Recommended Settings and Extensions in VS Code

Table of contents

VS Code is highly customizable and has a rich ecosystem of extensions that can extend VS Code's functionality. In this section, we recommend several setting configurations and extensions that we have found provide the best user experience for data scientists using VS Code on Posit Workbench.

Recommended settings

We recommend configuring the following settings at the user-level. To edit your user settings file activate the Command Palette using `cmd` + `shift` + `p` and then type `Preferences: Open Settings (JSON)`.

```json
{
  "[python]": {
    "editor.rulers": [
      80
    ],
    "editor.tabSize": 4
  },
  "[r]": {
    "editor.rulers": [
      80
    ],
    "editor.tabSize": 2
  },
  "jupyter.sendSelectionToInteractiveWindow": true
}
```

`[python]`

All values in this section apply only to `.py` files. Enable a vertical ruler at 80 spaces and set tab size to 4 spaces to align with best practice per pep8.

`[r]`

All values in this section will apply only to `.r` files. Choose values that align with your style guide of choice. For example, enable a vertical ruler at 80 spaces and set tab size to 2 spaces to align with best practice per the tidyverse style guide.

`jupyter.sendSelectionToInteractiveWindow`

Send the selected code from a `.py` file directly to an interactive jupyter window by pressing **Shift + Enter**

Recommended extensions

We recommend installing the following extensions. To read more about installing extensions, please refer to the VS Code Extensions section.

Quarto

The Quarto extension provides language support for `.qmd` files. Learn more about Quarto at https://quarto.org.

Python

The official Python extension from Microsoft provides rich language support for the Python Language. Features include:

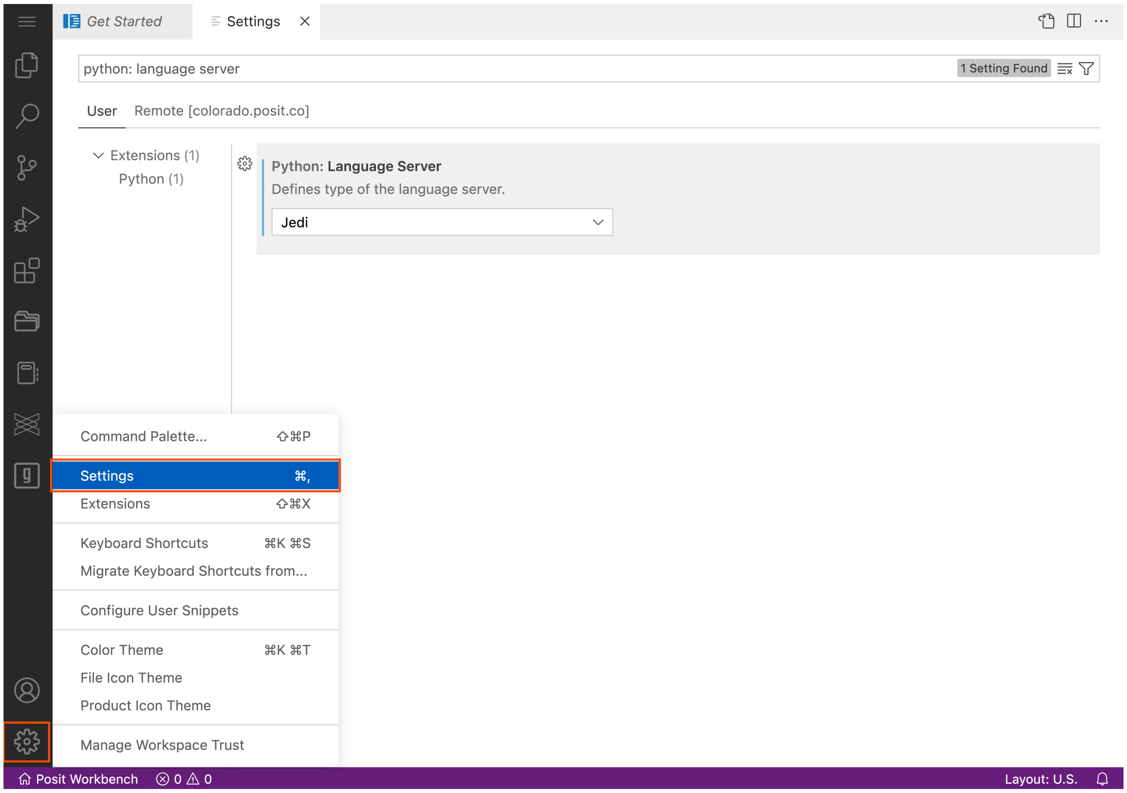
- The ability to create, edit, and execute Jupyter Notebooks
- Code refactoring
- Code auto-completion
- Code formatting
- Debugging tools
- Unit testing tools

ℹ Note

When using the Python extension on Posit Workbench, VS Code is not able to use the Pylance language server (source).

