CSCI 406 – Advanced Web Development
Lab 4 – REST API Website

For this lab you will start with your Blogger application from Lab 3 as the basis and modify it to include a REST API for interacting with its data. Once the REST API is done, you will then be updating the front-end and **app_server** controllers in order to use the REST API for its data needs. Lastly, you will publish this version of your application to your previously created GitHub repository as a new branch named: "Lab 4".

PUBLIC SERVICE ANNOUNCEMENT: This lab will take some time. Expect to spend at least 4 to 6 hours developing, testing and updating the REST API required. In addition, you will likely have to spend about that same amount of time (4 to 6 hours) updating views and controllers on the front-end in order for your pages to consume and interact with the REST APIs. Along the way, expect to get stumped and have to solve problems (e.g. database not returning data, syntax errors in JavaScript, environment not installed with all required modules, etc.) and that problem solving to take significant time.

DUE DATE: *This lab is due to the instructor by Monday, October 12th at 11:59pm.*

Instructions:

1. Using your Amazon Lightsail MEAN instance, make a copy of your previous Lab 3 application either using Linux copy commands or GIT commands to pull down the previous application to a new (different than previous) directory.

2. Your application will be similar to the application you wrote in lab 3 but will be updated to include and consume a REST API.

3. Your web application must adhere to the following specification:

Specification:
   i. **Port 80:** Your application must load from a browser via port 80, which is the standard HTTP port for a server.

   ii. **Add an "app_api" directory:** Your program should be updated such that it includes an API area holding the fundamental pieces of your REST API.

   iii. **Create a REST API:** Having read part "*2:6 – Writing a REST API*" you will be armed with the information and know how in order to create the "/api/" endpoints that interact with the MongoDB in order to:

a) ***return a list of blogs*** (GET "/api/blogs/")

b) ***return a single blog*** given an id (GET "/api/blogs/5c6b00d4fd313a7adef90b8a"),

c) **add a blog** (POST "/api/blogs/")

d) ***update a blog*** given an id (PUT "/api/blogs/5c6b00d4fd313a7adef90b8a")

e) ***delete a blog*** given an id (DELETE "/api/blogs/5c6b00d4fd313a7adef90b8a")

Important: Test the endpoints using Postman (see the book for details) and/or a browser by interacting with the /api/ URL endpoints.

iv. **Consume the REST API:** Having read part "*2:7 – Consuming a REST API*" you are now armed with the information and know how required to update your application's controllers and views (in **app_server**) to use the new REST API.

When finished you will have pages that perform the following:

a) Show a list of blogs, allowing the user to click one to edit or delete

b) Add a new blog

c) Edit an existing blog and save it

d) Delete an existing blog (or cancel deletion)

4. Your application must be setup to run even with your MEAN instance is not connected.

5. Using your GitHub account and the repository created and used prior for Lab 2, save this lab as a new branch ("Lab 4").

6. Below is the instructor's example (*please remove any sample blogs you add*):

   http://13.58.205.86/

7. Get your application working, first by getting the REST API up, running and tested, and then by integrating the API into controllers and views for (in the order below), also testing as you go:
   a. listing the blogs
   b. adding a blog
   c. editing a blog
   d. deleting a blog

8. Once you have the application working and it is available via port 80 even when disconnected from your MEAN instance and you have your app uploaded to your GitHub

repo, please send the full URL of your app AND your GitHub repo to the instructor via email with subject of "Lab4". Important:

mailto:thomas.rogers@millersville.edu?subject=Lab4