# Unit Testing Configuration

Guide for installing and setting up the needed libraries and frameworks to allow local testing of JavaScript code for the ["JavaScript Advanced" course @ SoftUni](https://softuni.bg/modules/76) in WebStorm and Visual Studio Code.

## Visual Studio Code Configuration

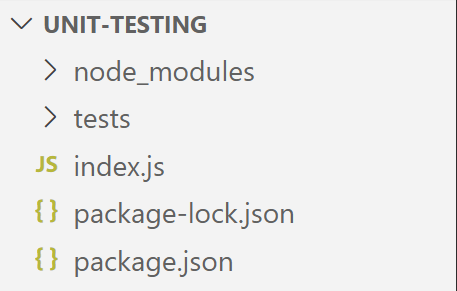
Open the **terminal** in Visual Studio Code and write the following **commands** to **install** **Mocha** and **Chai**:







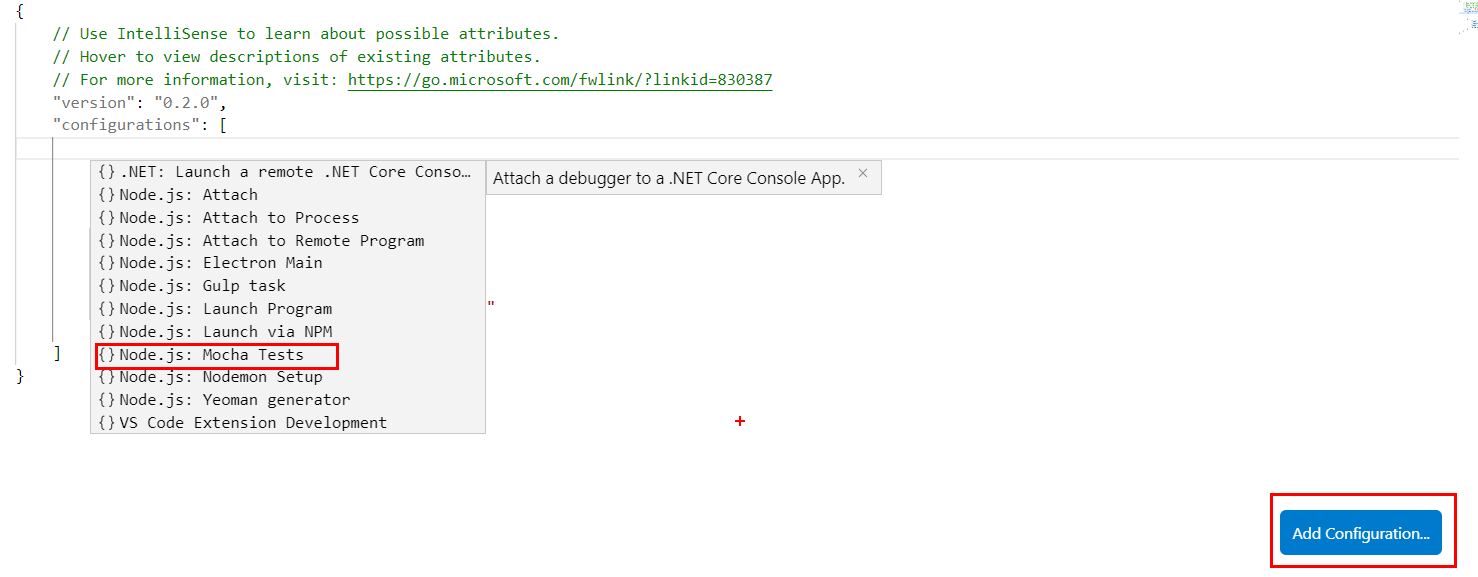
The structure of your *Unit Testing* folder should look something like this:

