**Custom Extension In Visual Studio Code**

# OVERVIEW

Visual Studio Code editor is a lightweight source code editor (***read as IDE***) developed By Microsoft Inc . VS Code editor runs on your machine's desktop (Windows, macOS and Linux).

This editor has built-in support for Javascript, Typescript,Node.js and has many extensions (plug-ins) to support other languages such as C++, C#, Java, Python, PHP, Go and many more.

[Introduction of Visual Studio Code.](https://code.visualstudio.com/docs/getstarted/introvideos)

[Download Visual Studio Code Editor.](https://code.visualstudio.com/Download)

[Setting up Visual Studio Code.](https://code.visualstudio.com/docs/setup/setup-overview)

# 

# Why to create an Extension (Plug-in)

1. To Customize Visual Studio Code Editor as per custom needs.
2. To plug custom feature/functionality directly into the Visual Studio Code UI and contribute functionality through the same APIs used by VS code.

# 

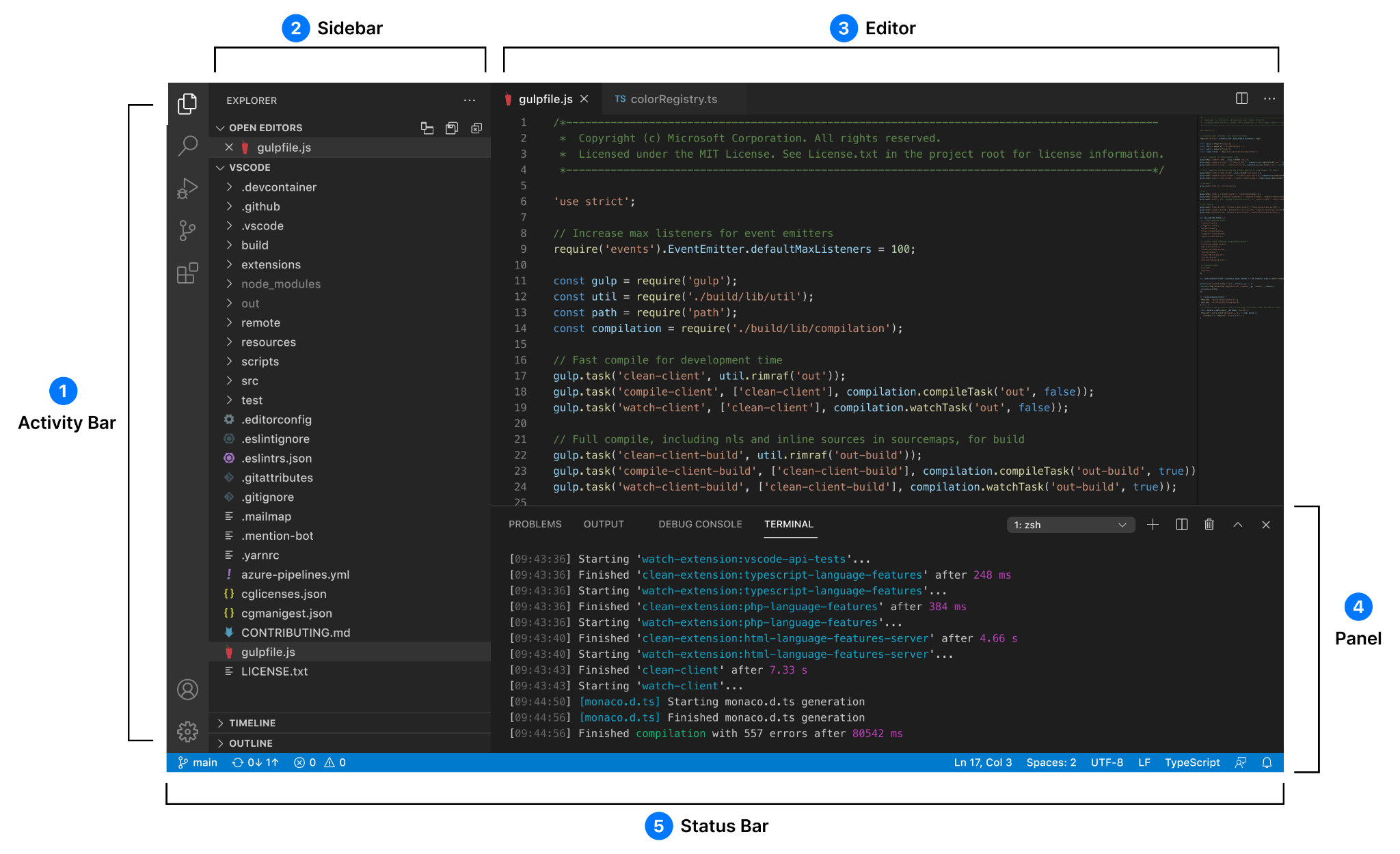
# Visual Code Editor Architecture

VS Code UI architecture comprises of 2 types of elements :

