Serverless Development 101

**HOP02C – Integrate Serverless Development Tools**

1/10/2019 Developed by Kevin Wang

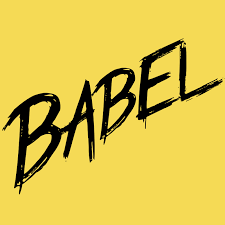
1/10/2019 Checked by Clark Jason Ngo

1/10/2019 Tested by Tuan Khai and Minh Truong

2/4/2019 Revised by Sam Chung

4/9/2020 Reviewed by Apiwat Chuaphan

Center for Information Assurance (CIAE) @City University of Seattle (CityU)

In this activity, we will initialize a NodeJS project, install all dependences, and integrate with Babel and a Linter.

Note:

* Babel is a JavaScript compiler (static analysis tool) that can convert the new syntax code into a backwards compatible version of JavaScript. This will enable JavaScript to run in both current and older browsers and environments. More information: <https://babeljs.io/>
* A linter can help to analyze source code in order to flag programming errors, bugs, stylistic errors, and suspicious constructs. More information: <https://eslint.org/>
* We will use the Airbnb’s JavaScript Style Guide. More information: <https://github.com/airbnb/javascript>

**Learning Outcomes**

* Learn to integrate the NodeJS, Babel, and ESLinter.

**Initialize a NodeJS Project and set up the development environment**

1. Open your terminal (Windows: Windows + R, then “cmd”. MacOS: Command + Space, then “terminal”).
2. Go to your working directory called ‘cs547’ and navigate into the folder.  
   >>> cd /cs547  
   Windows user

