

Warning: This is the development version. The latest stable version is Version 2.3.x.

Installation

Python Version

We recommend using the latest version of Python. Flask supports Python 3.8 and newer.

Dependencies

These distributions will be installed automatically when installing Flask.

- [Werkzeug](#) implements WSGI, the standard Python interface between applications and servers.
- [Jinja](#) is a template language that renders the pages your application serves.
- [MarkupSafe](#) comes with Jinja. It escapes untrusted input when rendering templates to avoid injection attacks.
- [ItsDangerous](#) securely signs data to ensure its integrity. This is used to protect Flask's session cookie.
- [Click](#) is a framework for writing command line applications. It provides the flask command and allows adding custom management commands.
- [Blinker](#) provides support for [Signals](#).

Optional dependencies

These distributions will not be installed automatically. Flask will detect and use them if you install them.

- [python-dotenv](#) enables support for [Environment Variables From dotenv](#) when running flask commands.
- [Watchdog](#) provides a faster, more efficient reloader for the development server.

 v: latest ▼

greenlet

You may choose to use `gevent` or `eventlet` with your application. In this case, `greenlet>=1.0` is required. When using PyPy, `PyPy>=7.3.7` is required.

These are not minimum supported versions, they only indicate the first versions that added necessary features. You should use the latest versions of each.

Virtual environments

Use a virtual environment to manage the dependencies for your project, both in development and in production.

What problem does a virtual environment solve? The more Python projects you have, the more likely it is that you need to work with different versions of Python libraries, or even Python itself. Newer versions of libraries for one project can break compatibility in another project.

Virtual environments are independent groups of Python libraries, one for each project. Packages installed for one project will not affect other projects or the operating system's packages.

Python comes bundled with the `venv` module to create virtual environments.

Create an environment

Create a project folder and a `.venv` folder within:

macOS/Linux

Windows

```
$ mkdir myproject
$ cd myproject
$ python3 -m venv .venv
```

Activate the environment

Before you work on your project, activate the corresponding environment.  **v: latest** ▼

macOS/Linux

Windows

```
$ . .venv/bin/activate
```

Your shell prompt will change to show the name of the activated environment.

Install Flask

Within the activated environment, use the following command to install Flask:

```
$ pip install Flask
```

Flask is now installed. Check out the [Quickstart](#) or go to the [Documentation Overview](#).