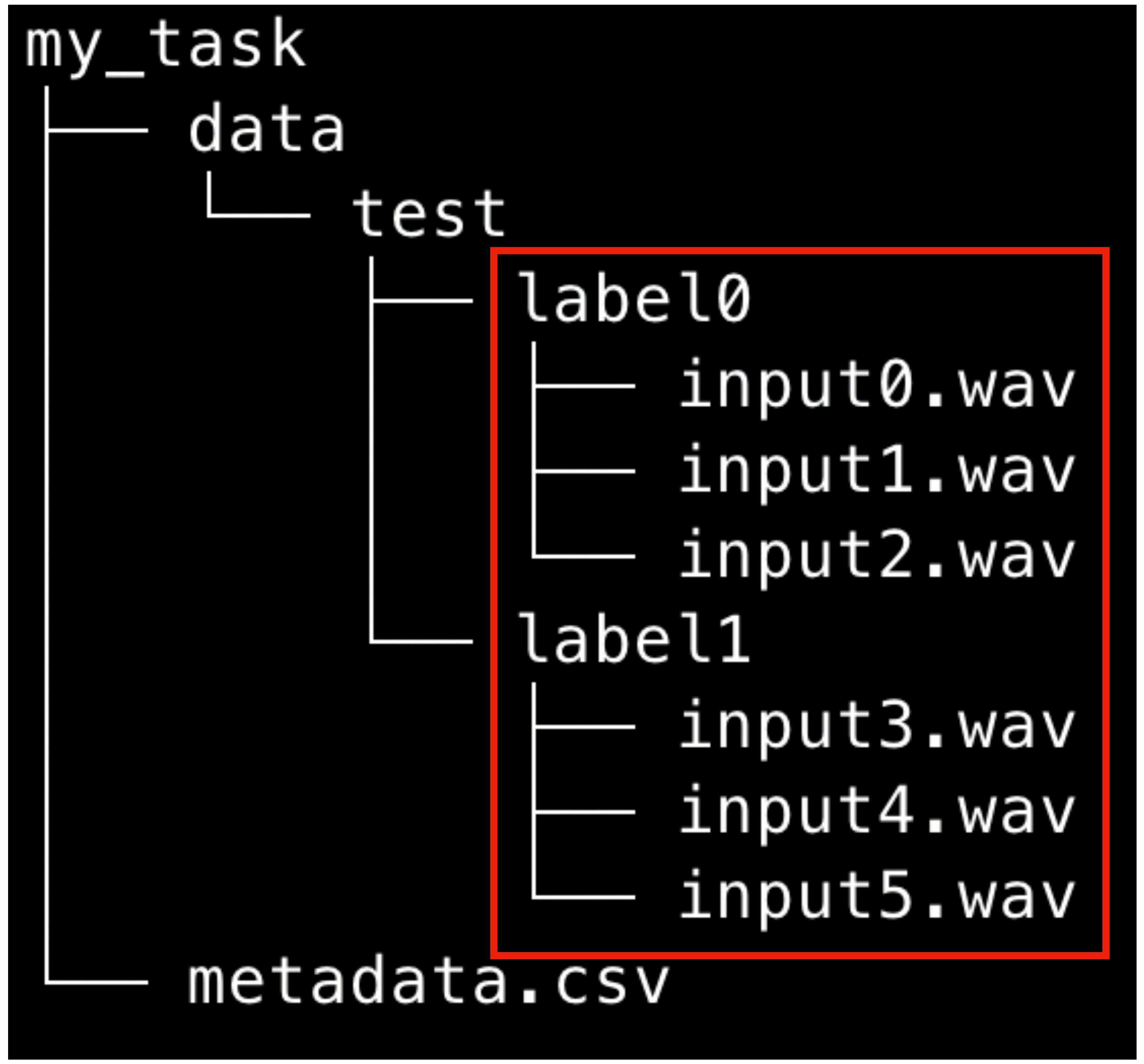
Huggingface Dataset Tutorial

Classification Task Version

Directory Structure

- Prepare your data in the following structure:



```
my_task
├── data
│   └── test
│       ├── label0
│       │   ├── input0.wav
│       │   ├── input1.wav
│       │   └── input2.wav
│       └── label1
│           ├── input3.wav
│           ├── input4.wav
│           └── input5.wav
└── metadata.csv
```

Arrange your data by their labels.

About Metadata

- Include at least four attributes (columns) in “metadata.csv”:
  - file_name - the file path relative to root (my_task).
  - file - The file name (do not need to include parent directory).
  - instruction - The text instruction specifying the task.
  - label - The corresponding label.
    - For classification tasks, please use **textual format**.

