Python Workers

- Introduction
  - The pywrangler CLI tool
  - Python Worker Templates
- Next Up

import { WranglerConfig } from "~/components";

Cloudflare Workers provides a first-class Python experience, including support for:

- Easy to install and fast-booting Packages, including FastAPI, Langchain, Pydantic and more.
- A robust foreign function interface (FFI) that lets you use JavaScript objects and functions directly from Python — including all Runtime APIs
- An ecosystem of services on the Workers Platform accessible via bindings, including:
  - State storage and databases like KV, D1, Durable Objects
  - Access to Environment Variables, Secrets, and other Workers using Service Bindings
  - AI capabilities with Workers AI, Vectorize
  - File storage with R2
  - Durable Workflows, Queues, and more

Introduction

A Python Worker can be as simple as four lines of code:

```
from workers import WorkerEntrypoint, Response


class Default(WorkerEntrypoint):
    async def fetch(self, request):
        return Response("Hello World!")
```

Similar to other Workers, the main entry point for a Python worker is the `fetch` handler which handles incoming requests sent to the Worker.

In a Python Worker, this handler is placed in a `Default` class that extends the `WorkerEntrypoint` class (which you can import from the `workers` SDK module).

:::note

You must add the `python_workers` compatibility flag to your Worker.

We'd love your feedback. Join the #python-workers channel in the Cloudflare Developers Discord and let us know what you'd like to see next. :::

The `pywrangler` CLI tool

To run a Python Worker locally, install packages, and deploy it to Cloudflare, you use pywrangler, the CLI for Python Workers.

To set it up, first, ensure uv and Node are installed.

Then set up your development environment:

```
uvx --from workers-py pywrangler init
```

This will create a `pyproject.toml` file with `workers-py` as a development dependency. `pywrangler init` will create a wrangler config file. You can then run `pywrangler` with:

```
uv run pywrangler dev
```

To deploy a Python Worker to Cloudflare, run `pywrangler deploy`:

```
uv run pywrangler deploy
```

Python Worker Templates

When you initialize a new Python Worker project and select from one of many templates:

```
uv run pywrangler init
```

Or you can clone the examples repository to explore more options:

```
git clone https://github.com/cloudflare/python-workers-examples
cd python-workers-examples/hello
```

Next Up

- Learn more about the basics of Python Workers
- Learn details about local development, deployment, and how to Python Workers work.
- Explore the package docs for instructions on how to use packages with Python Workers.
- Understand which parts of the Python Standard Library are supported in Python Workers.
- Learn about Python Workers' foreign function interface (FFI), and how to use it to work with bindings and Runtime APIs.

□□□ #2
□ □□□ □□□ 2026-11-01 05:43:31 UTC
□ □□□ □□□ 2026-11-01 05:43:34 UTC