1. **Containers :**- The Outer layer of Visual Studio Code Editor
2. **Items :**- The Inner layer of Visual Studio Code Editor and, are placed inside of various containers

# 

# *Containers :-*



1. **Activity Bar** : Lets you to switch between views *(for example - Explorer, Search, Source Control etc Or, Any other Third-party extension)*.
2. **Sidebar :** Contains different views *(like Explorer, Extension Marketplace etc)* to assist users while working on their project.
3. **Editor :** Main area to edit files. Users can open as many editors as they like side by side vertically and horizontally.
4. **Panels :** Display different panels*(Problems,Output,Debug etc or an integrated terminal)* below the editor region. Panels can also be moved to the right for more vertical space.
5. **Status Bar :** Displays information about the opened projects and files edited by the user.

# *Items :-*

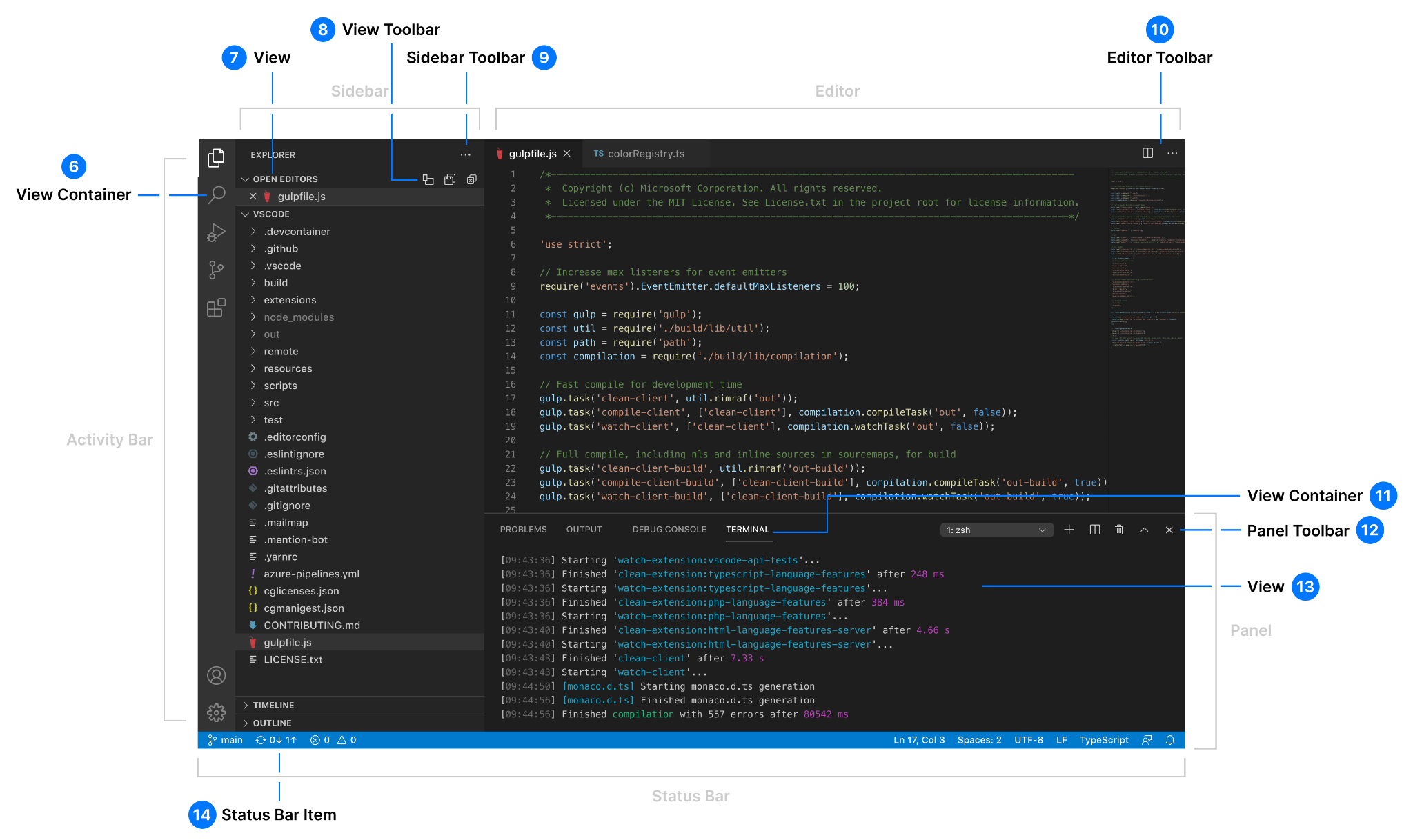
1. **View Containers :** Part of Activity Bar. Each container will be presented by a unique icon. A View Container contains a list of views that are displayed in the Activity Bar or Panel along with the built-in View Containers.  
   For example: Explorer, Source Control etc are view containers in VS Code Editor.
2. **View :** Containers of content that can appear in the Sidebar or Panel.   
     
