

Development Server

Starting with Flask 0.11 there are multiple built-in ways to run a development server. The best one is the **flask** command line utility but you can also continue using the **Flask.run()** method.

Command Line

The **flask** command line script ([Command Line Interface](#)) is strongly recommended for development because it provides a superior reload experience due to how it loads the application. The basic usage is like this:

```
$ export FLASK_APP=my_application
$ export FLASK_ENV=development
$ flask run
```

This enables the development environment, including the interactive debugger and reloader, and then starts the server on *http://localhost:5000/*.

The individual features of the server can be controlled by passing more arguments to the **run** option. For instance the reloader can be disabled:

```
$ flask run --no-reload
```

Note:

Prior to Flask 1.0 the **FLASK_ENV** environment variable was not supported and you needed to enable debug mode by exporting **FLASK_DEBUG=1**. This can still be used to control debug mode, but you should prefer setting the development environment as shown above.

In Code

The alternative way to start the application is through the **Flask.run()** method. This will immediately launch a local server exactly the same way the **flask** script does.

Example:

```
if __name__ == '__main__':
    app.run()
```

This works well for the common case but it does not work well for development which is why from Flask 0.11 onwards the **flask** method is recommended. The reason for this is that due to how the reload mechanism works there are some bizarre side-effects (like executing certain code twice, sometimes crashing without message or dying when a syntax or import error happens).

It is however still a perfectly valid method for invoking a non automatic reloading application.