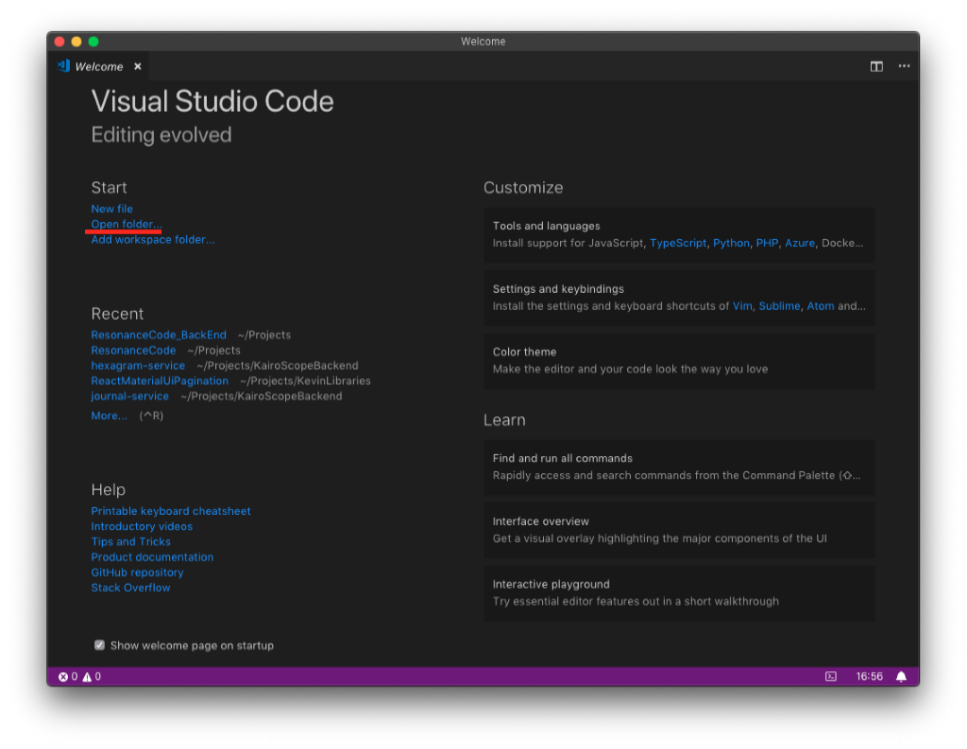
>>> dir

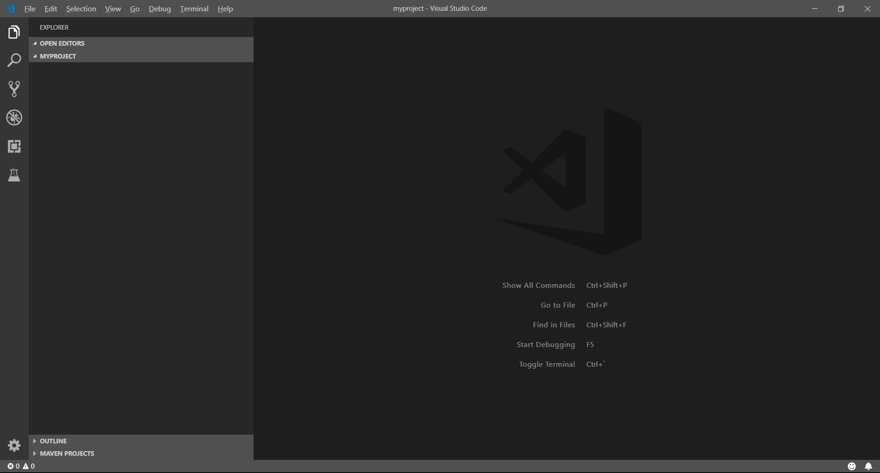
Or Mac user

>>> ls

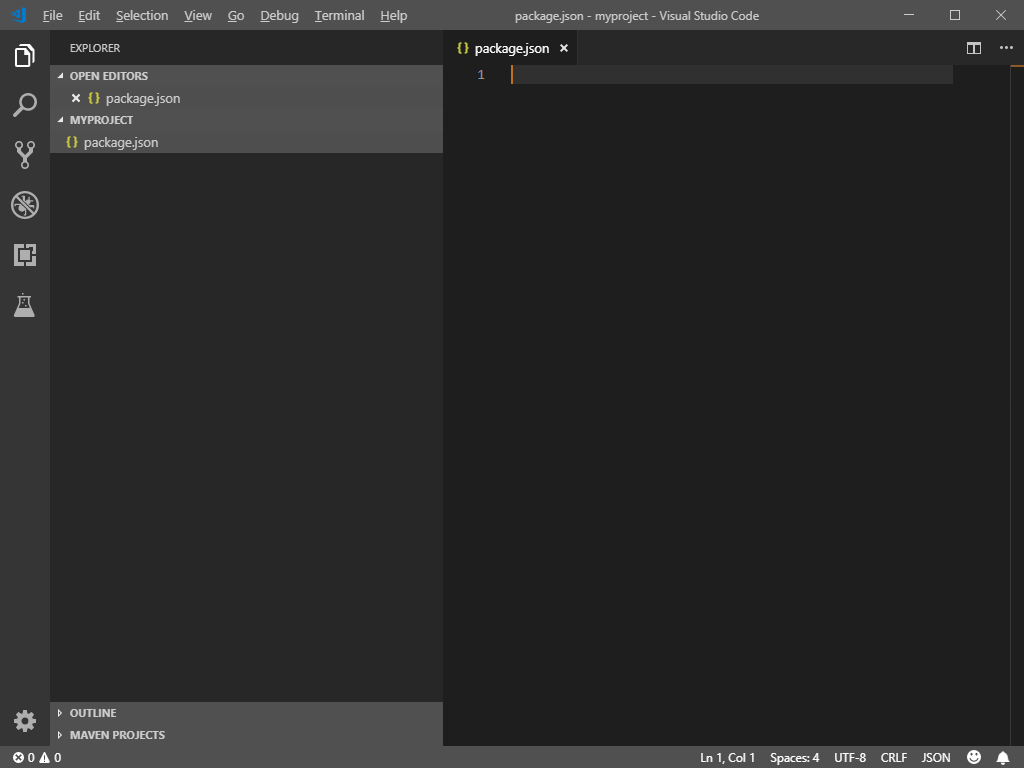
1. Create a folder (with command “mkdir”) and access (with command “cd”) that folder directory.   
   Type in the terminal: (Changed the yourfoldername to the name you want).   
   In this case, we chose ‘**bookstore’**  
   >> mkdir bookstore   
   >> cd bookstore|

1. Open the project in your VSCode.  
   Click the “Open folder” (or from the file menu and click open) and select the folder that we just created.

