

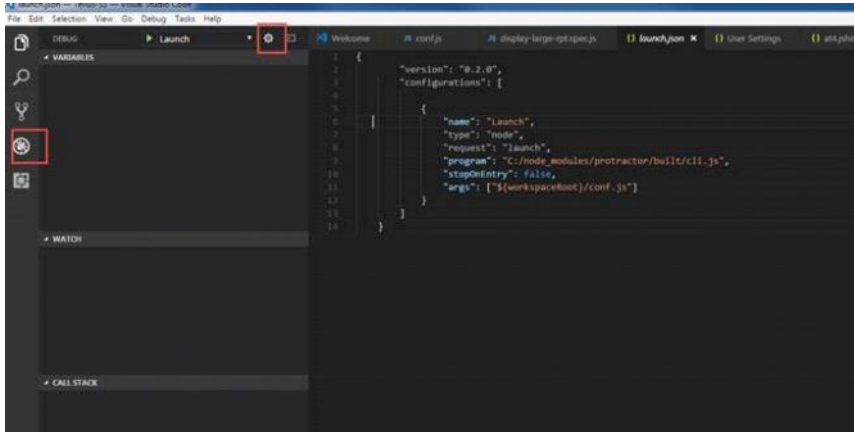
Getting Started with VS Code

Tuesday, December 05, 2017 10:56 PM

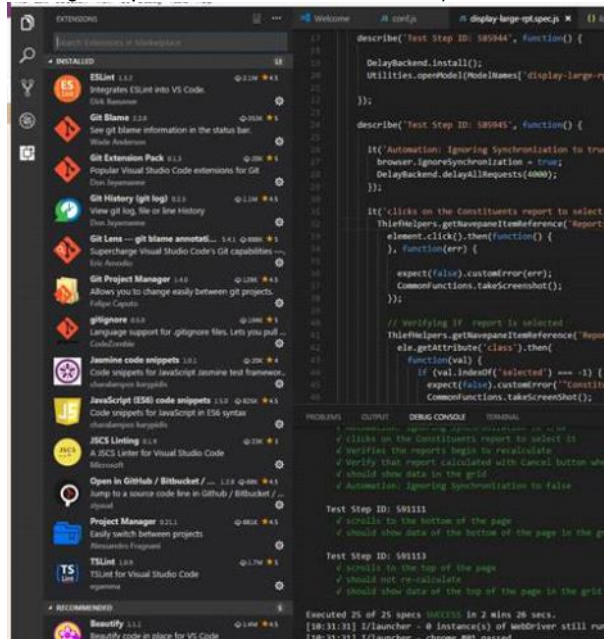
Here are my findings while setting the project in visual studio code.

- Install Visual studio code.
- Open your project in VSC editor.
- For the configuration settings to run our conf.js file: In Debug section click on the cogwheel button and edit the **launch.json**. It is similar like web storm configurations.

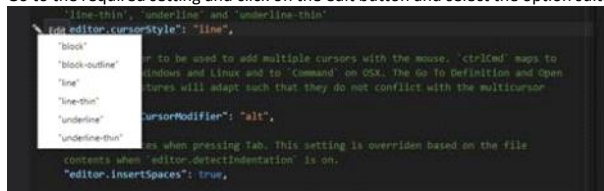
```
"name": "Launch",
"type": "node",
"request": "launch",
"program": "C:/node_modules/protractor/built/cli.js",
"stopOnEntry": false,
"args": ["${workspaceRoot}/conf.js"]
```



- Installing the required modules: From the extensions tab you can install all the required modules. (Adding the screenshot that includes the modules that I've installed).



- Setting User settings: Click on the cog Wheel from down left and select settings or select settings from File menu to set the User settings. Here you can edit the settings as per our project requirements.
- Go to the required setting and click on the edit button and select the option suitable for the project.