Metadata - Example

Attributes

Data 0
Data 1
Data 2
…

```
file_name,file,instruction,label
data/test/angry/b84f83d2_nohash_1.wav,b84f83d2_nohash_1.wav,this is instruction,angry
data/test/happy/3bc21161_nohash_0.wav,3bc21161_nohash_0.wav,this is instruction,happy
data/test/happy/dca6b373_nohash_0.wav,dca6b373_nohash_0.wav,this is instruction,happy
data/test/sad/6a861f21_nohash_0.wav,6a861f21_nohash_0.wav,this is instruction,sad
data/test/sad/e7117d00_nohash_0.wav,e7117d00_nohash_0.wav,this is instruction,sad
```

- We provide a Python script for generating metadata automatically.

Generating Metadata

Python Script

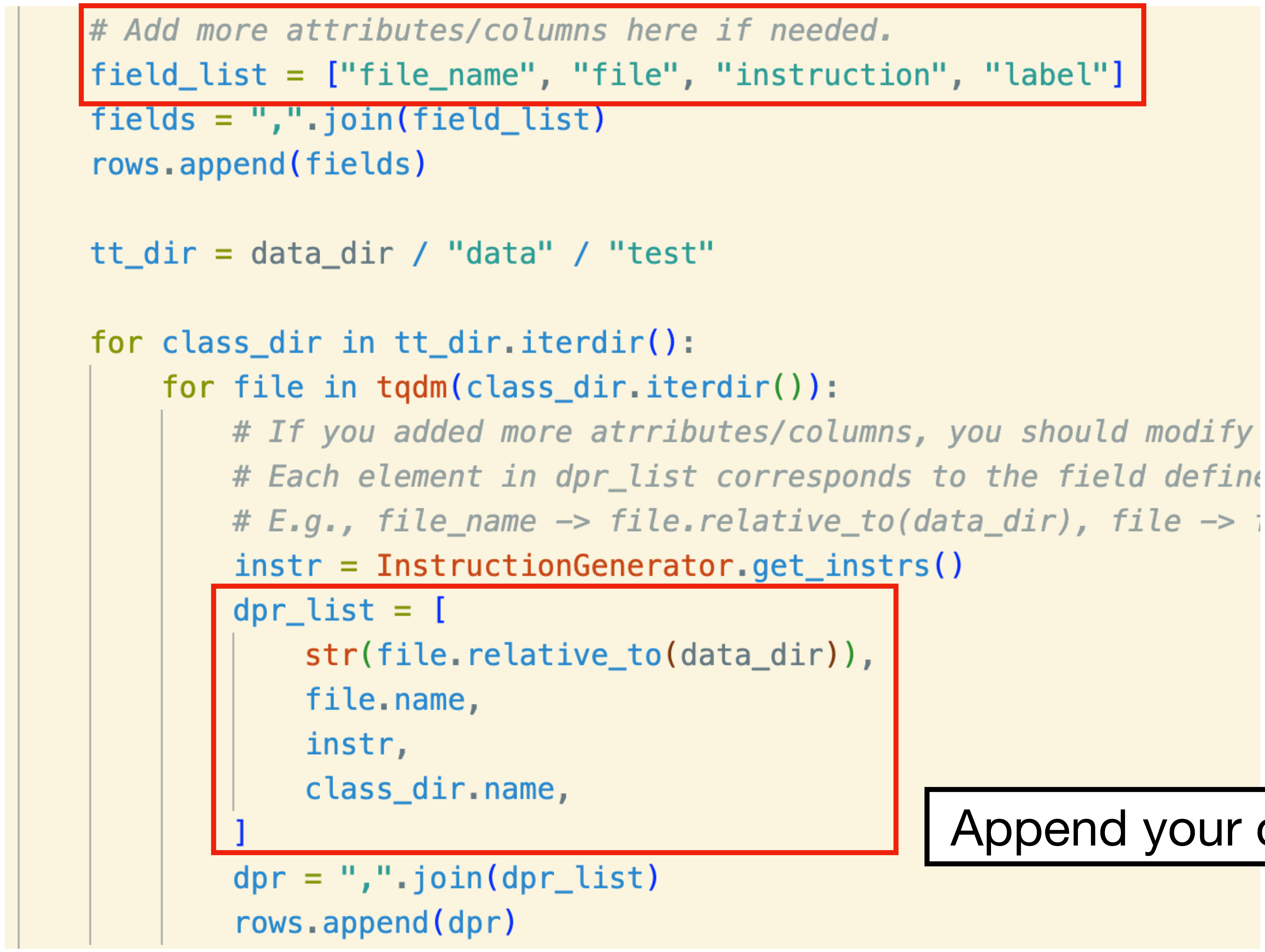
- Usage

```
python generate_metadata.py --data_dir DATA_DIR --seed SEED
```

  - DATA_DIR: The dataset directory (e.g., my_task).
  - SEED: The random seed for sampling instructions.

