
Article

[Guillaume Rongier](#) · Nov 30, 2022 2m read

Embedded Python Virtual Environnement (venv)

If you are using Python, you can use the built-in venv module to create a virtual environment. This module is the recommended way to create and manage virtual environments.

A virtual environment is a tool that helps to keep dependencies required by different projects separate by creating isolated python virtual environments for them. It solves the “ Project X depends on version 1.x but, Project Y needs 4.x ” dilemma, and keeps your global site-packages directory clean and manageable.

So if like me you work a lot with Python, you can use the venv module to create a virtual environment for your project. This will allow you to install packages without affecting the global Python installation.

You will find here two neat alias to create and activate a virtual environment.

Python aliases

```
alias venv="python3 -m venv .venv; source .venv/bin/activate"
alias irisvenv="python3 -m venv .venv; source .venv/bin/activate; pip install https://github.com/grongierisc/iris-embedded-python-wrapper/releases/download/v0.0.3/iris-0.0.3-py3-none-any.whl"
```

Let dive into the details of each alias.

Python venv

The first alias is a simple one. It will create a virtual environment in the current directory and activate it.

```
python3 -m venv .venv
```

Create an environment named .venv in the current directory.

```
source .venv/bin/activate
```

Activate the environment.

InterSystems IRIS venv

The second alias is the same as the first one except that it will install the InterSystems IRIS Python wrapper in the virtual environment.

```
python3 -m venv .venv
```

Create an environment named .venv in the current directory.

```
source .venv/bin/activate
```

Activate the environment.

```
pip install https://github.com/grongierisc/iris-embedded-python-  
wrapper/releases/download/v0.0.3/iris-0.0.3-py3-none-any.whl
```

Install the InterSystems IRIS Python wrapper.

This module is a wrapper around the InterSystems IRIS Embedded Python API. It allows you to connect to an InterSystems IRIS instance and execute SQL queries.

To make it work you need to have an environment variable named IRISINSTALLDIR pointing to the InterSystems IRIS installation directory.

```
export IRISINSTALLDIR=/opt/iris  
export LD_LIBRARY_PATH=$IRISINSTALLDIR/bin:$LD_LIBRARY_PATH  
# for MacOS  
export DYLD_LIBRARY_PATH=$IRISINSTALLDIR/bin:$DYLD_LIBRARY_PATH
```

Conclusion

I hope you will find this post useful. If you have any questions or comments, please feel free to leave a comment below.

[#Embedded Python](#) [#Python](#) [#InterSystems IRIS](#)

Source URL: <https://community.intersystems.com/post/embedded-python-virtual-environment-venv>