Examples Languages Python

Python

Python by default will attempt to install your packages globally, or in the Nix Store (which it does not have permissions to modify). To use Python with Devbox, we recommend setting up a Virtual Environment using pipenv or Poetry (see below).

Example Repo

Adding Python to your Project

`devbox add python@3.10`, or in your `devbox.json` add:

```
  "packages": [
    "python@3.10"
  ],
```

This will install Python 3.10 in your shell. You can find other versions of Python by running `devbox search python`. You can also view the available versions on **Nixhub**

Installing Packages with Pip

Example Repo

Open in **Devspace**

pip is the standard package manager for Python. Since it installs python packages globally, we strongly recommend using a virtual environment.

The `python` package automatically comes bundled with `pip`, and the `python` plugin for Devbox will automatically create a virtual environment for installing your packages locally

Your virtual environment is created in the `.devbox/virtenv/python` directory by default, and can be activated by running `. $VENV_DIR/bin/activate` in your Devbox shell. You can activate the virtual environment automatically using the init_hook of your `devbox.json`:

```
{
    "packages": [
        "python@3.10"
    ],
    "shell": {
        "init_hook": ". $VENV_DIR/bin/activate"
    }
}
```

> ⓘ INFO
>
> For Fish or other shells, you may need to use a different activation script. See the venv docs for more details.

Devbox installs the virtual environment in `.devbox/virtenv/python/.venv` by default. You can modify this path by setting the `VENV_DIR` environment variable in your devbox.json:

```
{
    "packages": [
        "python@3.10"
    ],
    "env": {
        // Install your virtual environment in `.venv`
        "VENV_DIR": ".venv"
    },
    "shell": {
        "init_hook": ". $VENV_DIR/bin/activate"
    }
}
```

If you need to install a specific version of Pip, you can run `devbox add python3xxPackages.pip`, where `3xx` is the major + minor version (e.g., python310 = python@3.10) of Python you want to install:

```
{
    "packages": [
        "python@3.10"
        "python310Packages.pip"
    ],
    "shell": {
        "init_hook": ". $VENV_DIR/bin/activate"
    }
}
```

Pipenv

Example Repo

Open in **Devspace**

pipenv is a tool that will automatically set up a virtual environment for installing your PyPi packages.

You can install `pipenv` by adding it to the packages in your `devbox.json`. You can then manage your packages and virtual environment via a Pipfile

```
{
    "packages": [
        "python310",
        "pipenv"
    ],
    "shell": {
        "init_hook": "pipenv shell"
    }
}
```

This init_hook will automatically start your virtualenv when you run `devbox shell`.

Poetry

Example Link

Open in **Devspace**

Poetry is a packaging and dependency manager for Python that helps you manage your Python packages, and can automatically create a virtual environment for your project.

You can install Poetry by adding it to the packages in your `devbox.json`. You can then manage your packages and virtual environment via a `pyproject.toml`

```
{
    "packages": [
        "python3",
        "poetry"
    ],
    "shell": {
        "init_hook": "poetry shell"
    }
}
```

This init_hook will automatically start Poetry's virtualenv when you run `devbox shell`, and provide you with access to all your packages.

Disabling the Python Plugin

If you would prefer to disable the Python Plugin, you can add python using `devbox add python --disable_plugin` or in your `devbox.json`:

```
{
    "packages": {
        "python": {
            "version": "3.10",
            "disable_plugin": true
        }
    }
}
```

Edit this page