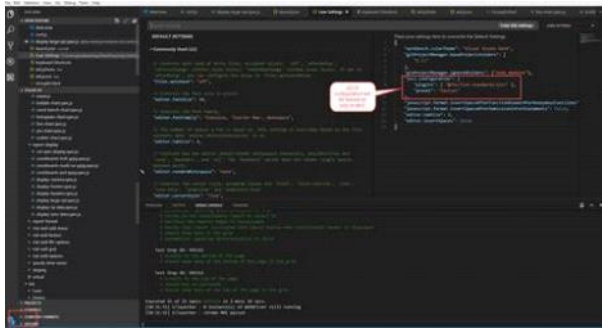
- Following are the changes that I've added

```
"gitProjectManager.baseProjectsFolders": [
  "C:\\",
],
"gitProjectManager.ignoredFolders": ["node_modules"],
"jscs.configuration": {
```

```

"plugins": [ "@feds/lint-standards/jscs" ],
"preset": "factset"
},
"javascript.format.insertSpaceAfterFunctionKeywordForAnonymousFunctions": false,
"javascript.format.insertSpaceAfterSemicolonInForStatements": false,
"editor.tabSize": 2,
"editor.insertSpaces": false

```



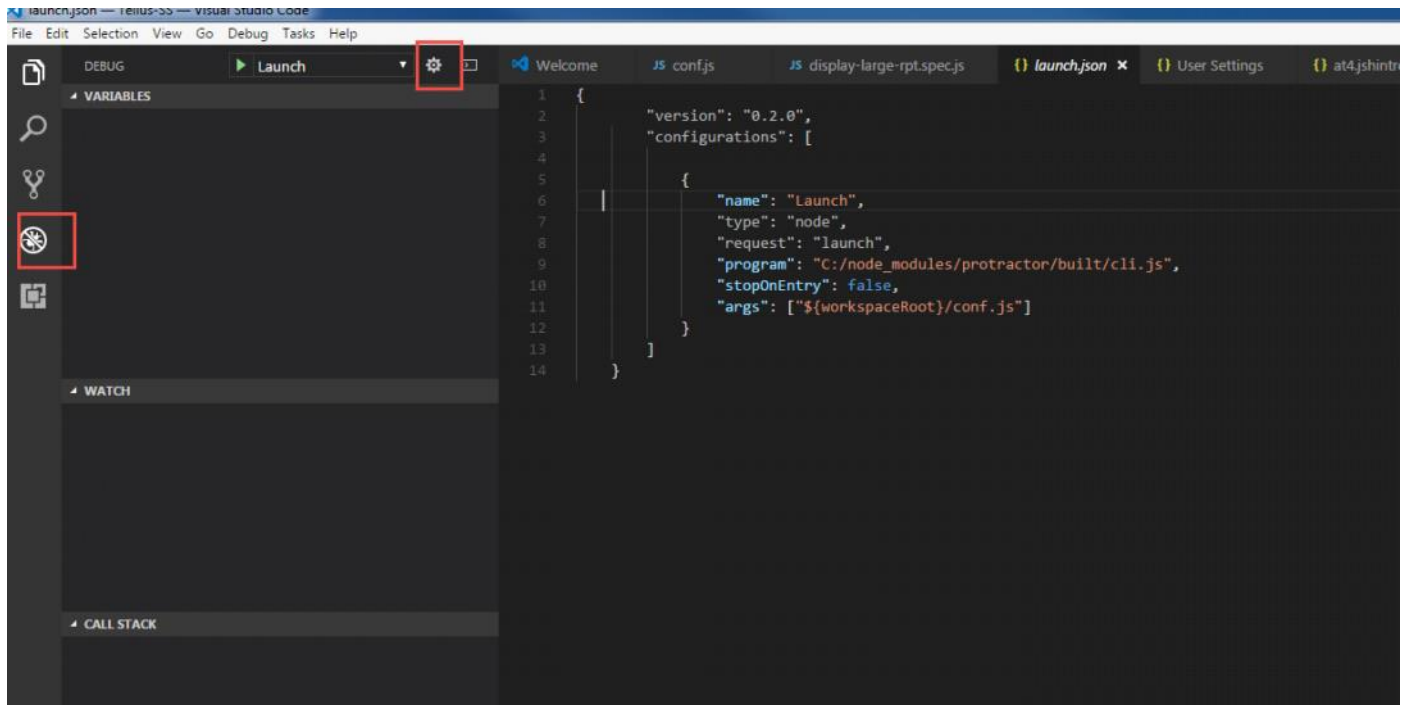
- To run your project by clicking on the **Run** button in DEBUG section. I observed that there is no separate buttons for run and debug.
- To fix all jscs errors in your file go to the file and **Alt+Shift+F** or right click on the file and select **Format Document** option.

