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

**PE08 - Basic Node and Express - Get Route Parameter Input from the Client**

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-route-parameter-input-from-the-client

**How to Submit**

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

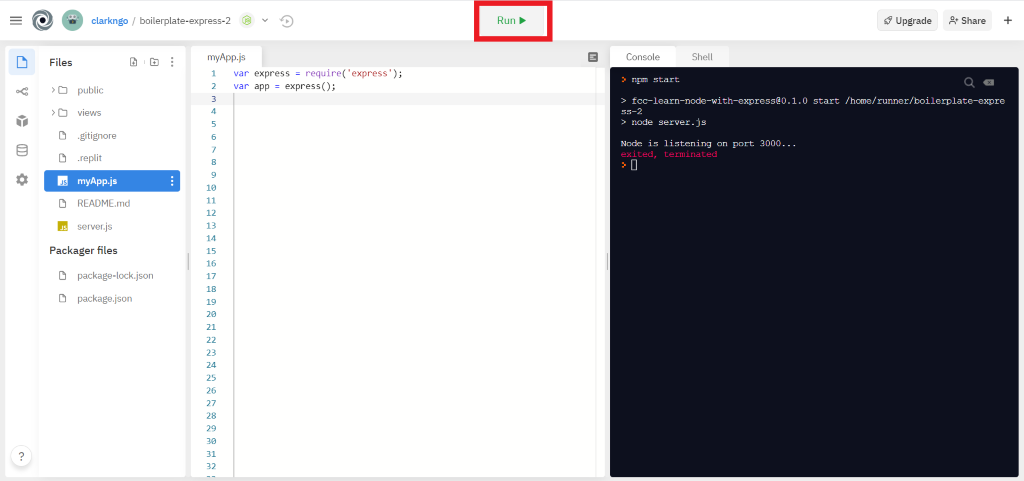
## **Basic Node and Express - Get Route Parameter Input from the Client**

When building an API, we have to allow users to communicate to us what they want to get from our service. For example, if the client is requesting information about a user stored in the database, they need a way to let us know which user they're interested in. One possible way to achieve this result is by using route parameters. Route parameters are named segments of the URL, delimited by slashes (/). Each segment captures the value of the part of the URL which matches its position. The captured values can be found in the req.params object.

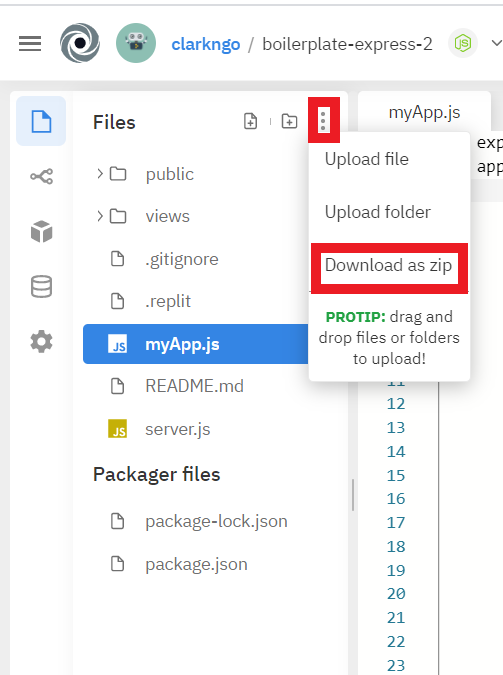
route\_path: '/user/:userId/book/:bookId'  
actual\_request\_URL: '/user/546/book/6754'  
req.params: {userId: '546', bookId: '6754'}

Build an echo server, mounted at the route GET /:word/echo. Respond with a JSON object, taking the structure {echo: word}. You can find the word to be repeated at req.params.word. You can test your route from your browser's address bar, visiting some matching routes, e.g. your-app-rootpath/freecodecamp/echo.

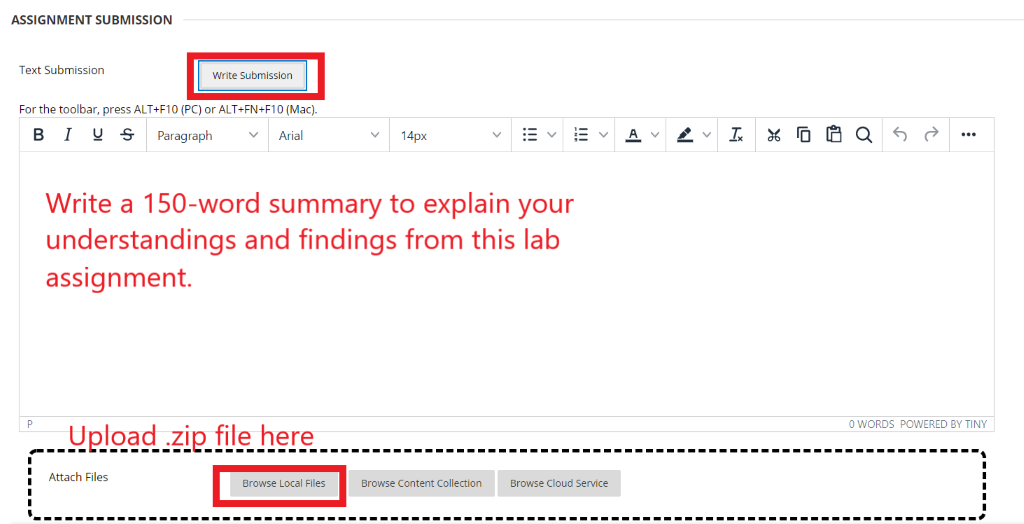
1. Start a new project on Repl.it using [this link](https://repl.it/github/freeCodeCamp/boilerplate-express).
2. Click Run to test your code



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.



When we were doing dynamic web programming, the user could navigate to different pages using either the URL or the links. And the user could provide inputs through either the URL or the components such as textfields.

When we are doing Web Service programming (also called API programming), one way to send the request and its parameters is through the URL. But the way we provide parameters is different from the way we do it in dynamic web programming. Over there, we had to sent parameters in the following form

<https://www.google.com/search?q=test>

In Web Services, even a “parameterless” URL can contain parameters. The parts of the URL can be used as parameters.

route\_path: '/user/:userId/book/:bookId'  
 actual\_request\_URL: '/user/546/book/6754'  
 req.params: {userId: '546', bookId: '6754'}

We use the code below. When the user goes to projectURL/:word/echo, we service the user with this particular response. The :word can be any string.

app.get("/:word/echo", (req, res) => {

const { word } = req.params;

res.json({

echo: word

});

});