Generating Metadata

Adding Attributes

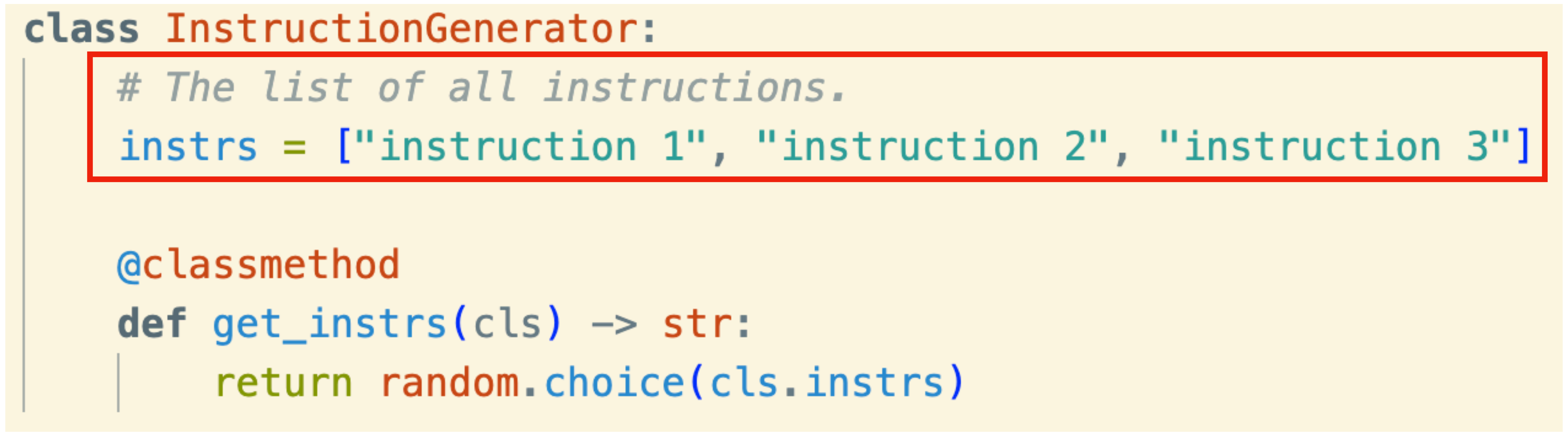


```
# Add more attributes/columns here if needed.
field_list = ["file_name", "file", "instruction", "label"]
fields = ",".join(field_list)
rows.append(fields)

tt_dir = data_dir / "data" / "test"

for class_dir in tt_dir.iterdir():
    for file in tqdm(class_dir.iterdir()):
        # If you added more atrributes/columns, you should modify
        # Each element in dpr_list corresponds to the field define
        # E.g., file_name -> file.relative_to(data_dir), file ->
        instr = InstructionGenerator.get_instrs()
        dpr_list = [
            str(file.relative_to(data_dir)),
            file.name,
            instr,
            class_dir.name,
        ]
        dpr = ",".join(dpr_list)
        rows.append(dpr)
```

Append your own attributes here.

Generating Metadata

Adding Instructions

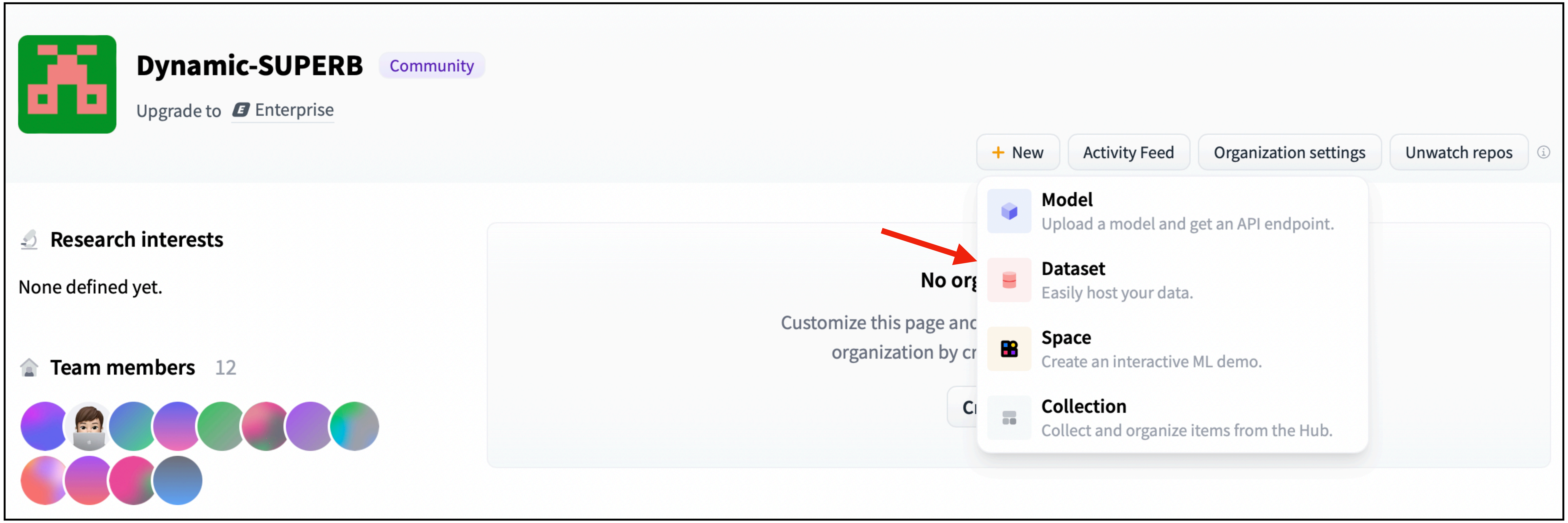


```python
class InstructionGenerator:
    # The list of all instructions.
    instrs = ["instruction 1", "instruction 2", "instruction 3"]

    @classmethod
    def get_instrs(cls) -> str:
        return random.choice(cls.instrs)
```

