

21.1.2020

Node.js Workshop 1: Editor set-up and GitHub integration, Command Line

After completing this workshop the student knows how to:

- Use Visual Studio Code to edit and debug files
- Integrate editor to GitHub and pull/push code from it
- Use Nodemon -command line tool to run the programs

Create a new folder called WS1 for these assignments. Place all your code there.

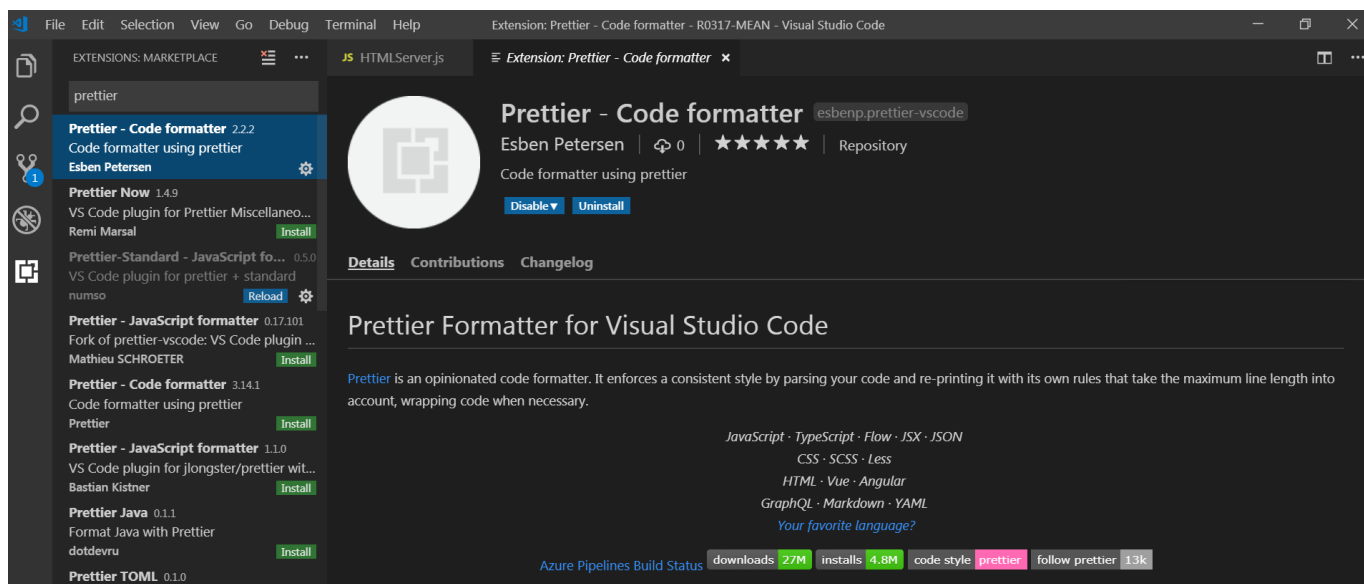
Basics

See the basics of using VSCode:

<https://code.visualstudio.com/docs/editor/codebasics>

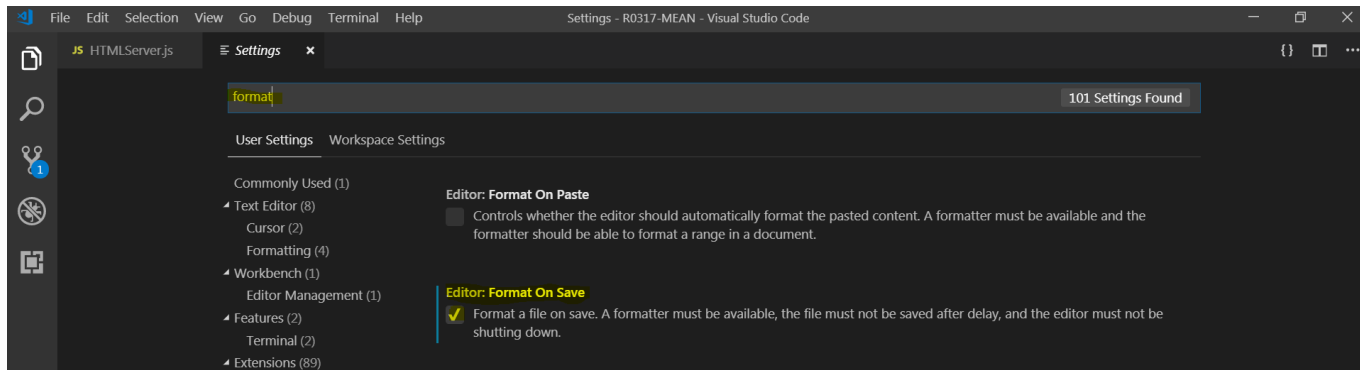
Enhancing the editor

1. Install package called “prettier” to your coding editor (Visual Studio Code or Atom). It formats JSON and JavaScript nicely.
2. Click Extensions-button on the left side menu and then type in “prettier”. Then click “install”.



3. Once you have installed it, you can use it with CTRL + CMD + P (MacOS) or CTRL + Shift + P (Windows) to manually format a file or a selection of code.
4. If you don't want to format your file manually every time, you can format it on save as well.
5. Open up the VS Code User's settings/preferences as UI, search for "Format On Save" and make sure to activate it.

21.1.2020



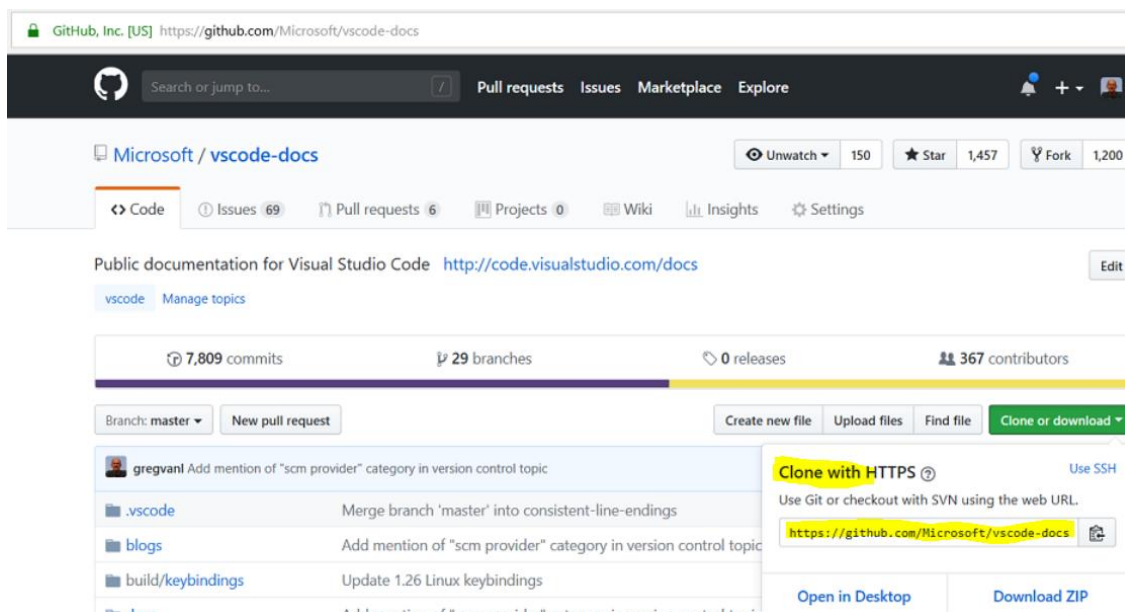
Debugging

See the basic usage of VSCode and Node apps below. Try to run and debug the demo app.
<https://code.visualstudio.com/docs/nodejs/nodejs-tutorial>

Synchronizing with GitHub

Modern web development is all about version control. For this, you will be using Git and GitHub. You can clone a Git repository with the “Git: Clone” command in the Command Palette (Ctrl+Shift+P). You will be asked for the URL of the remote repository (for example on GitHub) and the parent directory under which to put the local repository.

For a GitHub repository, you would find the URL from the GitHub Clone or download dialog. You would then paste that URL into the Git: Clone prompt. Then give a local folder name where to store the files. Now all the files are saved into your computer.