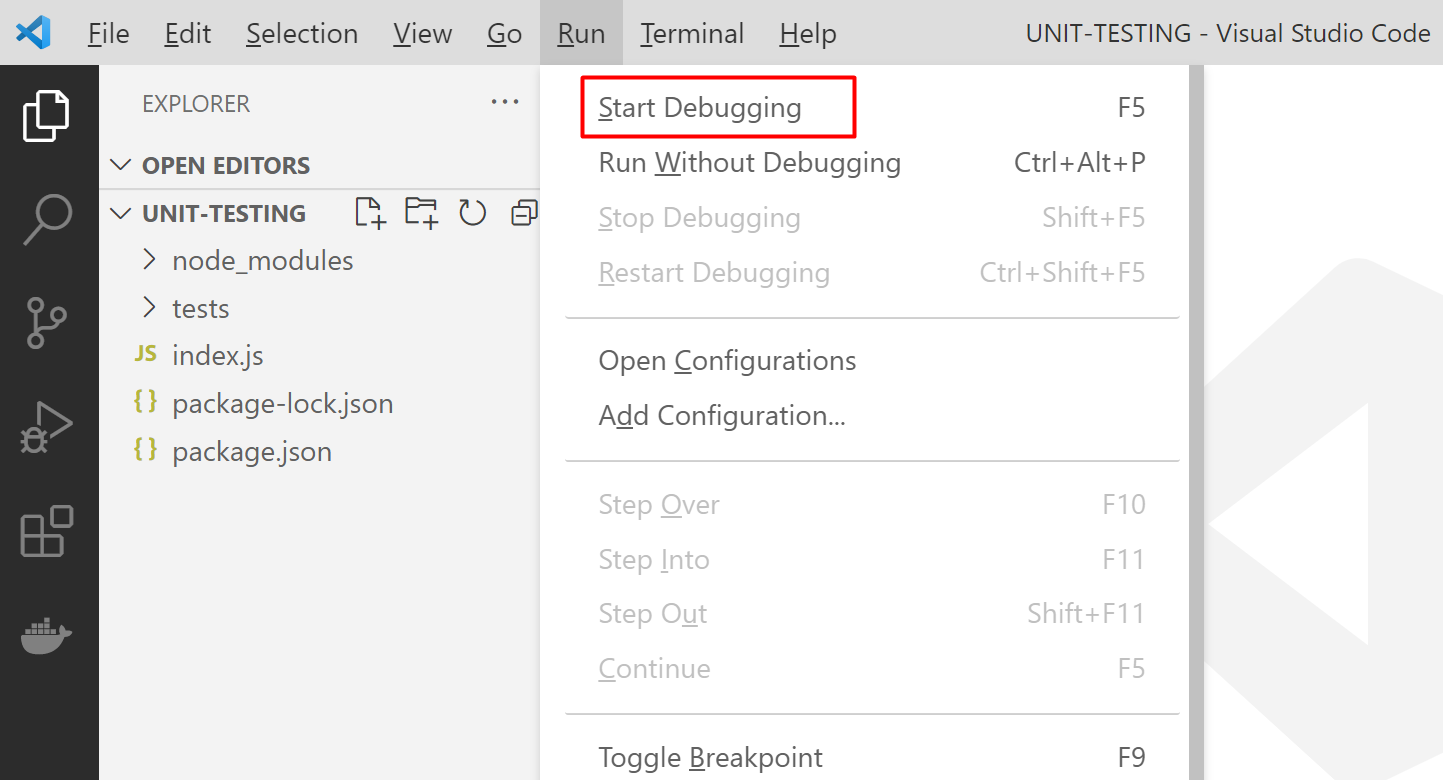


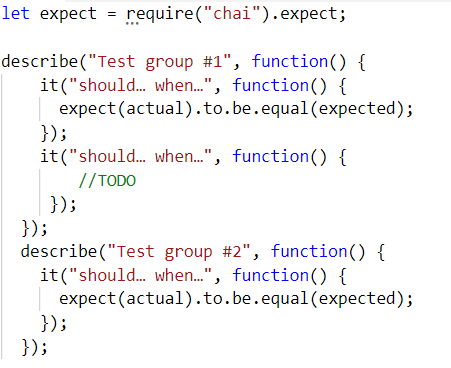
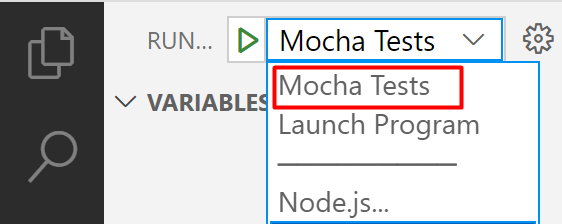
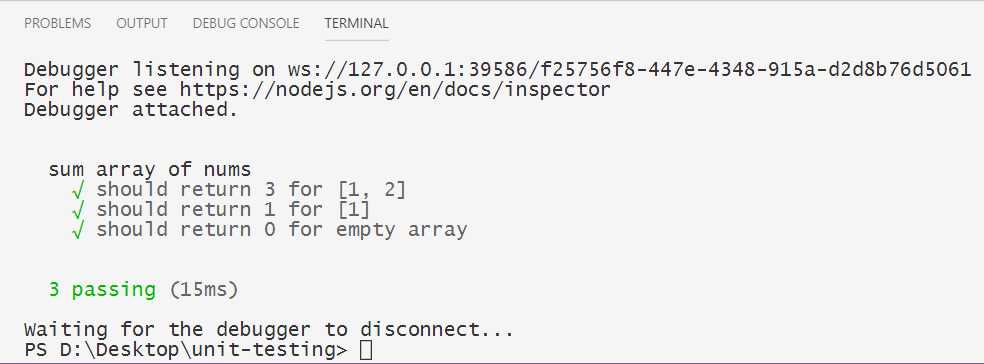
In order to be able to **run unit testing**, you have to do some **configurations**. Start by clicking on **Run** in the menu above and choose **Open Configurations**.