****

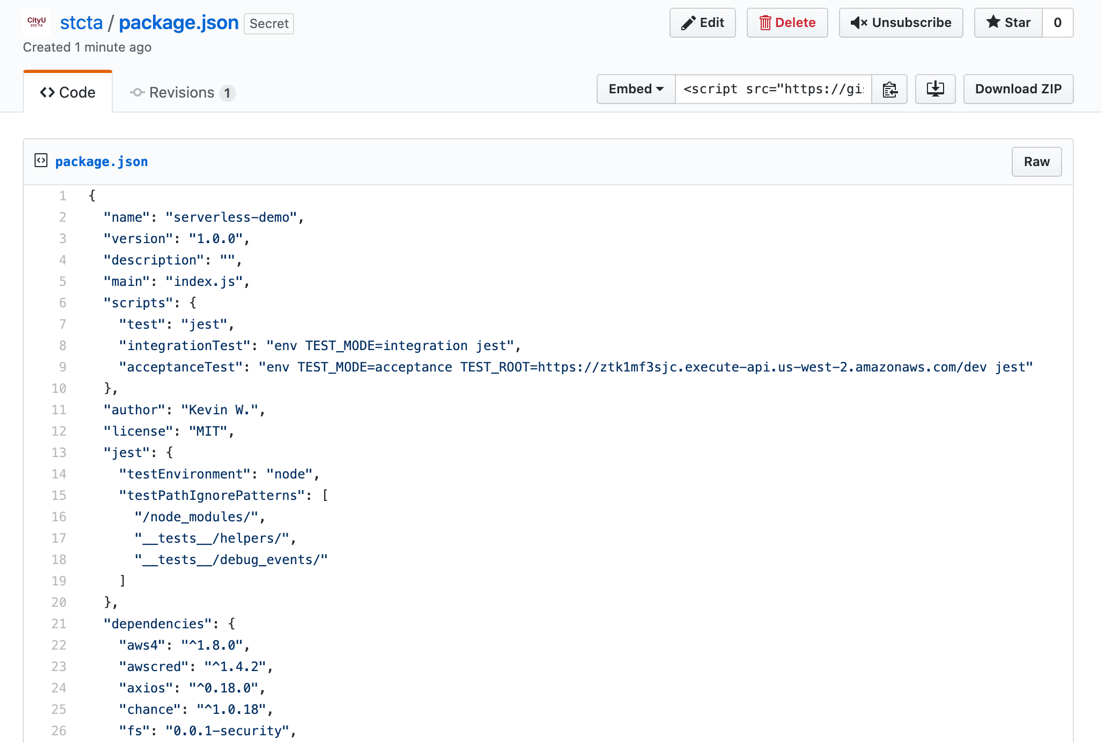


1. Install NPM package dependences.
   1. Create a ‘**package.json**’ file in the root folder, which is **bookstore**



* 1. Copy the content from [here](https://bit.ly/3ebnsEU)Press the “Raw” button” and copy all of them into your ‘package.jason’.