Getting started with VS code editor - Protractor

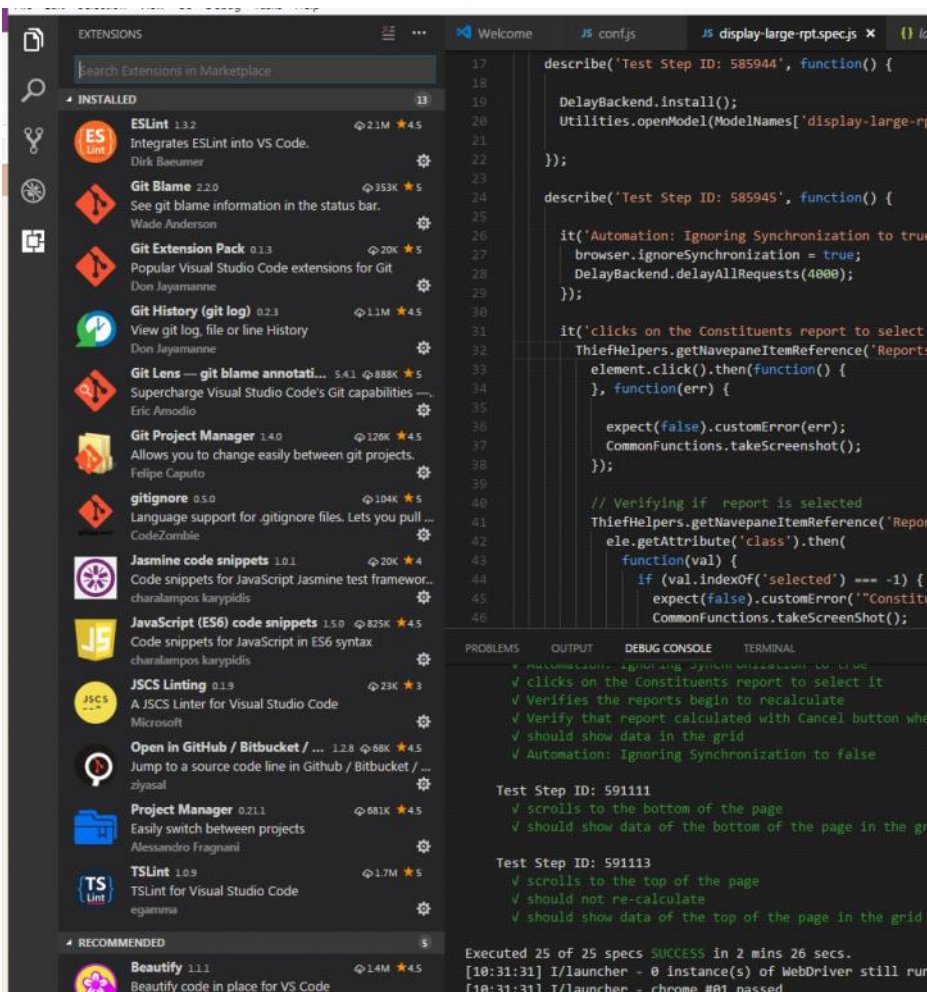
Thursday, October 12, 2017 4:52 PM

- Install Visual studio code.
- Open your project in VSC editor.
- **For the configuration settings to run our conf.js file:** In Debug section click on the cogwheel button and (select node.js) edit the **launch.json**. It is similar like web storm configurations.

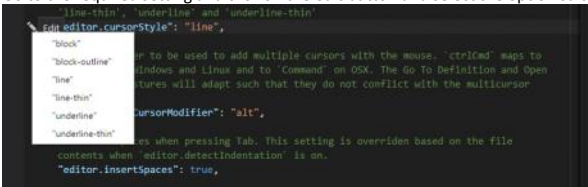
```
"name": "Launch",  
"type": "node",  
"request": "launch",  
"program": "C:/node_modules/protractor/built/cli.js",  
"stopOnEntry": false,  
"args": ["${workspaceRoot}/conf.js"]
```



- **Installing the required modules:** From the extensions tab you can install all the required modules. (Adding the screenshot that includes the modules that I've installed).