Картина, която съдържа текст, софтуер, Компютърна икона, номер

Описанието е генерирано автоматично

This will add a mocha configuration in **launch.json** and you should change the code to match the following:

Then click on **Run** and choose **Start Debugging**.

The following window will show up. Select **Mocha Tests** for **configuration**. Now you should be able to make unit tests in VS Code. ☺

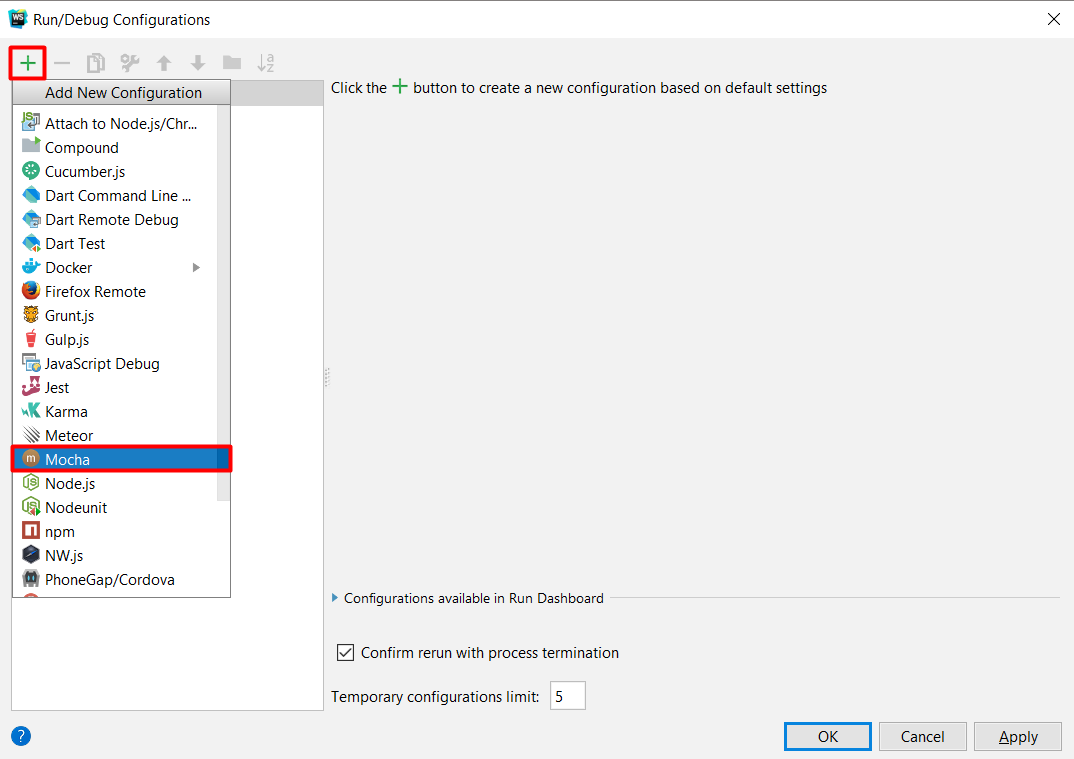
This is a **sample structure** of what **testing** **groups** and **classes** should look like. Don’t forget to **require chai** in the JavaScript file.

## WebStorm Configuration

Open the **terminal** in **WebStorm** and type in the following command:



From the menu above, select **Run** and **Edit Configurations**. In order to **add new configuration**, click on the green plus and select **Mocha** from the dropdown menu.



