**CS628 Full-Stack Development II – Backend**

**PE10 - Basic Node and Express - Get Data from POST Requests**

Developed by Clark Ngo on December 29, 2020

School of Technology & Computing (STC)

City University of Seattle (CityU)

**Before You Start**

* Some steps are not explained in the tutorial**.** If you are not sure what to do:
  1. Consult the resources listed below.
  2. If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

Students will be able to:

* Learn Basics of Node and Express

**Resources**

* https://www.freecodecamp.org/learn/apis-and-microservices/basic-node-and-express/get-data-from-post-requests

**How to Submit**

* **Upload .zip file**
* **Write a 150-word summary to explain your understandings and findings from this lab assignment.**

## **Basic Node and Express - Get Data from POST Requests**

Mount a POST handler at the path /name. It’s the same path as before. We have prepared a form in the html frontpage. It will submit the same data of exercise 10 (Query string). If the body-parser is configured correctly, you should find the parameters in the object req.body. Have a look at the usual library example:

route: POST '/library'  
urlencoded\_body: userId=546&bookId=6754  
req.body: {userId: '546', bookId: '6754'}

Respond with the same JSON object as before: {name: 'firstname lastname'}. Test if your endpoint works using the html form we provided in the app frontpage.

Tip: There are several other http methods other than GET and POST. And by convention there is a correspondence between the http verb, and the operation you are going to execute on the server. The conventional mapping is:

POST (sometimes PUT) - Create a new resource using the information sent with the request,

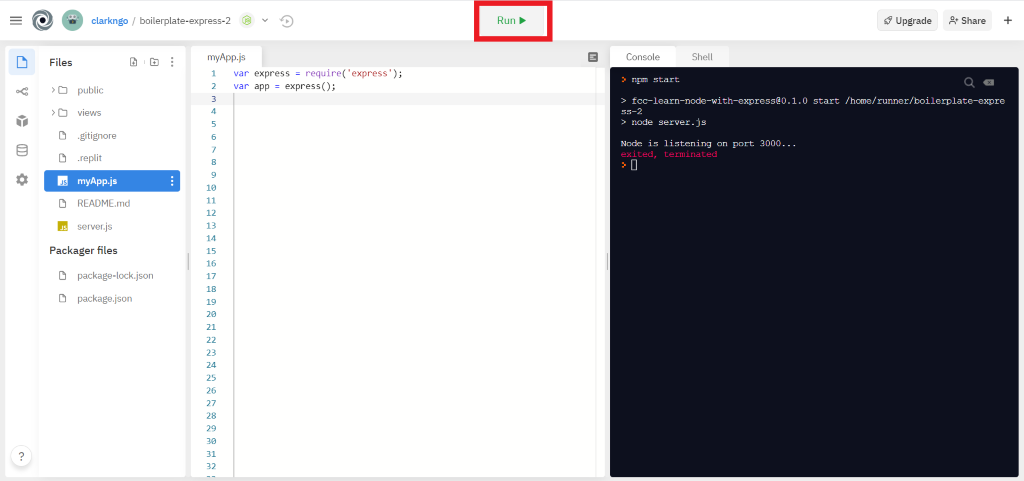
GET - Read an existing resource without modifying it,

PUT or PATCH (sometimes POST) - Update a resource using the data sent,

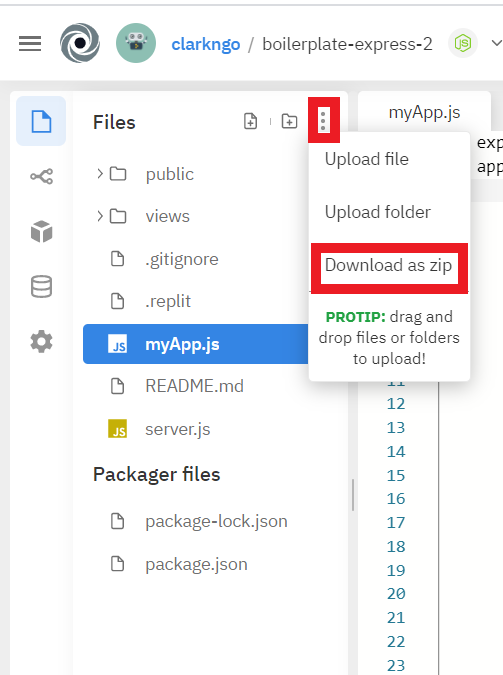
DELETE => Delete a resource.

There are also a couple of other methods which are used to negotiate a connection with the server. Except from GET, all the other methods listed above can have a payload (i.e. the data into the request body). The body-parser middleware works with these methods as well.

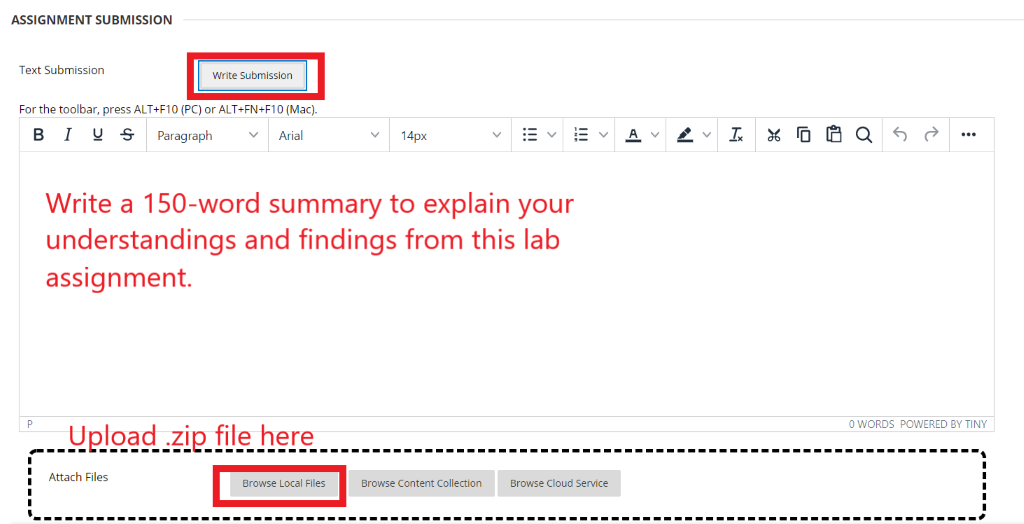
1. Start a new project on Repl.it using [this link](https://repl.it/github/freeCodeCamp/boilerplate-express).
2. Click Run to test your code



1. Download the project as .zip.



1. In BlackBoard, upload the .zip file and write a 150-word summary to explain your understandings and findings from this lab assignment.



The index file we created in a much earlier module is not used to send a post request to the post request handler web service. The index.html file has a form that targets /name with the method post.

<form action="/name" method="post">

<label>First Name :</label>

<input type="text" name="first" value="John"><br>

<label>Last Name :</label>

<input type="text" name="last" value="Doe"><br><br>

<input type="submit" value="Submit">

</form>