
pipnest

Dec 28, 2022

Contents:

1	Usage	3
1.1	Project structure	3
1.2	Initialize project	3
1.3	Package	3
1.4	Upload to PyPI	4
2	Indices and tables	5

pipnest packages your NEST extension modules into pip installable packages. This is a benefit when you're distributing a Python model for the NEST simulator and have a dependency on an extension module. Instead of having to provide installation instructions to your end-users you can have pip download and build the extension module into the target machine's nest installation.

1.1 Project structure

- Prepare a folder with 1 subfolder that contains a NEST extension modules.
- The source code of the extension module can not be in the root repository.
- The name of your module should be the name of your folders plus “module”.

Your project structure should look like this:

```
- my_module
  - my_module
    - my_file1.cpp
    - my_file2.cpp
    - ...
```

In this example your NEST module should be called the `my_modulemodule`.

1.2 Initialize project

Run the `pipnest init` command from the root folder and fill in the requested metadata:

```
cd my_module
pipnest init .
```

This should create a `setup.py` and `README.md`, be sure to edit the latter. Check that your module is present under the `packages` keyword argument in `setup.py`.

1.3 Package

Create the source distribution:

```
python setup.py sdist
```

1.3.1 Test your package locally

- Remove any already installed versions from the `$NEST_INSTALL_DIR/lib/nest` folder
- Run `pip install dist/*`. (If an MPI error occurs, restart your terminal and try again)

1.4 Upload to PyPI

Use twine to upload your package:

```
twine upload dist/* --skip-existing
```


CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`