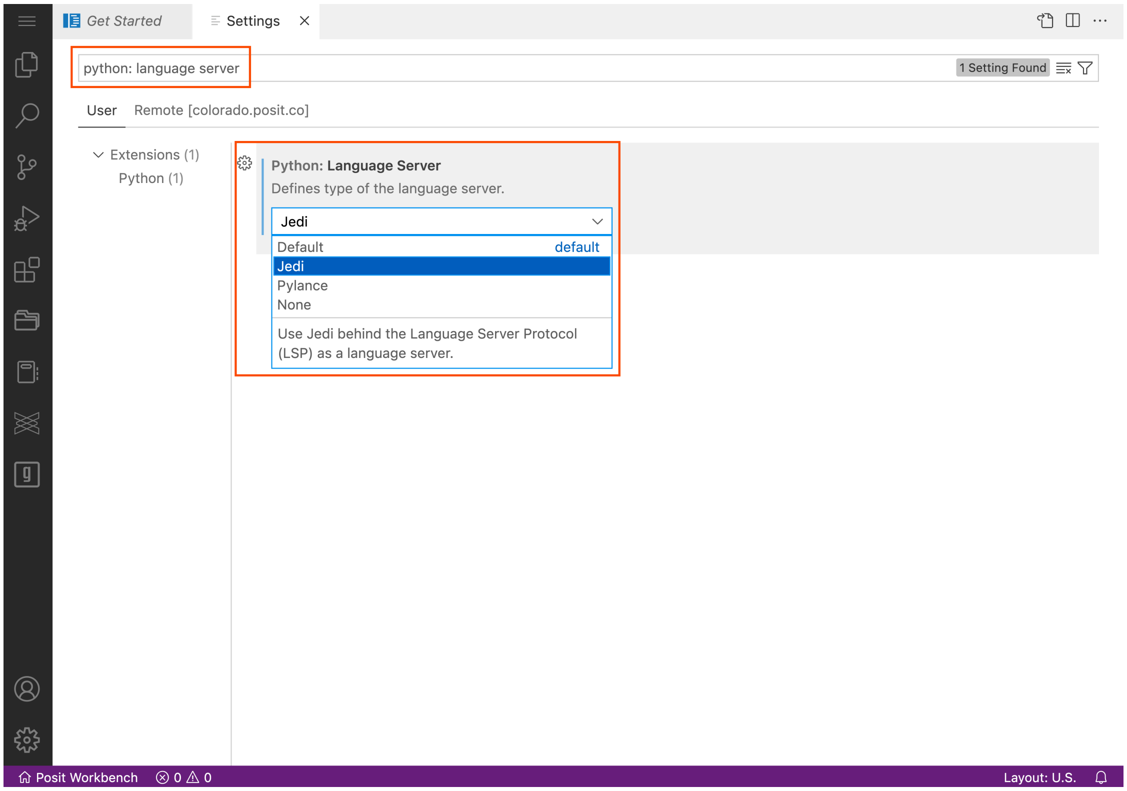
> Pylance is licensed for use in Microsoft products and services only, so it cannot be used with Posit Workbench.

Pylance is an optional dependency.
Users can switch the Python Language Server Protocol (LSP) to the other bundled Language Server, Jedi. Activating Jedi will re-enable rich language support such as code completion, error reporting, type checking, and other useful Python support that otherwise would be missing as Pylance is non-functional.
Users can modify settings from the UI editor or edit `settings.json` files directly. Please refer to the screenshots below to access the UI editor, search for `Python: Language Server`, and set Jedi as the language server



python: language server

Python: Language Server
Defines type of the language server.
Jedi



Python: Language Server
Defines type of the language server.

Jedi

- Default — default
- Jedi
- Pylance
- None

Use Jedi behind the Language Server Protocol (LSP) as a language server.

> **ℹ Note**
>
> The Python extension automatically also installs the Jupyter extension. If you are working in an offline environment and want to install the Python extension, you will need to install the Jupyter extension manually.

R

The R extension provides support for the R programming language. For the extension to work, you must install the `languageserver` in R.

```r
install.packages("languageserver")
```

Databricks

The Databricks extension for VS Code allows you to remotely interact with the Databricks platform. This VS Code extension can make use of Databricks delegated credentials if it is available and enabled for your Workbench instance.

Code Spell Checker

The Code Spell Checker extension provides basic spell checking that works with code and documents. In text-based documents like `.md` and `.qmd` files, the spell checker checks all words. In code-based documents like `.py` and `.R` files, the spell checker only checks strings and comments.

```
{
  "cSpell.enableFiletypes": [
    "md",
    "plaintext",
    "python",
    "quarto",
    "r",
    "rmd"
  ]
}
```

Git Graph

Git Graph creates a visual and interactive graph of your git repository. The extension allows you to perform common git operations without using the terminal.

Project Manager

VS Code allows you to work on a specific project by opening the project's root directory with VS Code. The Project Manager extension extends this capability by adding several helpful features:

- Save folders or workspaces as projects that are exposed via the sidebar
- Auto-detection of all git repositories as projects
- The ability to quickly switch between projects

File Utils

File Utils provides a convenient way to quickly perform actions on files and directories. Users can perform common operations on files and directories, such as renaming, deleting, and moving the following by using the Command Palette or by right-clicking the file/directory.

Path Intellisense

The Path Intellisense extension provides auto-completion for file names and directories.