Assignment Zero

Purpose

The purpose of this assignment is to work with Postman become familiar with HTTP, and REST through the testing framework provided by Postman. Furthermore, to check in your first node program to github

You will create a Postman collection and create a REST test within the project. You will need to automate each test and include the required asserts (tests) for each request in the validation.

Requirements

- Sign-up for a free GitHub account: https://github.com/ – Homework assignments will be stored on GitHub.
  - Create individual repo: CSC3916_HW0
- Sign-up for a free WebStorm license (needed for future assignments): https://www.jetbrains.com/webstorm/
  - https://www.jetbrains.com/student/
- Create a REST request to get started:
  - Create an environment variable book_title for the query string for title
  - Url: https://www.googleapis.com/books/v1/volumes?q={{book_title}}
  - Use this request to get a JSON response of books with the name Turing in the title.
  - Parse the result and store the id in a postman variable
  - Asserts must include
    - validating books returned have the title turing (e.g. items[i].title)
    - Response status code (e.g. 200)
- Create a chained request that requests
  - Url: https://www.googleapis.com/books/v1/volumes/{{id}}
  - Using the id you stored from the first request, make sure the second request uses the ID pulled from the first request
  - Create Asserts that:
    - Validate response contains the ID from the request
    - Validate response status code (e.g. 200)
- Modify googlebooks.js
  - Return an object that contains

```
{
        data: response.data,
        status: response.status,
        statusText: response.statusText,
        headers: response.headers,
        requestHeader: response.config.headers
}
```

  - Review the HTTP Headers in the Request and Response – create text file **headers.txt** and describe each key value pair in the HTTP header in both request and response and check it in with the project to GitHub (e.g. what is the content-type and what does it mean)

Submissions

- Create a readme.md at the root of your github repository with the embedded (markdown) to your test collection
  - Within the collection click the (…), share collection → Embed
    - Static Button
    - Click update link
    - Include your environment settings
    - Copy to clipboard
- Check-in googlebooks command API with your change to google books
- Check-in text file headers.txt with header values
- Submit the Url to canvas with the REPO CSC_3916
  - (e.g. https://github.com/shawnmccarthy/week1)

Resources

Postman: https://www.getpostman.com/

http://blog.getpostman.com/2014/01/27/extracting-data-from-responses-and-chaining-requests/

https://github.com/shawnmccarthy/week1

```
var issues = JSON.parse(responseBody);

var closedState = postman.getEnvironmentVariable("closed_state");

var allIssuesAreClosed = issues.every(function(issue) {

  return issue.state === closedState;

});

tests["All issues are closed"] = allIssuesAreClosed;
```