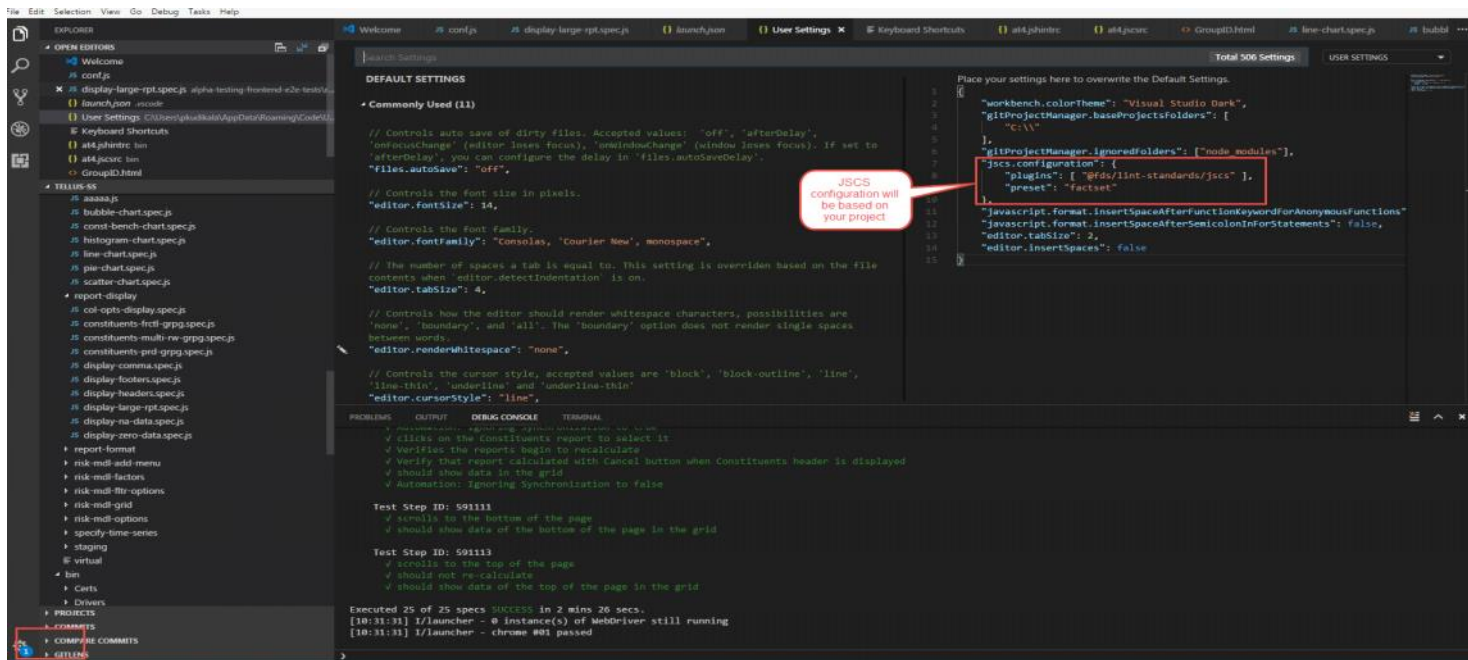


- **Setting User settings:** Click on the cog Wheel from down left and select settings or select settings from File menu to set the User settings. Here you can edit the settings as per our project requirements.
- Go to the required setting and click on the edit button and select the option suitable for the project.



- Following are the changes that I've added

```
"gitProjectManager.baseProjectsFolders": [
  "C:\\",
],
"gitProjectManager.ignoredFolders": ["node_modules"],
"jscs.configuration": {
  "plugins": [ "@feds/lint-standards/jscs" ],
  "preset": "factset"
},
"javascript.format.insertSpaceAfterFunctionKeywordForAnonymousFunctions": false,
"javascript.format.insertSpaceAfterSemicolonInForStatements": false,
"editor.tabSize": 2,
"editor.insertSpaces": false
```

