

Setup

- Clone microsoft/vscode
- Run **yarn** at /, this will install
 - Dependencies for /extension/json-language-features/
 - Dependencies for /extension/json-language-features/server/
 - devDependencies such as gulp
- Open /extensions/json-language-features/ as the workspace in VS Code
- In /extensions/json-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 **json-language-features** extension loaded
- Open a **.json** file to activate the extension. The extension will start the JSON language server process.
- Add **"json.trace.server": "verbose"** to the settings to observe the communication between client and server in the **JSON Language Server** output.
- Debug the extension and the language server client by setting breakpoints in **json-language-features/client/**
- Debug the language server process by using **Attach to Node Process** command in the VS Code window opened on **json-language-features**.
 - Pick the process that contains **jsonServerMain** in the command line. Hover over **code-insiders** resp **code** processes to see the full process command line.
 - Set breakpoints in **json-language-features/server/**
- Run **Reload Window** command in the launched instance to reload the extension

Contribute to vscode-json-languageservice

microsoft/vscode-json-languageservice is the library that implements the language smarts for JSON. The JSON language server forwards most of requests to the service library. If you want to fix JSON issues or make improvements, you should make changes at microsoft/vscode-json-languageservice.

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

Linking **vscode-json-languageservice** in **json-language-features/server/**

- Clone microsoft/vscode-json-languageservice
- Run **npm install** in **vscode-json-languageservice**
- Run **npm link** in **vscode-json-languageservice**. This will compile and link **vscode-json-languageservice**

- In `json-language-features/server/`, run `yarn link vscode-json-languageservice`

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

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