- You may add your own instructions in the class **InstructionGenerator**.
- All instructions are sampled uniformly, but you can modify it on your own.

Uploading Dataset

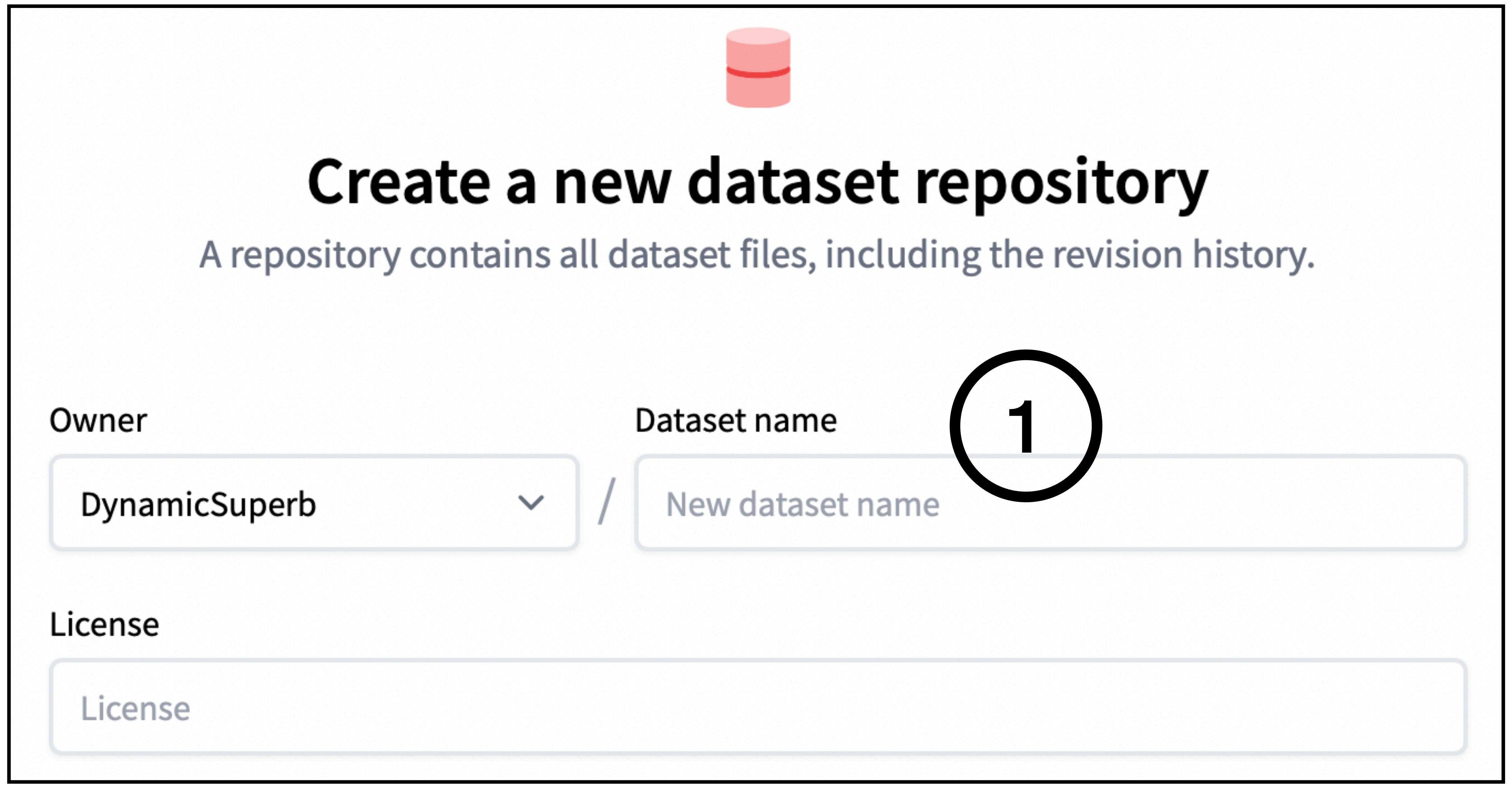
Creating New Repository on Huggingface



- Visit our Huggingface page via this link.