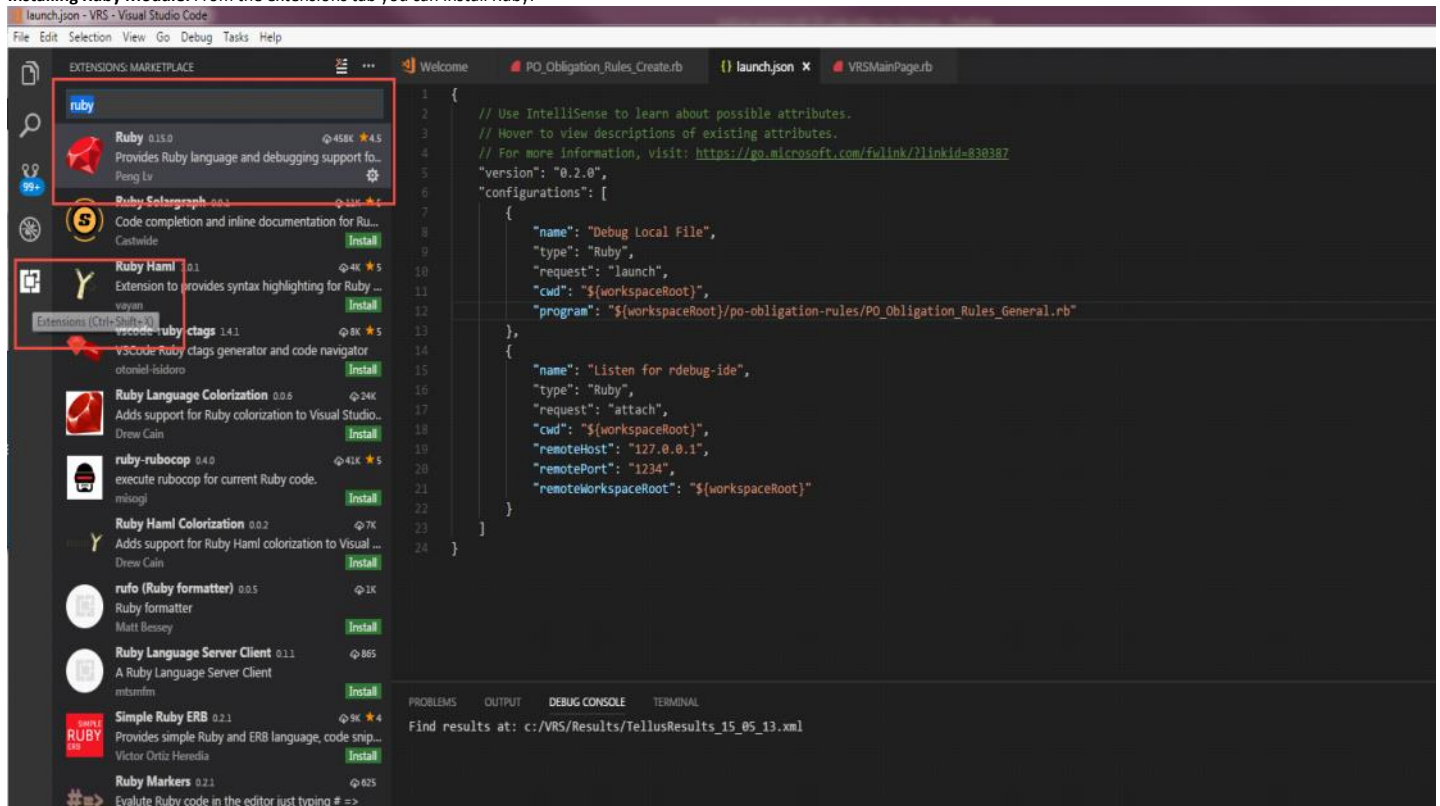


- To run the script normally select Run without debugging option from Debug menu or **CTRL+F5**.
- For Debug: Click on **Run** button in DEBUG section or **F5**.
- To fix all jscs errors in your file go to the file and **Alt+Shift+F** or right click on the file and select **Format Document** option.

Getting started with VS code editor for Selenium

Thursday, October 12, 2017 4:52 PM

- Install Visual studio code.
- Open your project in VSC editor.
- **Installing Ruby Module:** From the extensions tab you can install Ruby.



- Configuration Setup for running the file:

In Debug section click on the cogwheel button and (select Ruby) edit the **launch.json**

```
// Use IntelliSense to learn about possible attributes.
// Hover to view descriptions of existing attributes.
// For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
"version": "0.2.0",
"configurations": [
  {
    "name": "Debug Local File",
    "type": "Ruby",
    "request": "launch",
    "cwd": "${workspaceRoot}",
    "showDebuggerOutput": true,
    "program": "${workspaceRoot}/po-obligation-rules/PO_Obligation_Rules_General.rb" || "${file}"
  },
  {
    "name": "Listen for rdebug-ide",
    "type": "Ruby",
    "request": "attach",
    "cwd": "${workspaceRoot}",
    "remoteHost": "127.0.0.1",
    "remotePort": "1234",
    "remoteWorkspaceRoot": "${workspaceRoot}"
  }
]
```