   Features of View are :-

**\*\*** Views can contain Tree View (or WebView)

**\*\*** Custom Views can have custom display view actions.

**\*\*** Views can also be placed/plugged into other/existing view containers *(like File Explorer and SCM)*.

**\*\*** Views can also be added to any view container in the panel or sidebar Or, In their own Custom View container.



1. **View Toolbar**
2. **Sidebar Toolbar**
3. Editor Toolbar
4. View Container
5. Panel Toolbar
6. View
7. Status Bar Item

# Let’s Build Our First VS Code Extension *Prerequisites :-*

1. [Node.js](https://nodejs.org/en/) must be installed on your machine.
2. [Git](https://git-scm.com/) installed as we will use Git to manage our example.

# Step 1 - **Yeoman and Generator**

**\*\*** Install [Yeoman](https://yeoman.io/) (A Scaffolding tool for modern WebApps) ,  
**\*\*** [VS Code Extension Generator](https://www.npmjs.com/package/generator-code) (Readily available templates to develop extensions using JavaScript or TypeScript).

| npm install -g yo generator-code  Step 2 - **Run the generator**  yo code |
| --- |

Step 3 - **Choose/provide choices for the type of extension required to be developed :**

For Example :

? What type of extension do you want to create? New Extension (TypeScript)

? What's the name of your extension? MyFirstCustomExtension

*### Press <Enter> to choose default for all options below Or type your choices for each options and Press Enter ###*

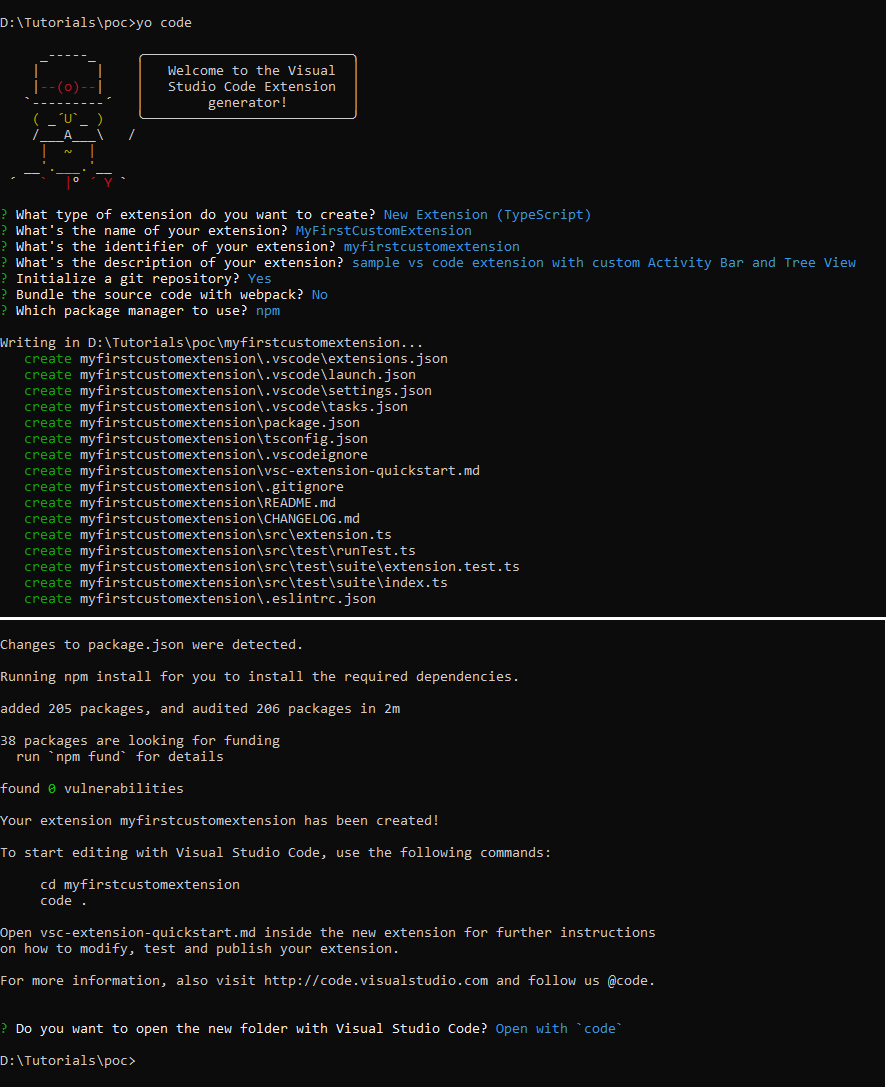
? What's the identifier of your extension? myfirstcustomextension