- Don't forget to **join the organization** to get the permission for doing this.

Uploading Dataset

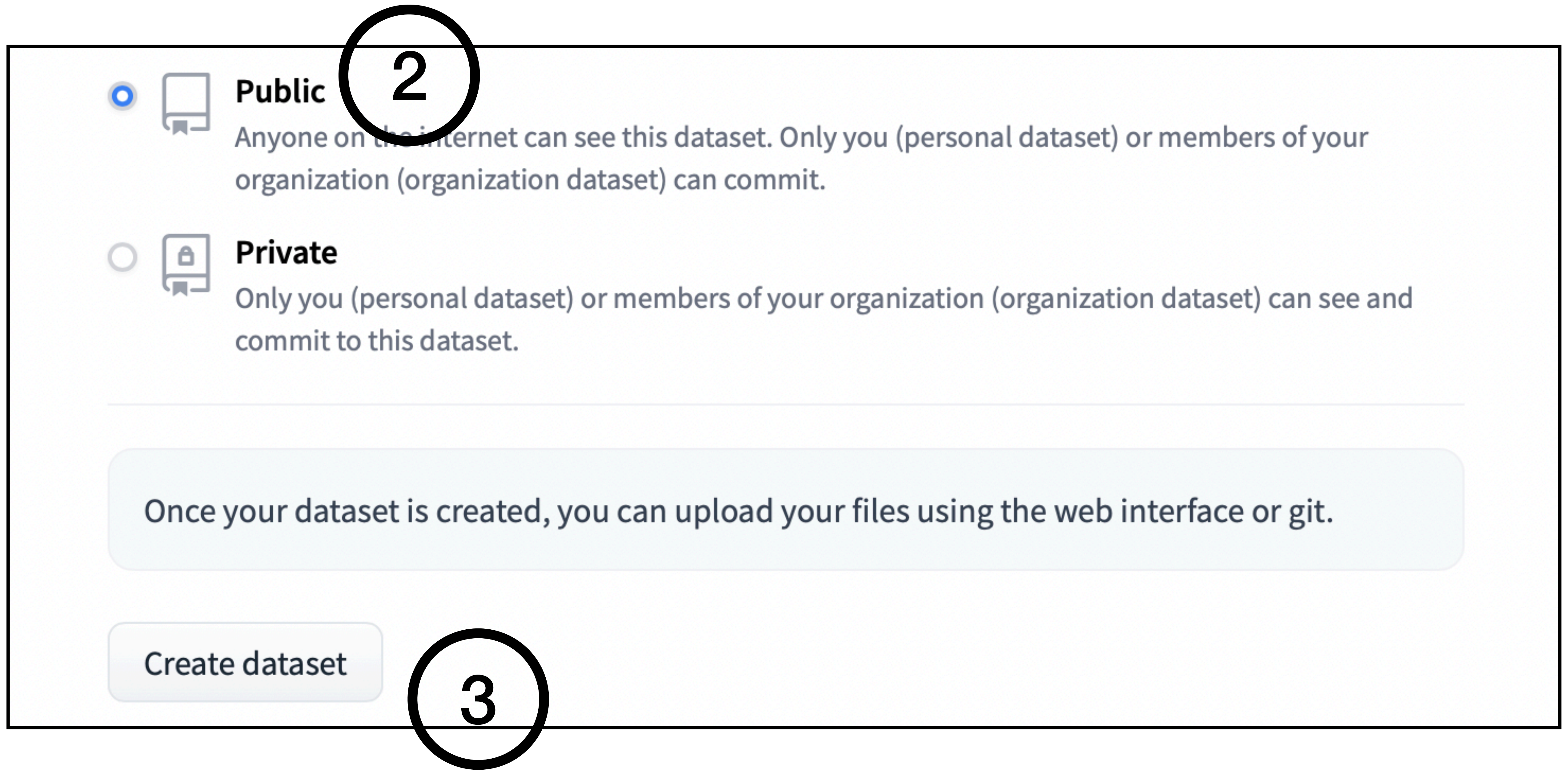
Creating New Repository on Huggingface



Create a new dataset repository

A repository contains all dataset files, including the revision history.