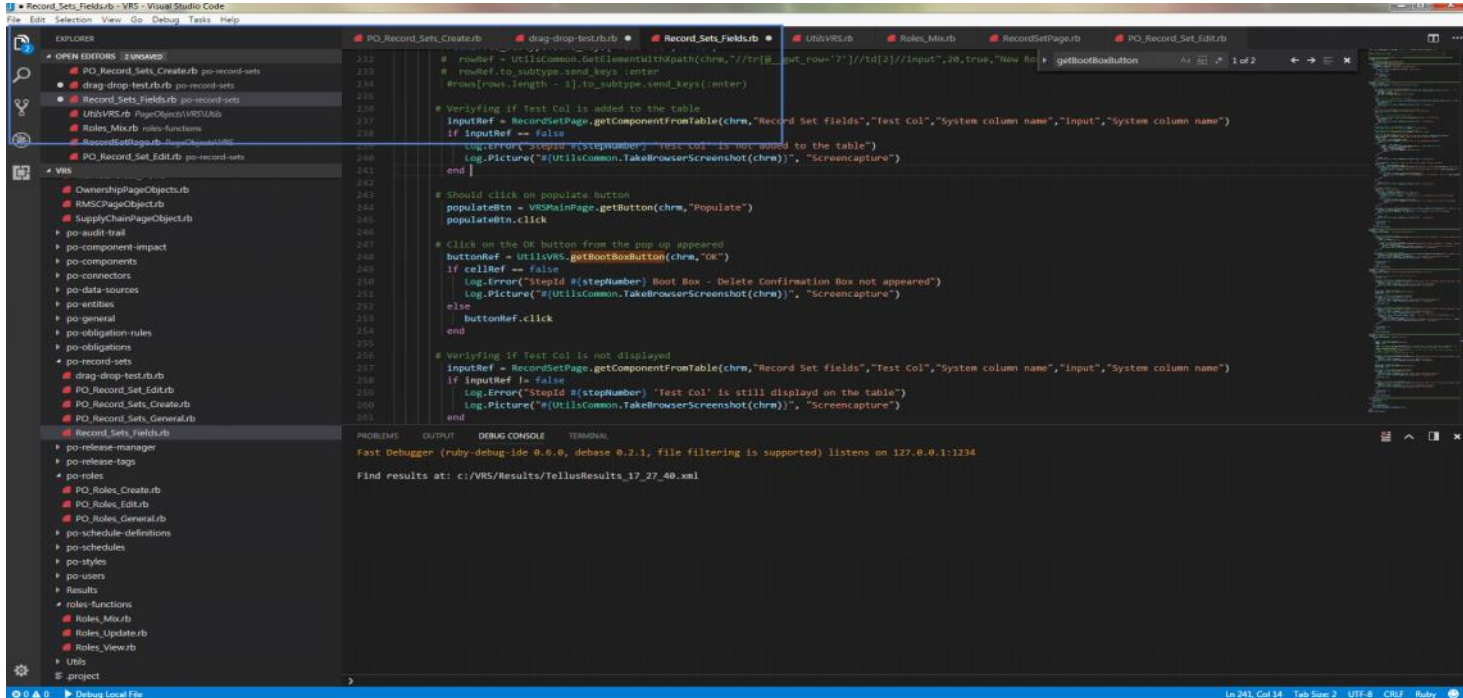
- **User Settings Explanation:**

- **"cwd"**: Current Working directory
- **"program"**: The location of the script to run. We can give the location of the script or we can also specify as "\${file}" to run the currently opened file.
- **"showDebuggerOutput"**: This value should be declared as true if you want to debug the script.
- To run your project by clicking on the **Run** button in DEBUG section. I observed that there is no separate buttons for run and debug. Shortcut for triggering the run is **F5**.
- To install the keyboard shortcuts same as WebStorm you can install **IntelliJ IDEA Key Bindings for Visual Studio Code** from the Extensions tab.
- To get the automatic code completion. Ex. Type begin and then do **Ctrl+Space**, select begin then it automatically populate the code like

```
begin
  rescue => exception
end
```

- **Useful Extensions:** Note: Once you install any extension you should restart sometimes in order to initialize the installed extension sometimes.

