

COMPX322-24A: Assignment Three

Due Date: *Monday May 20th, 10 am*

Web Services: *REST API for Event Management Application*

For this coursework you are required to implement a REST API for an Event Management Application, which allows users to manage their events. You will use:

- HTTP Verbs to make requests [**C**(POST), **R**(GET), **U**(PUT), **D**(DELETE)]
- HTTP Response codes to indicate status
- MySQL database
- Node.js and Express.js backend

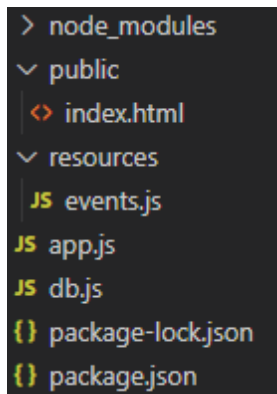
Application Description

This is a simple Node.js RESTful CRUD API using Express and a MySQL database. This RESTful API should be able to respond to the following requests:

- Create a new event and save it to the database.
- Get a list of all events in the database.
- Get events by ID.
- Get events by Name.
- Update event information by ID.
- Delete events by ID.
- Delete all events from the database.

Implementation

You have been provided with a **EventsDatabase.sql** file to create the database table. For the directory of your assignment implementation, use *restful-assn3*. Your directory should contain the following files and folders:



NOTE: Do not include the node_modules folder in your submission!

What you need to do:

1. Create the project directory. Initialise the project by running **npm init -y**.
2. Use npm to install **express**, **cors**, and **mysql2**.
3. Create any remaining files and folders so that your directory matches the image shown above.
4. Download and run **EventsDatabase.sql** on your MySQL instance to create an *events* table with five rows inserted in the table.
5. Your **app.js** file should create an Express app that listens on port **3000**. The Express app should serve static files from the **public** folder.
6. Your **index.html** file should be a simple page that gives a brief description of your API functions. Include the method (e.g. GET) and path (e.g. '/events').
7. Your **db.js** file should create a connection to your MySQL database and export that connection using **CommonJS** syntax.
8. Your **events.js** file should **require** the database connection, define the event model and handler functions, then export them using **CommonJS** syntax.
9. Your **app.js** file should define the routes, using the handlers from your **events.js** file.
10. Your API should support the following CRUD functions:
 - a. Create a new event and save it to the database.
 - b. Get a list of all events in the database.
 - c. Get events by ID.
 - d. Get events by Name.
 - e. Update event information by ID.
 - f. Delete events by ID.
 - g. Delete all events from the database.
11. Test your API using Postman. Add a brief description of at least one of your Postman tests to the bottom of your **index.html** page.

What to Submit and How

All pertinent material you have developed for this assignment must be submitted electronically using Moodle. The submitted files must be sufficient to recreate your app by running *npm install* followed by *npm start*. **Do not include your *node_modules* directory, as this is not needed to reconstitute your project using *npm install*. Moreover, it could contain binaries files specific to the computer platform you developed the assignment on that are incompatible with the computer system used to test your submitted assignment solution.**

You may choose between submitting a ZIP file or a 'tar-ball' (*.tar.gz*). For the former, use the name *restful-assn3.zip* and *restful-assn3.tar.gz* for the latter. **Marks will be deducted for submitted assignments that do not meet these requirements.**

In the COMPX322 Moodle site, you will see an *Assignment 3* hyperlink to the submission page. This link allows you to upload your *tar.gz/zip* file. You can do this as many times as you want up to the submission deadline for the assignment.

When you submit a file, Moodle will ask you to confirm that what you have submitted is your own work, and will provide you with a 'receipt' that establishes that you have indeed submitted something. No other mechanism for submission will be accepted.

How Your Work will be Assessed

The assignment will be marked out of 50 as follows:

API meets functional requirements: <ul style="list-style-type: none">• Create event• Retrieve all events• Retrieve event by event ID• Retrieve event by event Name• Update event by event ID• Delete event by event ID• Delete all events	35 marks
Set up the Routes	5 marks
Index.html <i>Contains a description of each of the API functions, as well as a description of at least one of your Postman tests</i>	5 marks
Directory contains the correct files and folders. Code is relatively clean and has comments	5 marks

The deduction for incorrectly submitted files is capped at 2 marks.