Owner	DynamicSuperb
Dataset name	New dataset name
License	License



[✓ Public]

Anyone on the internet can see this dataset. Only you (personal dataset) or members of your organization (organization dataset) can commit.

Private

Only you (personal dataset) or members of your organization (organization dataset) can see and commit to this dataset.

Once your dataset is created, you can upload your files using the web interface or git.

- **Make sure the data is allowed to be remixed and redistributed.**
- The name should follow our rules (please refer to the next page).

Uploading Dataset

Naming Rules

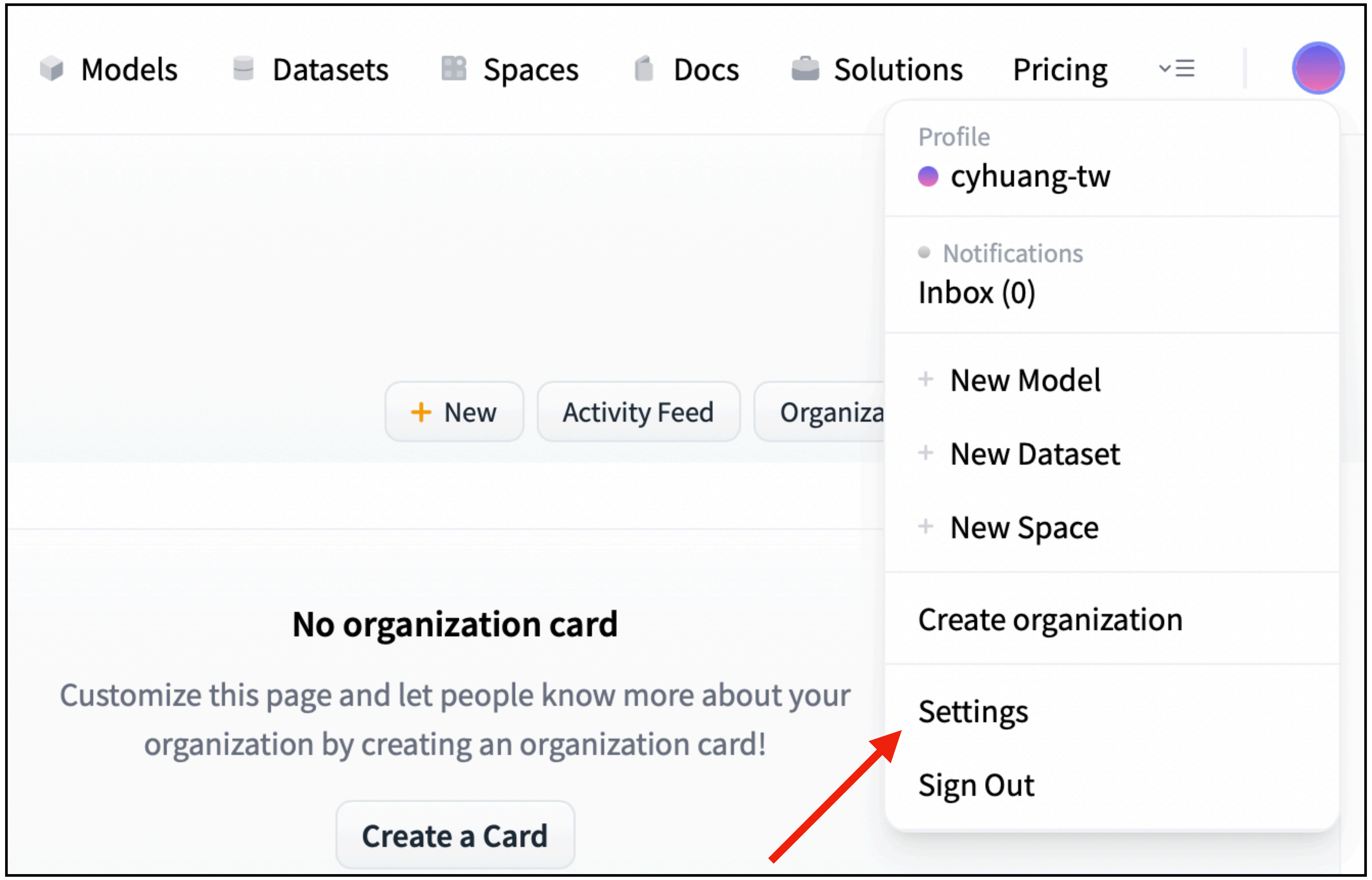
- The instance name should follow the format: **Task_Dataset**
- For task and dataset, use **CamelCase** (e.g., NoiseDetection, LibriSpeech).
- If you need to specify a specific subset of the dataset, use **hyphens** (-).
    - E.g., LibriSpeech-Test-Clean, LibriSpeech-Test-Other.
- If there are multiple datasets, connect them with the **underlines** (_).
    - E.g., VCTK_Wham, LibriSpeech-Test-Clean_LJSpeech.

