

Make the Project Installable

Making your project installable means that you can build a *distribution* file and install that in another environment, just like you installed Flask in your project's environment. This makes deploying your project the same as installing any other library, so you're using all the standard Python tools to manage everything.

Installing also comes with other benefits that might not be obvious from the tutorial or as a new Python user, including:

- Currently, Python and Flask understand how to use the `flaskr` package only because you're running from your project's directory. Installing means you can import it no matter where you run from.
- You can manage your project's dependencies just like other packages do, so `pip install yourproject.whl` installs them.
- Test tools can isolate your test environment from your development environment.

Note:

This is being introduced late in the tutorial, but in your future projects you should always start with this.

Describe the Project

The `setup.py` file describes your project and the files that belong to it.

`setup.py`

```
from setuptools import find_packages, setup

setup(
    name='flaskr',
    version='1.0.0',
    packages=find_packages(),
    include_package_data=True,
    zip_safe=False,
    install_requires=[
        'flask',
    ],
)
```

`packages` tells Python what package directories (and the Python files they contain) to include. `find_packages()` finds these directories automatically so you don't have to type them out. To include other files, such as the static and templates directories, `include_package_data` is set. Python needs another file named `MANIFEST.in` to tell what this other data is.

`MANIFEST.in`

```
include flaskr/schema.sql
graft flaskr/static
graft flaskr/templates
global-exclude *.pyc
```

This tells Python to copy everything in the `static` and `templates` directories, and the `schema.sql` file, but to exclude all bytecode files.

See the [official packaging guide](#) for another explanation of the files and options used.

Install the Project

Use `pip` to install your project in the virtual environment.

```
$ pip install -e .
```

This tells pip to find `setup.py` in the current directory and install it in *editable* or *development* mode. Editable mode means that as you make changes to your local code, you'll only need to re-install if you change the metadata about the project, such as its dependencies.

You can observe that the project is now installed with `pip list`.

```
$ pip list
```

Package	Version	Location
click	6.7	
Flask	1.0	
flaskr	1.0.0	/home/user/Projects/flask-tutorial
itsdangerous	0.24	
Jinja2	2.10	
MarkupSafe	1.0	
pip	9.0.3	
setuptools	39.0.1	
Werkzeug	0.14.1	
wheel	0.30.0	

Nothing changes from how you've been running your project so far. `FLASK_APP` is still set to `flaskr` and `flask run` still runs the application, but you can call it from anywhere, not just the `flask-tutorial` directory.

Continue to [Test Coverage](#).