptable discussion & critical analysis. Meeting criteria II & III. Basic understanding. Good presentation with clarity	Very good discussion and critical evaluation. Meeting ciriteria II to IV. Demonstrates good subject understanding. Confident and convincing presentation.	Excellent discussion with strong critical analysis. Meeting criteria II to V. Well researched discussion and critical evaluation. Demonstrates deep understanding