? What's the description of your extension? sample vs code extension

? Initialize a git repository? Yes

? Bundle the source code with webpack? No

? Which package manager to use? Npm



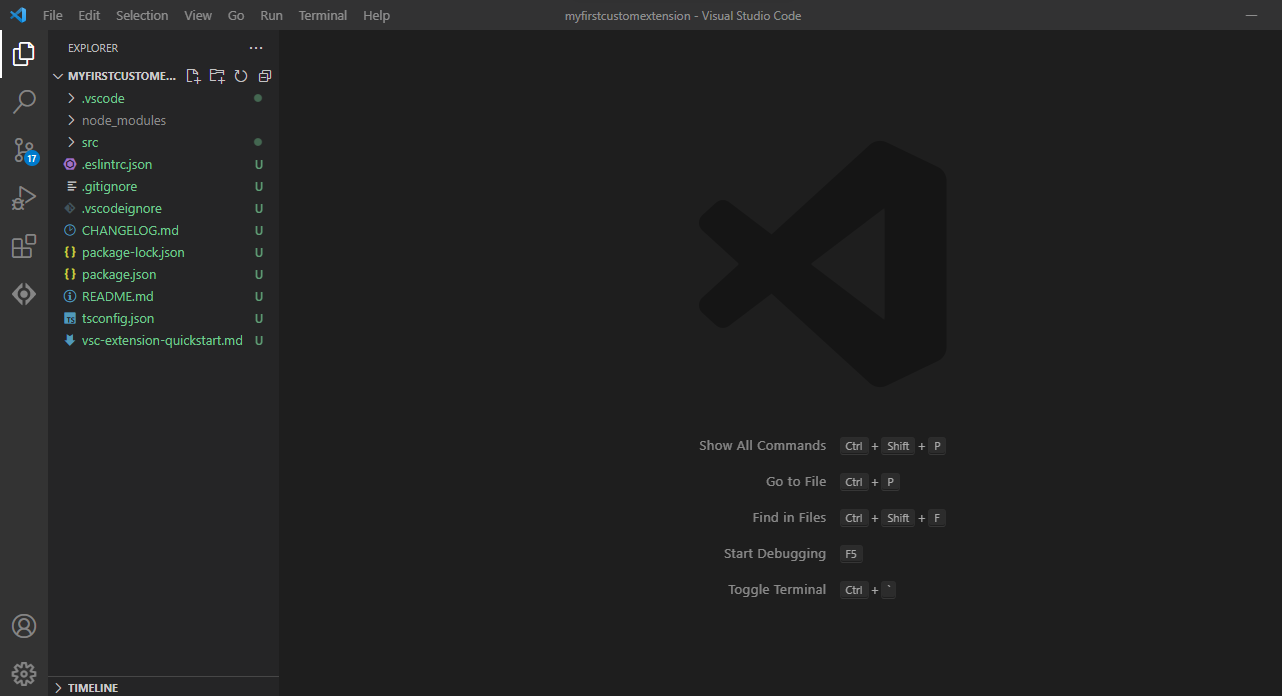
**Note:**

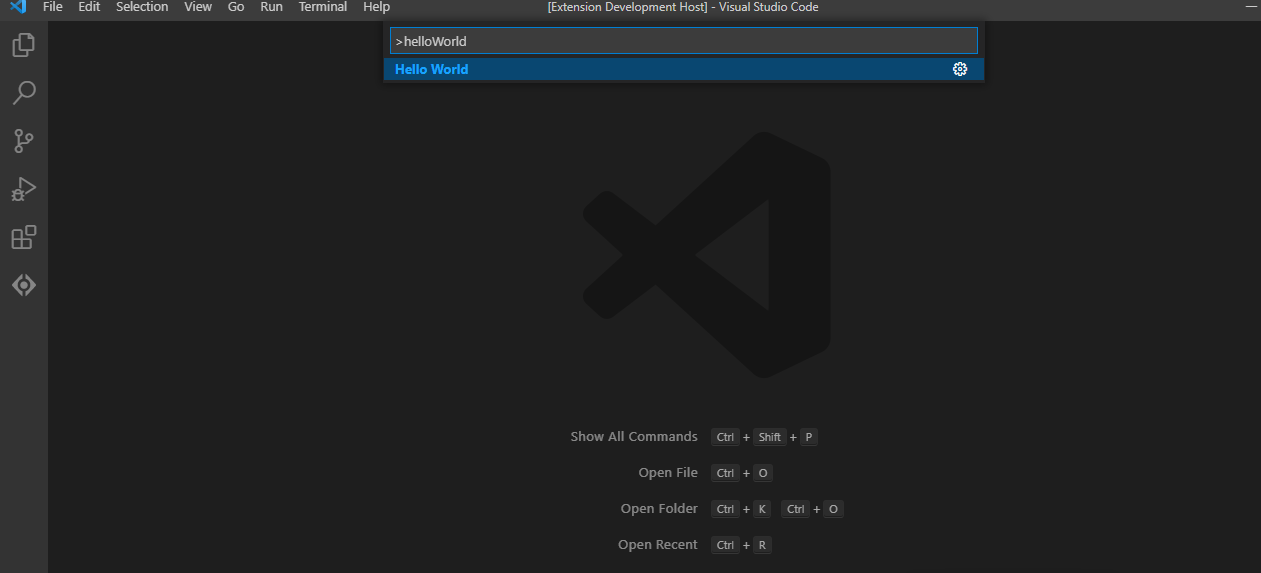
* Developers can either select 'TypeScript'*(Object-oriented Programming language) or 'JavaScript'(Scripting Language)* for extension development.