21.1.2020

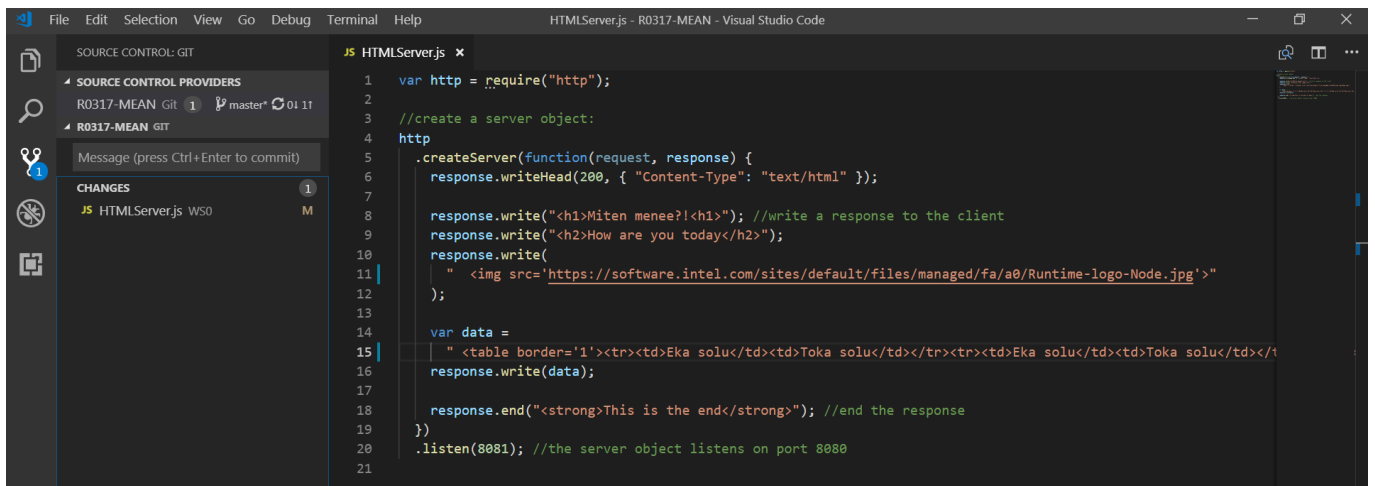
Now do the same thing for your own GitHub-repository created in WS0.

Step 1: Open the GitHub repo in your browser (“R0314 Mean”) and copy the clone-link

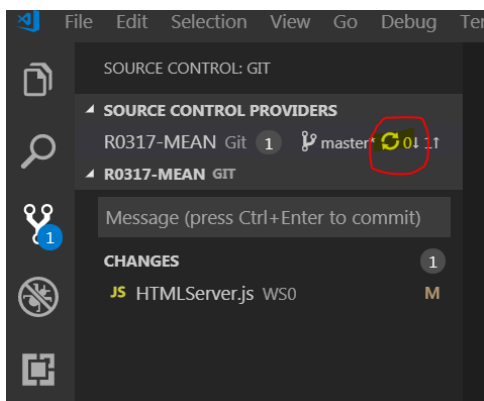
Step 2: Run Git:Clone command from Visual Studio Code (Ctrl-Shift-P)

Step 3: Edit some of the files created earlier, and note how the changes are displayed in the editor (see image below; blue lines after line number indicate changes on your local copy). You can even create a text file with some dummy data in it.

Step 4: Move folder WS1 under your local copy of the cloned repository



Step 5: Lets do “Push” operation, which will synchronize the code back to your GitHub-repo. Click on the “Synchronize changes” icon on the left side menu. You also need to add a short “Message” to every commit operation, describing what has changed. In this you could just type “added WS1”.



Step 6: Finally go and see the GitHub repo in your Browser. All the files with recent changes should be there.

So this is the workflow when editing files and syncing the via Git. If there were many people working on the same code, then you would see their changes when refreshing the code as well. And you can continue working anywhere you like, by cloning the repo again.

21.1.2020

Nodemon tools

When working with Node.js apps, you often times need to restart the app in order to see your changes in effect. This happens especially when working with server apps. In order to get around this we can use a tool called “nodemon”. It watches the directory and restarts the app automatically if some files within the directories change.

You can install nodemon with “npm” tool by typing “npm install nodemon” in your Node command prompt.

Then you must run your app by typing “nodemon myApp.js”. Now your app restarts every time you edit the source files and save.

Try running some of the apps you have created during workshops 0 and 1 using nodemon. While the app is running, edit the file and save and you should see the app restarting automatically.