

Setup

- Clone microsoft/vscode
- Run **yarn** at /, this will install
 - Dependencies for `/extension/css-language-features/`
 - Dependencies for `/extension/css-language-features/server/`
 - devDependencies such as **gulp**
- Open `/extensions/css-language-features/` as the workspace in VS Code
- In `/extensions/css-language-features/` run **yarn compile**(or **yarn watch**) to build the client and server
- Run the **Launch Extension** debug target in the Debug View. This will:
 - Launch a new VS Code instance with the `css-language-features` extension loaded
- Open a `.css` file to activate the extension. The extension will start the CSS language server process.
- Add `"css.trace.server": "verbose"` to the settings to observe the communication between client and server in the **CSS Language Server** output.
- Debug the extension and the language server client by setting breakpoints in `incss-language-features/client/`
- Debug the language server process by using **Attach to Node Process** command in the VS Code window opened on `css-language-features`.
 - Pick the process that contains `cssServerMain` in the command line. Hover over `code-insiders` resp `code` processes to see the full process command line.
 - Set breakpoints in `css-language-features/server/`
- Run **Reload Window** command in the launched instance to reload the extension

Contribute to vscode-css-languageservice

microsoft/vscode-css-languageservice contains the language smarts for CSS/SCSS/Less. This extension wraps the css language service into a Language Server for VS Code. If you want to fix CSS/SCSS/Less issues or make improvements, you should make changes at microsoft/vscode-css-languageservice.

However, within this extension, you can run a development version of `vscode-css-languageservice` to debug code or test language features interactively:

Linking `vscode-css-languageservice` in `css-language-features/server/`

- Clone microsoft/vscode-css-languageservice
- Run **yarn** in `vscode-css-languageservice`
- Run **yarn link** in `vscode-css-languageservice`. This will compile and link `vscode-css-languageservice`
- In `css-language-features/server/`, run **yarn link vscode-css-languageservice**

Testing the development version of `vscode-css-languageservice`

- Open both `vscode-css-languageservice` and this extension in a single workspace with multi-root workspace feature
- Run `yarn watch` in `vscode-css-languageservice` to recompile the extension whenever it changes
- Run `yarn watch` at `css-language-features/server/` to recompile this extension with the linked version of `vscode-css-languageservice`
- Make some changes in `vscode-css-languageservice`
- Now when you run **Launch Extension** debug target, the launched instance will use your development version of `vscode-css-languageservice`. You can interactively test the language features.