* Developers can either choose *'npm'* or *'yarn'* as Node Package Manager.
* Select Y/Yes Or N/No to initialize/skip your Git repository *(hidden directory .git will be created that stores all refs and objects Git uses to maintain project history)*.

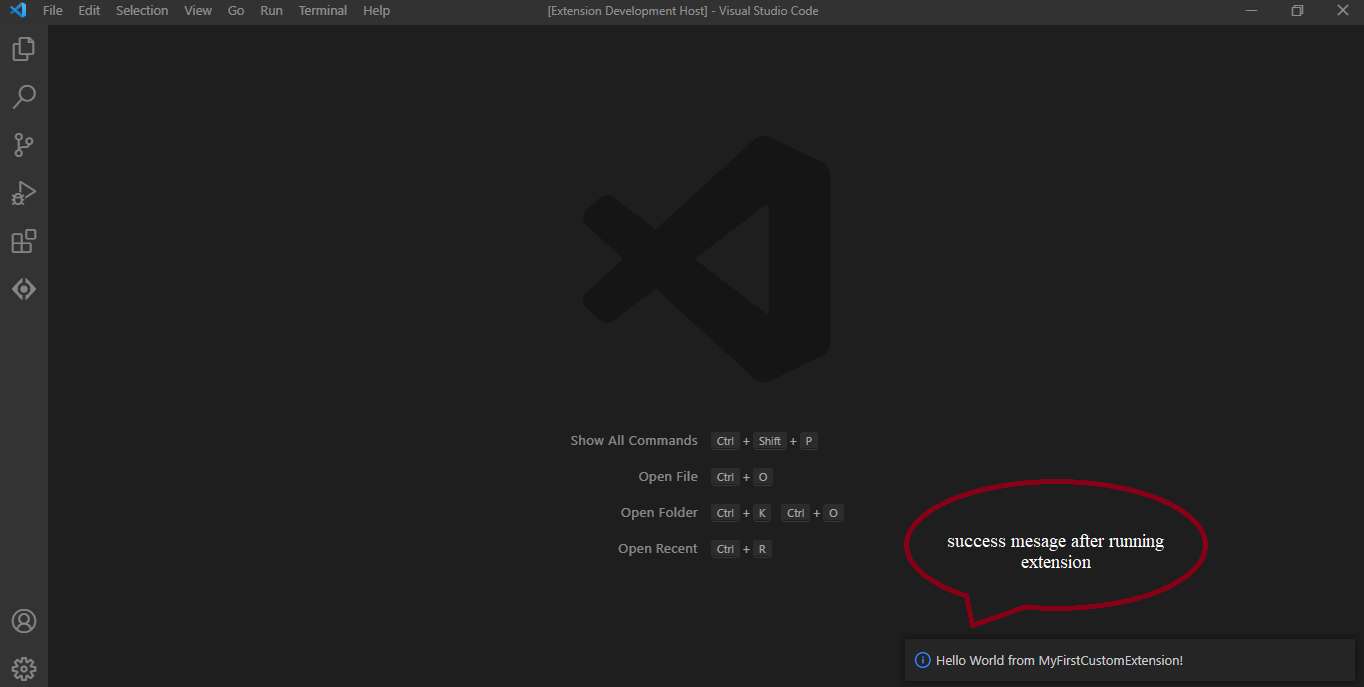
**[Git must be installed in the user's machine].**