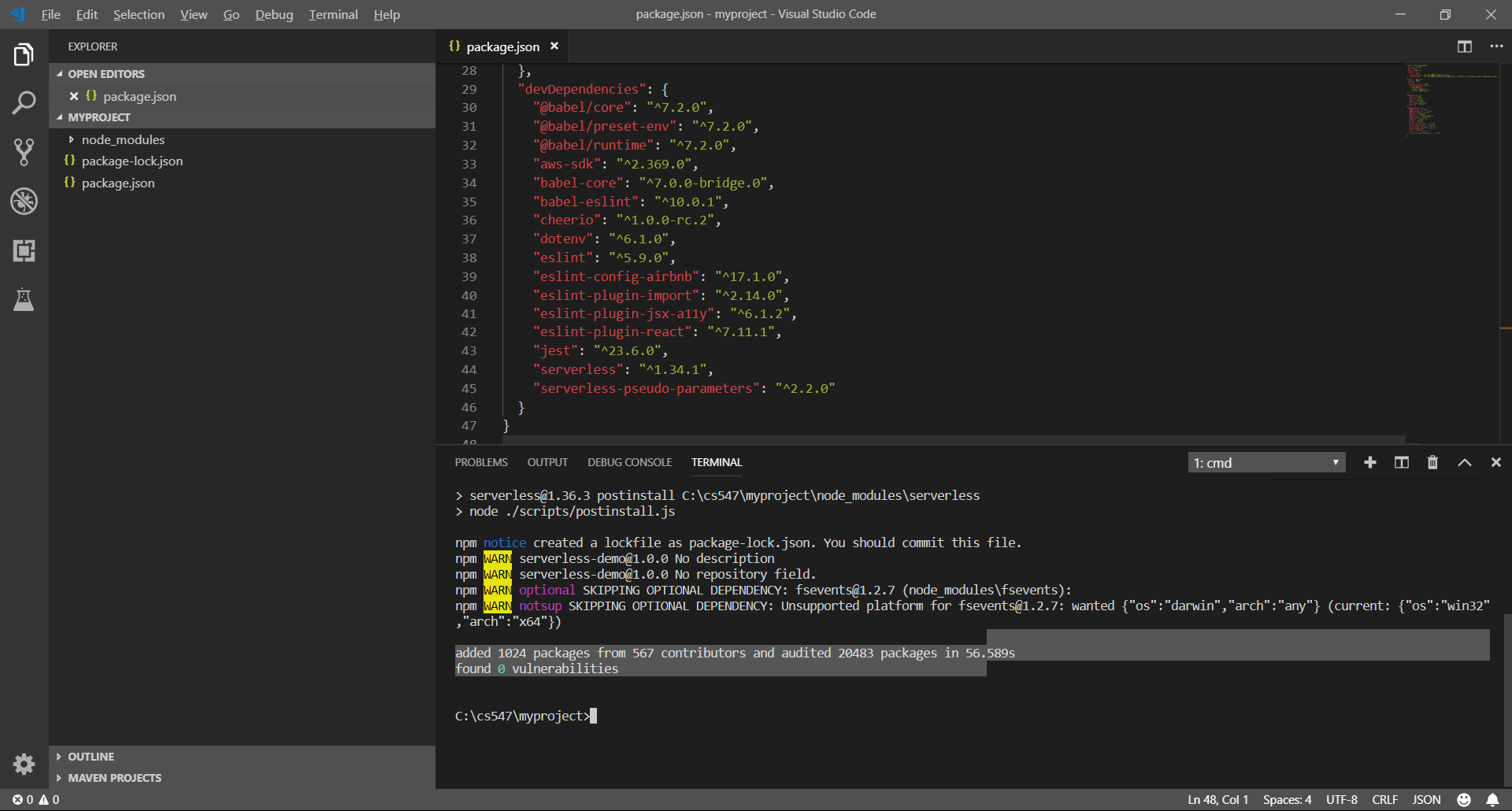
Change the name and author if you want.  
Save your ‘package.json’



* 1. Open the terminal in the VSCode (shortcut key: control + ~).
  2. Type “**npm**i” in the terminal to install all dependence includes in the ‘package.json’ file.

A screenshot of a computer

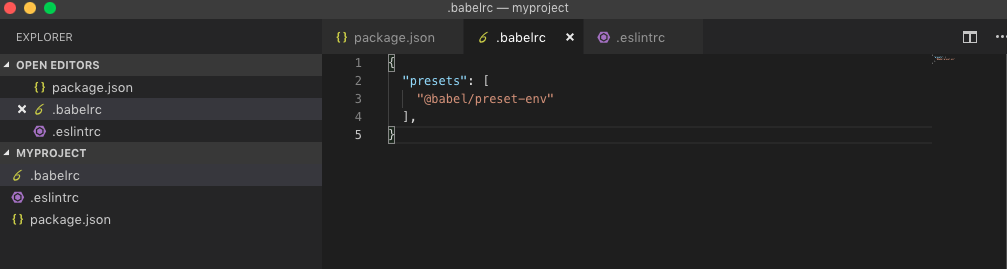
Description automatically generated



1. Set up a runtime configuration for the babel.   
   Create a ‘.babelrc’ file under the root folder and copy content from <https://bit.ly/2Xn2FY9> to the file.