Uploading Dataset

Login to Huggingface in Terminal

- pip install huggingface_hub
- huggingface-cli login
  - You will need to enter your **access token**.
  - If you don’t have any tokens, please refer to the next page for steps.

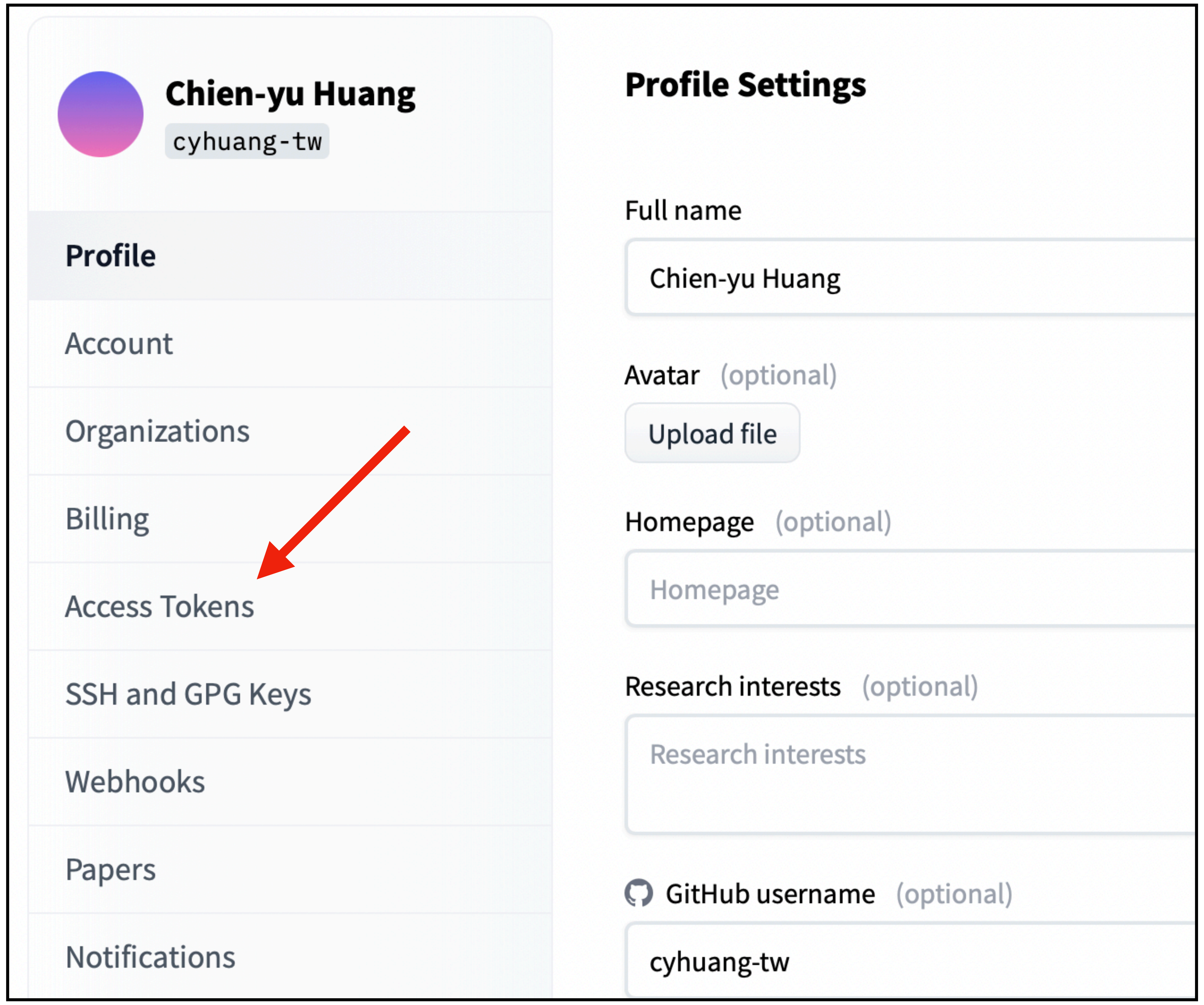
Retrieving Access Token



No organization card

Customize this page and let people know more about your organization by creating an organization card!