Step 4 - **Run/Debug the Extension.**

1. Inside VS Code Editor press F5 to compile and launch the new Extension Development Host window.
2. Open VS Code Editor's Command Palette [Ctrl+Shift+P].
3. Run the 'helloWorld' command.

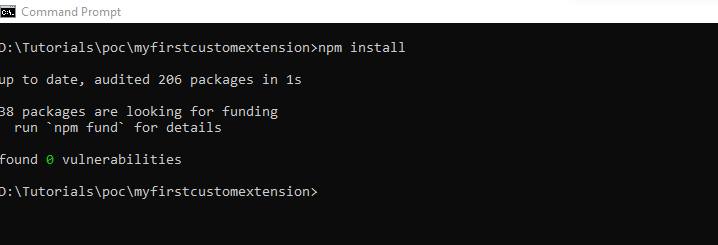


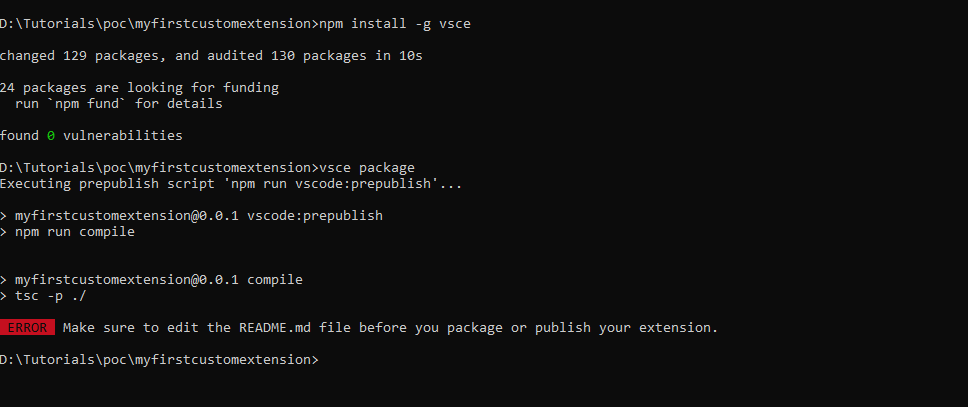




Step 5 - [**Packaging the extension**](https://code.visualstudio.com/api/working-with-extensions/publishing-extension)**.**

1. Run npm install to download dependencies through Node Package Manager.  
   npm install
2. Install *'****vsce****' (short for Visual Studio Code Extensions)* command-line tool for packaging, publishing and managing vs code extensions.  
   npm install -g vsce  
   vsce package

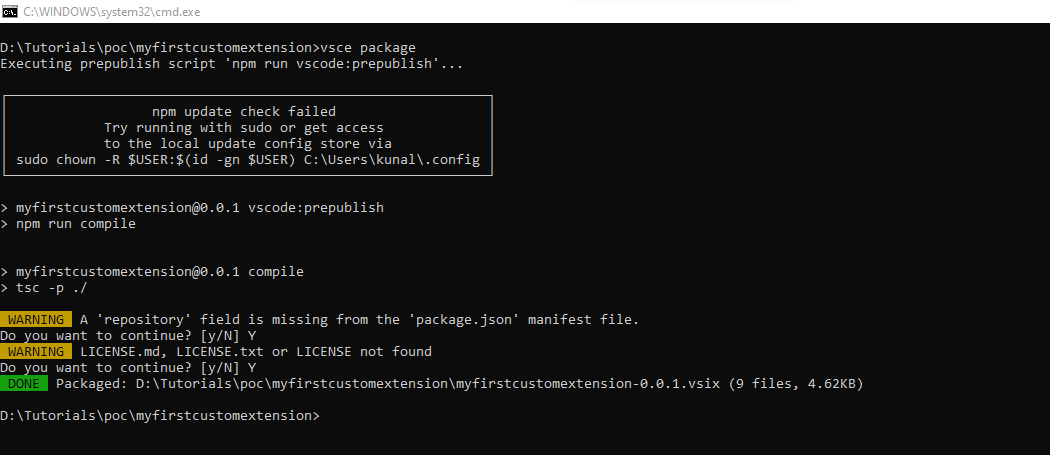




Step 6 - Error (*shown above image*) **while packaging extension.**

For now, just update the README.md file with these lines.

