Lab 2
Due 2/2, before class

Take the lab you created in lab 1, and instead of having your frontend read from a JSON object one your hard drive (or someone else's API), you are going to refactor it to retrieve data via an API of your design.

If you read a JSON object from your hard drive, you're good. If you got your data from API(s), create a JSON object that has 200 articles and save it to your hard drive.

Create a new Node project, name it "Lab 2" and put your name as the author. Install Express (and any other libraries you want).

You should create an API with Express that does the following:

1. GET / = retrieve your HTML page from Lab 1
2. GET /news = retrieve a listing of article numbers (hint: you can use headers for this if you don't want to send the list in the body of your response; hint2: consider pagination, since you have a lot of articles!)
3. GET /news/### = retrieve the specific article from the JSON object
4. POST /news = append a news article to the end of the JSON object
5. PUT /news = bulk update all your articles
6. PUT /news/### = update the specific article
7. DELETE /news/### = delete the specific article

(Please don't make DELETE /news)

You may rewrite any/all parts of your frontend to make your life easier. Your frontend ***may not*** directly grab any news articles: it may only get articles via the API. You may need to create a form or some other interactive feature to POST, PUT, and DELETE articles.

You must serve your project on your VM. In addition, the TA will run "npm i" on their local machine to see if your package.json file works correctly.

You may discuss the design and implementation of the lab as a group. But, as with Lab 1, you must code your solutions individually.

If you don't cite your sources, you will get a 0 for the lab, no exceptions no apologies later.

Grading:

API creation	20 pts (this includes your POST, PUT, DELETE interactive feature)
Project.json	10 pts
Creativity	10 pts
README.md	10 pts
TOTAL	**50 pts**