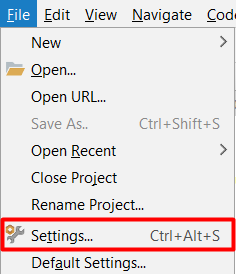
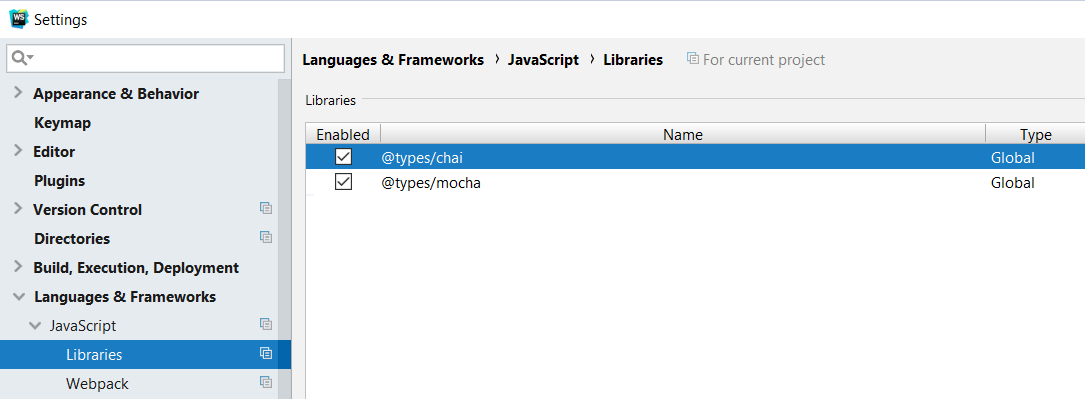
After that, you will have to choose a **name** for your configuration. A **good example** of one would be "**Mocha Tests**". For **Node interpreter**, choose the **directory** in which **node** has been **installed**. The **Working directory** should be your **current** working directory and for **Mocha package** you should select the directory in which Mocha has been installed.

Картина, която съдържа текст, екранна снимка, номер, софтуер

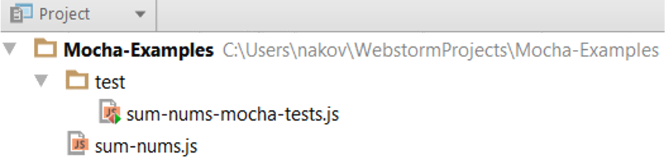
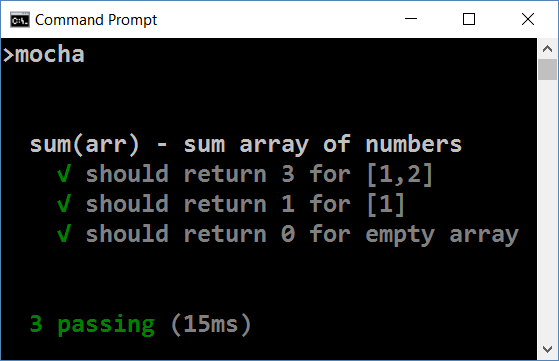
Описанието е генерирано автоматично

Now that you have set up the new configuration, it is time to get the "**auto complete**" and "**syntax checks**" working for Mocha and Chai. To do that, you must add them as **libraries** in WebStorm.

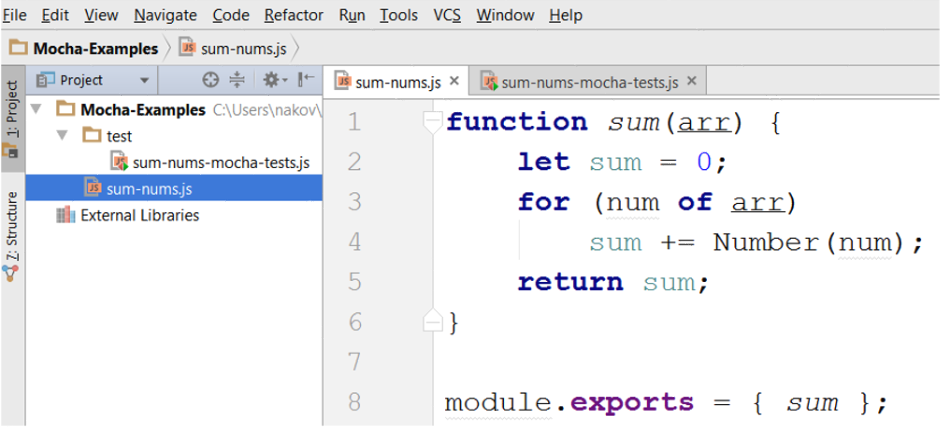
Go to **File** and select **Settings**. Then click on **Languages & Frameworks**, go to the **JavaScript** menu and select **Libraries**.