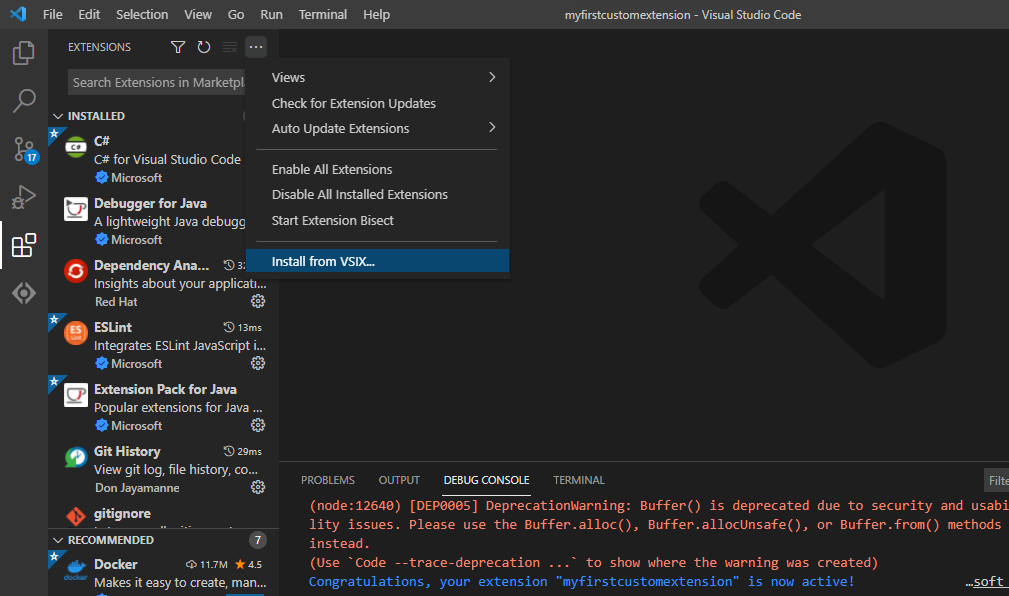
# myfirstcustomextension README  
Sample extension

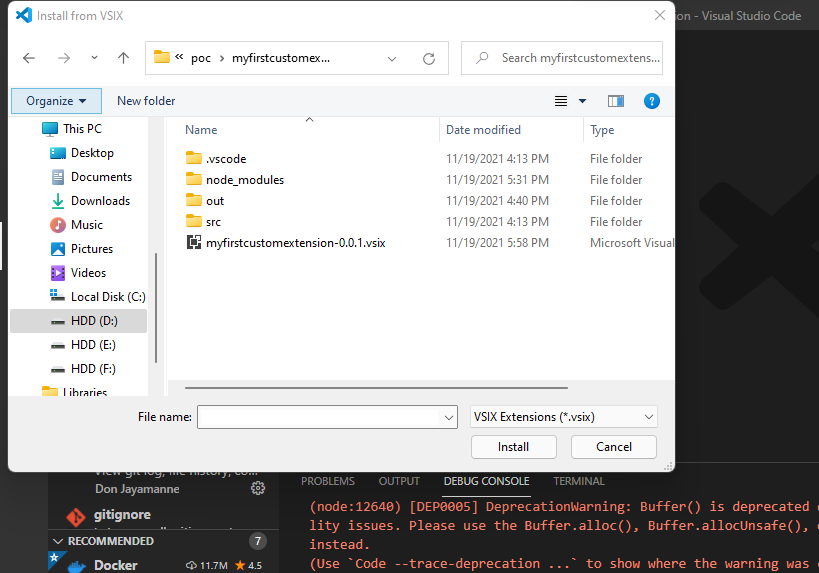


Inside your extension/project folder , file ***'myfirstcustomextension-0.0.1.vsix'*** will be created.  
This \*.vsix file can be distributed within team members to install VS Code extension/debug from local environment.