- 1) Ruby Solargraph: To get the ruby methods extensions. Ctrl+Space to get the suggestions.
- It will show the Un Saved Files:



If you observe the above screenshot on the top left for the tab File Explorer it is showing as 2 it means we have 2 Unsaved Files and in the Open Editors tab to the right of File Explorer tab we have 2 files called drag-drop-tested.rb, Record_Sets_Fields.rb with dot left to its file name so those are the files that are unsaved

- When we migrate to GIT it will be easy for us to track the Un Staged, Modified files. It will directly show as U - Un staged, M- Modified to the right of the file names.
- It will directly identify the merge conflicts and show as diff of 2 version in the Editor when we want to stage the file.
- To Add Custom Snippets in VSCode. Please navigate to

- 1) **File > Preferences > User Snippets**
- 2) Select your language from the new window/palette that appeared
- 3) It will create language.json file. Ex: ruby.json or javascript.json
- 4) Add your Snippet Code. Attached a snippet code



ruby

Custom Snippets

Saturday, December 09, 2017 4:08 PM

Subject	Custom Snippets
From	Santhoshi Chepuru
To	Ramya Parthasarathy; Shiva Kumar Macharla; Remya Viswanathan; Vasu Nakka
Cc	Deekshith kumar Lingoju; Bhupender Agarwal
Sent	Thursday, December 07, 2017 4:52 PM

To Add Custom Snippets in VSCode. Please navigate to

1. **File > Preferences > User Snippets**
2. Select your language from the new window/palette that appeared
3. It will create language.json file. Ex: ruby.json or javascript.json
4. Add you Snippet Code.

I created one sample snippet file for ruby and javascript, Example Ruby looks like:

```
{
  "Test Step Logging": {
    "prefix": "TestStepLog",
    "body": [
      "stepNumber = '${1:stepNumber}'",
      "  Log.TestStep(stepNumber, \"${2:Description}\")",
      "  begin",
      "",
      "  rescue Exception => e",
      "    Log.Error e",
      "    return false",
      "  ensure",
      "    Log.CloseTestStep()",
      "  end"
    ],
    "description": "Test Step Logging"
  },
  "Error Logging": {
    "prefix": "Log.Error",
    "body": [
      "Log.Error(\"#{stepNumber}: ${1:Error Description}\")",
      "Log.Picture(\"#{UtilsCommon.TakeBrowserScreenshot(chrm)}\", \"Screencapture\
\\")",
      ""
    ],
    "description": "Error Logging"
  },
  "Test Case Logging": {
    "prefix": "TestCaseLog",
    "body": [
```



```

    "require File.expand_path(File.dirname(__FILE__) +
'../PageObjects/VRS/AllPageObjects.rb')",
    "#=====
=====#",
    "# NAME: ${1:Test Case
Name}
    "#=====
=====#",
    "",
    "##### TestCase
#####",
    "def ${2:Test Case Name}",
    "  begin",
    "    Log.TestScript(\"${3:Test Case Name}\")",
    "    Log.TestCase(${4:Test Case ID}, \"${5:Test Case Name}\")",
    "  ",
    "#####
#####",
    "  rescue Exception => e",
    "    Log.Error e",
    "    return false",
    "  ensure",
    "    Log.CloseTestCase()",
    "    Log.CloseTestScript()",
    "    UtilsCommon.CloseWorkStationAndBrowser(chrm)",
    "  end",
    "end",
    "${6:Test Case Name}"
  ],
  "description": "Test Case Logging"
}
}

```

So, in your test case if you type **TestStepLog** and then do Ctrl+Space or some times by default you can choose from the suggestions and can use the create snippet in your code. Reference

URL: <https://code.visualstudio.com/docs/editor/userdefinedsnippets>

I will be continuously update my findings in the One Note Created for the VSCode. Please do check them once in a week and please add your findings also so that it can be circulated to the team.

FYI: The way to create a snippet is common for all the languages. You can add your own custom snippets now.

Regards,
Santhoshi Chepuru
Automation Analyst II, QA Automation
FactSet Systems India Pvt. Ltd.
csanthoshi@factset.com
www.factset.com



FACTSET > SEE THE ADVANTAGE

FINANCIAL DATA | ANALYTICS | TECHNOLOGY | SERVICES