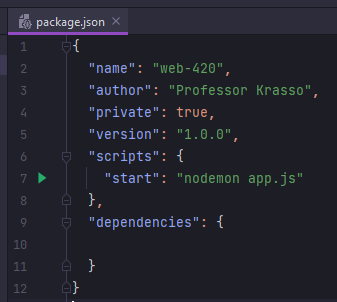
**Assignment 1 – GitHub and Project Setup**

**Instructions**

* ~~Add a package.json file to the root of the repository with the entries identified in Exhibit A.~~

**Exhibit A. package.json without dependencies**



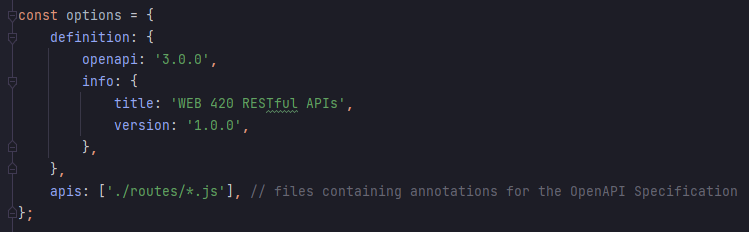
* ~~Using the terminal window, install the following npm package: bluebird, express, mongoose, swagger-jsdoc, swagger-ui-express, and nodemon (set nodemon to be installed as dev dependency).~~
  + ~~Note: the version numbers in Exhibit B. must match the versions you install. Otherwise, the code you write may not function the way I describe in the assignment’s instruction.~~

**Exhibit B. package.json with dependencies (as of 4-13-21)**



* ~~Add an app.js file with “require” statements for express, http, swagger-ui-express, swagger-jsdoc, and mongoose.~~
* ~~Create a new variable named app and assign it to express library (refer back to the solutions you built in web-340, if you need refresher).~~
* ~~Set the port to process.env.PORT || 3000~~
* ~~Set the app to use express.json()~~
* ~~Set the app to use express.urlencoded({‘extended’: true});~~
* ~~Define an object literal named options with the following properties/values~~

**Exhibit C. options object literal**



* ~~Create a new variable name openapiSpecification and call the swaggerJsdoc library using the options object literal. For example, const openapiSpecification = swaggerJsdoc(options);~~
* ~~Wire the openapiSpecification variable to the app variable (see Exhibit D).~~

**Exhibit D. Wiring the swaggerSpec variable**

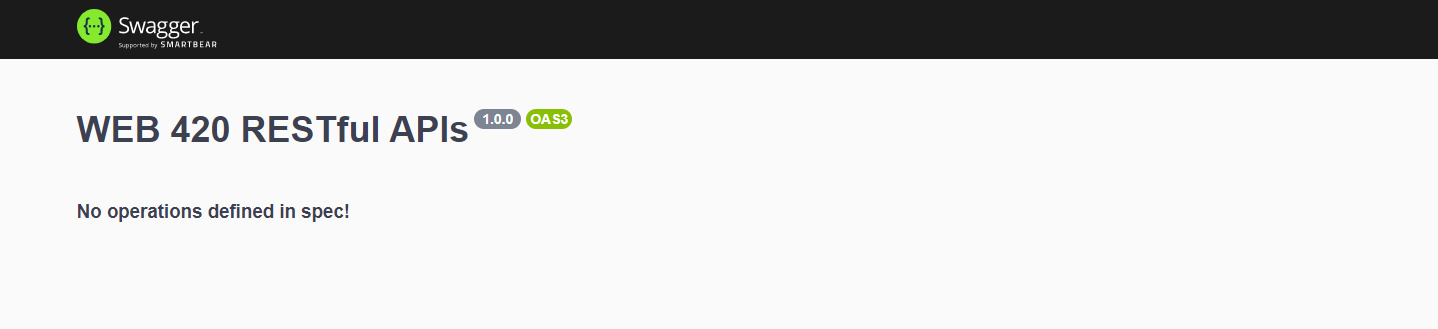


* Finally, use the http library to create a new server that listens on the port you set (port 3000). In the body of the createServer() function add a console.log() statement that says, “Application started and listening on port <portNumber>.”

**Testing**

* Using the terminal window, run the command **npm start**
* Next, open a new browser window and visit the URL: <http://localhost:3000/api-docs>
* Assuming the code you wrote matches the instructions, you should see what’s illustrated in Exhibit E. The most important part is the “green pill” with the text value of OAS3. This essentially means the OpenAPI Specification is set to 3.0.0. Make sure your version matches this. Otherwise, the code you write may not perform/match the assignments instructions.

**Exhibit E.** [**http://localhost:3000/api-docs**](http://localhost:3000/api-docs)



Note: with the exception of create the options object literal and wiring the openapiSpecification variable to the express library, everything else was covered in web-340. If you run into issues or need assistance, please refer back to the work you completed in WEB 340 Node.js