

How to use the Python SDK

Prerequisites

- An [API key](#)
- Familiarity with command-line tools
- Basic familiarity with Python; Python 3+ installed

Installation

Our Python SDK is hosted at the [Python Package Index](#) (PyPI). You can use any Python package manager, such as [pip](#) or [poetry](#).

For example, you can add this dependency to your project's build file:

```
1 pip install fluidstack
2 # or
3 poetry add fluidstack
```

Instantiate a client

Import `FluidStack` and instantiate a client with your API key:

```
Example
1 from FluidStack.client import FluidStack
2
3 client = FluidStack(
4     api_key = "<your_api_key>"
5 )
```

Now you can use the client to consume the API from your Python application.

⚠ Avoid placing your API key in any file that might be shared with others. For information on using a `.env` file instead of adding the API key directly into a file, see: [API Overview - Secure use of your API key](#).

Use the FluidStack client in your app

The FluidStack client simplifies making API requests. It stores the API key that you used to instantiate it, and it already knows our API server's base URL, so you can omit those details in your requests.

The SDK also provides code hints, type hints, parameter information, and other useful functions to speed up development.

For example, compare the tabs below:

```
With the SDK Without the SDK
1 client.instances.create(
2     name="my_instance",
3     gpu_type="RTX_46000_48GB",
4     ssh_key="my_ssh_key"
5 )
```

You can see that instead of using `requests` or a similar module to send an HTTP request and including the endpoint path and headers each time, the FluidStack client simplifies the request for you.

Call endpoints

Requests to API endpoints are implemented in the SDK as methods of the client. For example, the [GET /instances](#) endpoint is implemented like this:

```
List user instances with the SDK
1 client.instances.list()
```

The code shown above does not do anything with the response from the endpoint. It is up to you to handle the response.

For example, you could simply print the entire response to the terminal:

```
Print the entire response
1 print(client.instances.list())
```

Or you could loop through the list and print only the name and status for each instance:

```
Print each instance name and status
1 my_instances = client.instances.list()
2 for instance in my_instances:
3     print(f"Instance: {instance.name}, Status: {instance.status}")
```

Instances

List user instances [Create an instance](#) [Stop an instance](#) [Start an instance](#) [Terminate an instance](#)

On this page

Prerequisites

[Installation](#)[Instantiate a client](#)[Use the FluidStack client in your app](#)[Call endpoints](#)[Instances](#)[SSH Keys](#)[List available configurations and operating system templates](#)

```

1 client.instances.create(
2   name="my_instance",
3   gpu_type="my_gpu_type",
4   ssh_key="my_ssh_key",
5   gpu_count="my_gpu_count"
6   operating_system_label="my_os_label"
7 )
    
```

parameter	type	required	default
name	string	yes	n/a
gpu_type	string	yes	n/a
ssh_key	string	yes	n/a
gpu_count	string	no	1
operating_system_label	string	no	ubuntu_20_04_lts_nvidia

SSH Keys

List SSH keys [Delete an SSH key](#)

```

1 client.ssh_keys.delete(
2   ssh_key_name="my_ssh_key",
3 )
    
```

parameter	type	required
ssh_key_name	string	yes

List available configurations and operating system templates

[Configurations](#) [OS Templates](#)

```

1 client.configurations.list()
    
```

Parameters: None.

Was this page helpful? Yes No

List user instances
Up Next >