RStudio™ Support

Premium email support

Sign in

RStudio Support > How To Articles > RStudio Connect

Search

Installing and Configuring Python with RStudio

Kristopher Overholt

Today at 06:37

Follow

The following steps represent a minimal workflow for using Python with RStudio Connect via the reticulate package, whether you are using the RStudio IDE on your local machine or RStudio Server Pro.

Step 1) Install a base version of Python

If you are working on your local machine, you can install Python from Python.org or Anaconda.

If you are working on a server with RStudio Server Pro, your administrator can install a system-wide version of Python, or you can install Python in your home directory from Python.org or Anaconda.

Be sure to start a new terminal session to ensure your newly installed Python is active.

Also, ensure that your installation of Python has the `virtualenv` package installed by running:

```
pip install virtualenv
```

Step 2) Create a Python environment in your project

It is recommended that you use one virtual environment per project, similar to how packrat is used to manage R packages within a project.

Navigate into your your RStudio project directory by using the following command:

```
cd <project-dir>
```

Create a new virtual environment in a folder called `python` within your project directory using the following command:

```
virtualenv python
```

Step 3) Activate your Python environment

You can activate the `virtualenv` in your project using the following command in a terminal:

Recently viewed articles

Version Control with Git and SVN

Using Projects

Related articles

Best Practices for Using Python with RStudio Connect

Using the RStudio Terminal

Troubleshooting Python with RStudio Connect

FAQ for Using Python with RStudio Connect

Configuring Python with RStudio Server Pro and RStudio Connect

```
source python/bin/activate
```

You can verify that you have activated the correct version of Python using the following command in a terminal:

```
which python
```

Step 4) Install Python packages in your environment

You can install Python packages such as `numpy`, `pandas`, `matplotlib`, and other packages in your Python `virtualenv` by using `pip install` using the following command in a terminal:

```
pip install numpy pandas matplotlib
```

Step 5) Install and configure `reticulate` to use your Python version

Install the `reticulate` package using the following command in your R console:

```
install.packages("reticulate")
```

To configure `reticulate` to point to the Python executable in your `virtualenv`, create a file in your project directory called `.Rprofile` with the following contents:

```
Sys.setenv(RETICULATE_PYTHON = "python/bin/python")
```

You'll need to restart your R session for the setting to take effect. You can verify that `reticulate` is configured for the correct version of Python using the following command in your R console:

```
reticulate::py_config()
```

Step 6) Publish a project to RStudio Connect

You can then develop Shiny apps, R Markdown, and Plumber APIs with Python/R in the RStudio IDE and RStudio Server Pro using the `reticulate` package per https://blog.rstudio.com/2018/10/09/rstudio-1-2-preview-reticulated-python/ and https://rstudio.github.io/reticulate/ and deploy the applications to RStudio Connect.

For more details on each step, refer to the concepts and best practices in the support article for Best Practices for Using Python with RStudio Connect.

Was this article helpful?
14 out of 84 found this helpful

Comments