A screenshot of a cell phone

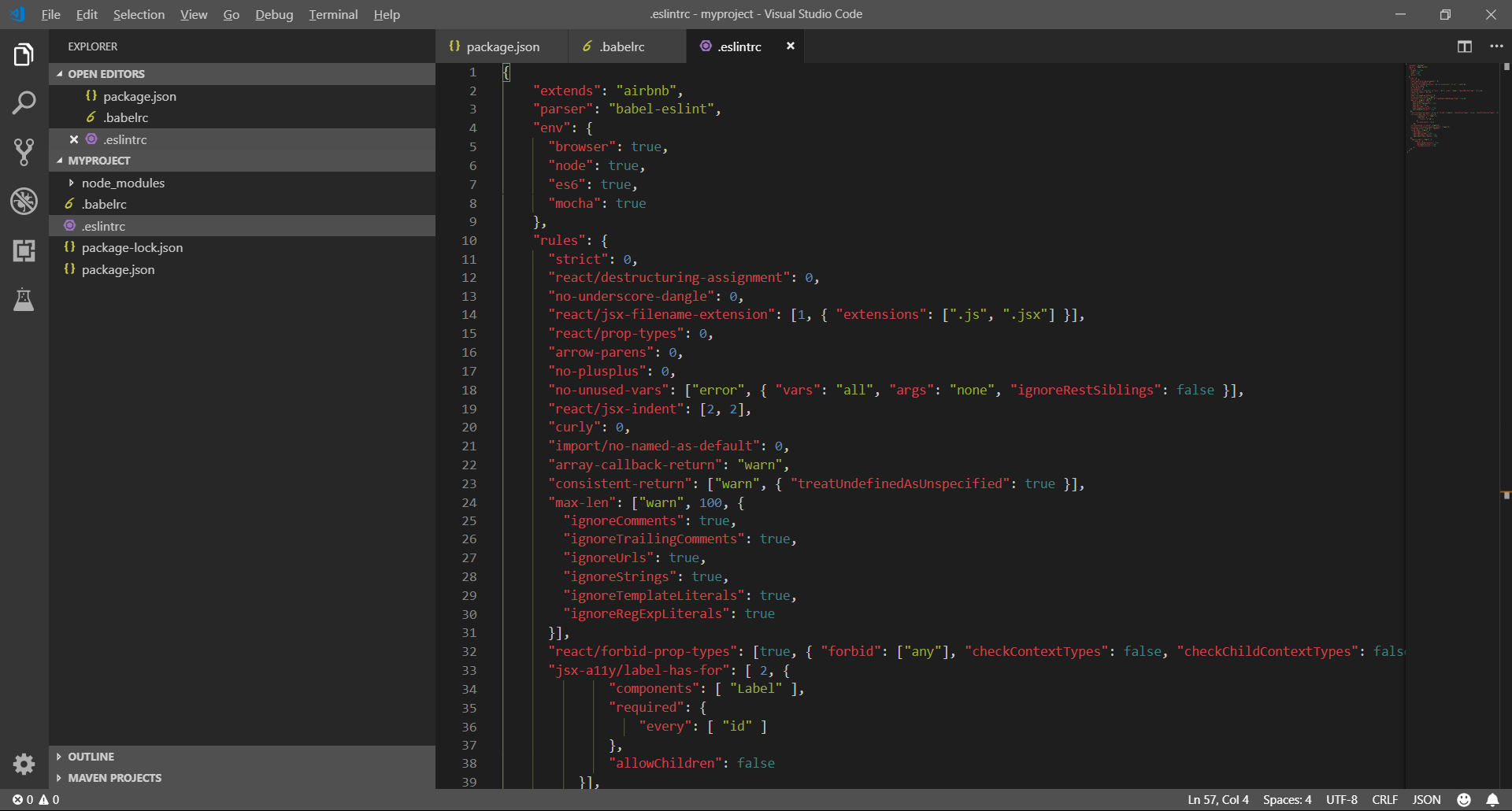
Description automatically generated



1. Set up a runtime configuration for the eslint.   
   Create a ‘.eslintrc’ file under the root folder and copy content from <https://bit.ly/3eh4mgV> to the file.

A screenshot of a social media post

Description automatically generated



1. Restart your IDE (quit and open VSCode) to enable both the babel and the eslint.

**Push your work to GitHub**

Open the terminal from the VSCode by hit the control + ~ key and type the following command:

Run the following commands to push your work to the GitHub repository:

>>> git add .

>>> git commit -m “Submission for Module 2”

>>> git push origin YOUR\_BRANCH\_NAME

**Note**: you should change the YOUR\_BRANCH\_NAME to your own branch name. It should be firstname-lastname (e.g. maria-gracia).

If you cannot remember, run the command “git status” to check