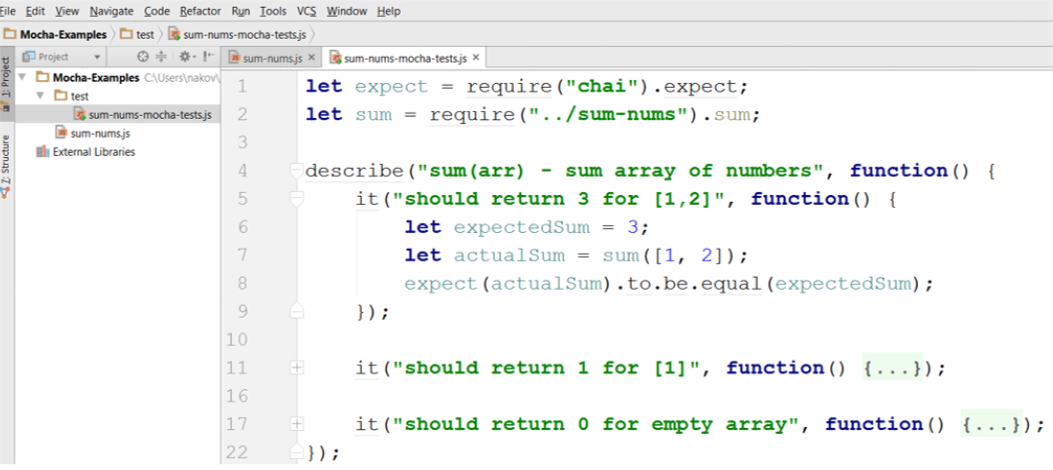
In order to run Mocha, you must create a **new folder** (for example, it could be called *test*) in your JS project. After that, you must **put your test code** in the folder you have just created. Provided you have named your folder *test*, your directory should look like this: *test/{test-group-name}.js*. Then run **Mocha** from the **console**.

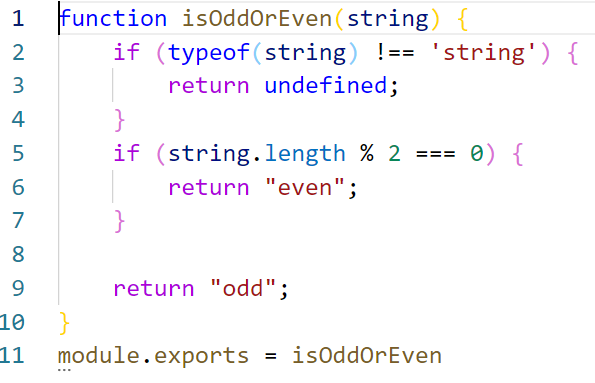
Here is what a sample project should look like with a source code to be tested:



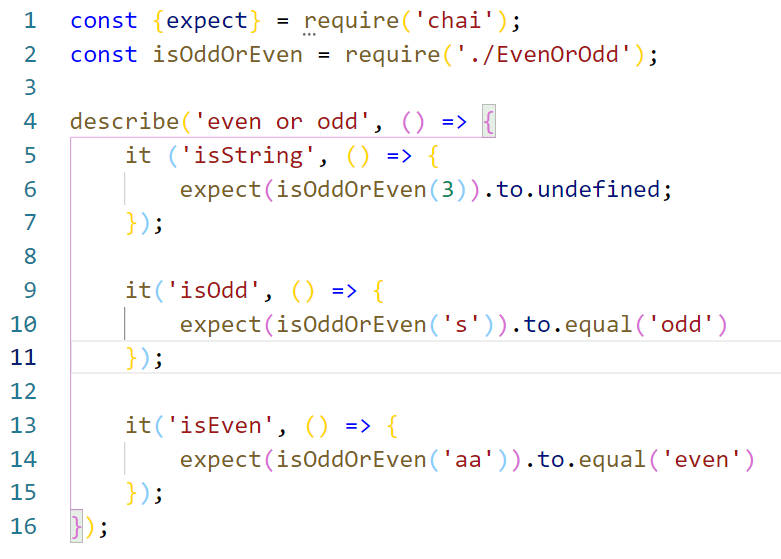
And here is a simple example of how testing of the function above should be done:



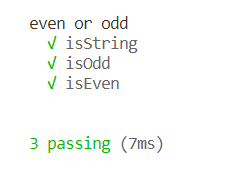
## Example



* Module.exports – A property that makes the code assigned to it public. Allows the assigned code to be used in other files.



* Require – Built-in methods. Loads the contest of the module into another file.



* Start the tests – Write in the terminal – **mocha {tests’ file name}.js**