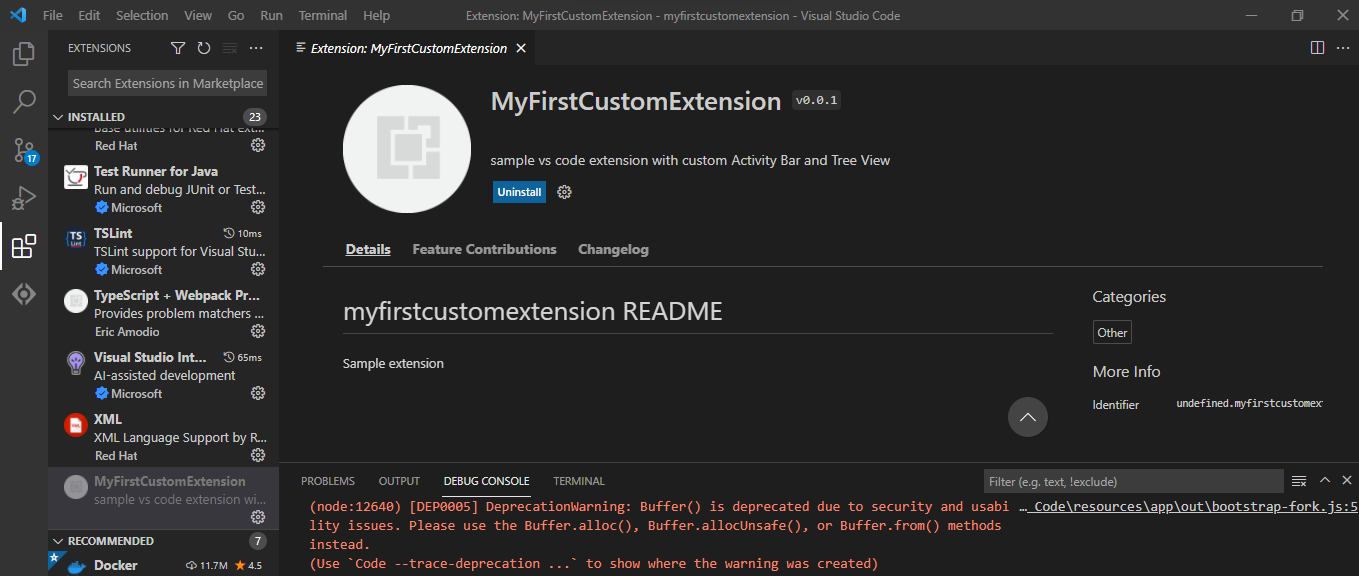
Step 7 - **Install newly package extension inside VS Code Editor**

1. Select *'Extensions (Ctrl+Shift+X)'* view from Activity Bar in VS Code editor.
2. Click on 3 dots (...) on the right side of the Extensions view.   
   Select Option - *‘Install from VSIX’.*Select *'myfirstcustomextension-0.0.1.vsix'* file.  
   Select *'Install'*
3. Under the *'Installed'* sub-view scroll down to look for the name of newly installed Extension *'MyFirstCustomExtension'*





1. Run extension from VS Code Editor. Open Command Palette *(Ctrl+Shift+P)* and run command 'Hello World'.  
   This time the extension will be launched without any Development Environment Host Window.



# Extension File Structure

The default structure of vs code extension built using the Yeoman and Extension Generator tool is as follows :

1. **.vscode :** A VS Code workspace is usually a project root folder. Settings related to Workspace, Debugging and Task configurations, List of extensions identifier *({publisherName}.{extensionName})* are stored in the project root folder.  
   *a) launch.json -> Configuration for launching and debugging the extension.  
   b) tasks.json -> Configuration for build task(s) that compiles TypeScript.*[Learn more at What is a VS Code "workspace"?](https://code.visualstudio.com/docs/editor/workspaces) article.
2. **node\_modules :** Download location for locally installed *(default)* packages *(project dependencies)* required for current project.

NPM drops packages *(project dependencies)* in the node\_modules folder whenever packages are installed locally.

1. **out :** Default build output directory.
2. **src/extension.ts :** Extension source code files.
3. **package.json :** Extension manifest.   
   This JSON file contains fields of Node.js such as scripts, required dependencies , Visual Studio Code specific attributes such as publisher name, contributions like *Activity Bar, Views, View Container, Commands*, version which are required to package extension.
4. **.gitignore :** Git file to Ignore build output and node\_modules.
5. **tsconfig.json :** Specifies the root files of the Typescript project and the compiler options required to compile the project. [Read More.](https://www.typescriptlang.org/docs/handbook/tsconfig-json.html)
6. **README.md :** Readable description of your extension's functionality.
7. **.vscodeignore :** Exclude some files from being included in the extension's package.   
   Development dependencies listed in *'devDependencies'* will be automatically ignored, no need to add them in the .vscodeignore file.

# Download From GitHub Sample Extension can be downloaded from [GitHub](https://github.com/KumarVariable/myfirstcustomextension).