Retrieving Access Token


Chien-yu Huang
cyhuang-tw

Profile
Account
Organizations
Billing
Access Tokens
SSH and GPG Keys
Webhooks
Papers
Notifications

Profile Settings

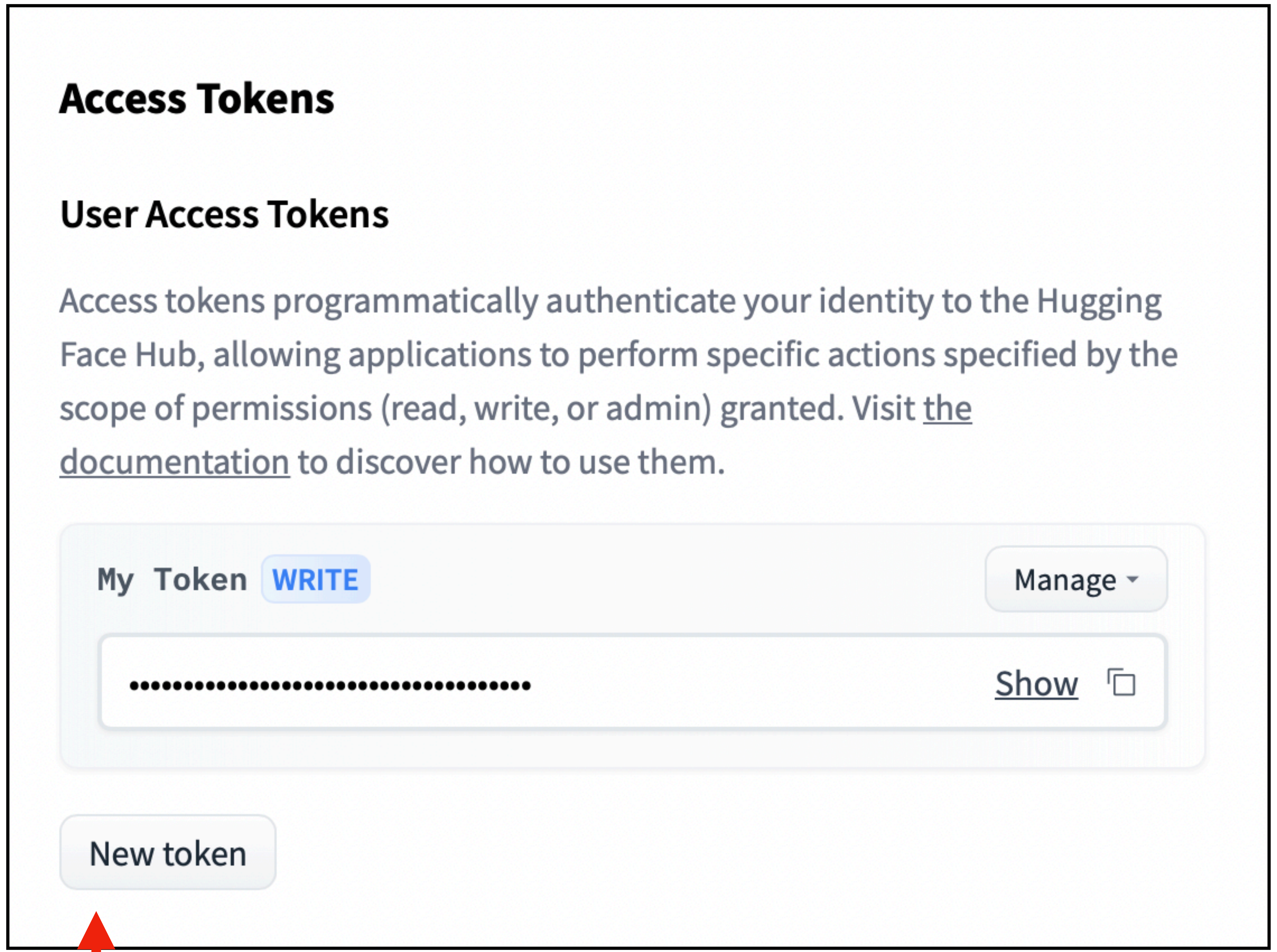
Full name: Chien-yu Huang

Avatar (optional)
Upload file

Homepage (optional)
Homepage

Research interests (optional)
Research interests

GitHub username (optional)
cyhuang-tw


Access Tokens

User Access Tokens

Access tokens programmatically authenticate your identity to the Hugging Face Hub, allowing applications to perform specific actions specified by the scope of permissions (read, write, or admin) granted. Visit the documentation to discover how to use them.

My Token WRITE
Manage
•••••••••••••••••••••••••••••••••••••• Show

New token

If you don't have any tokens yet, generate a new one.

Uploading Dataset

Python Script

- Usage

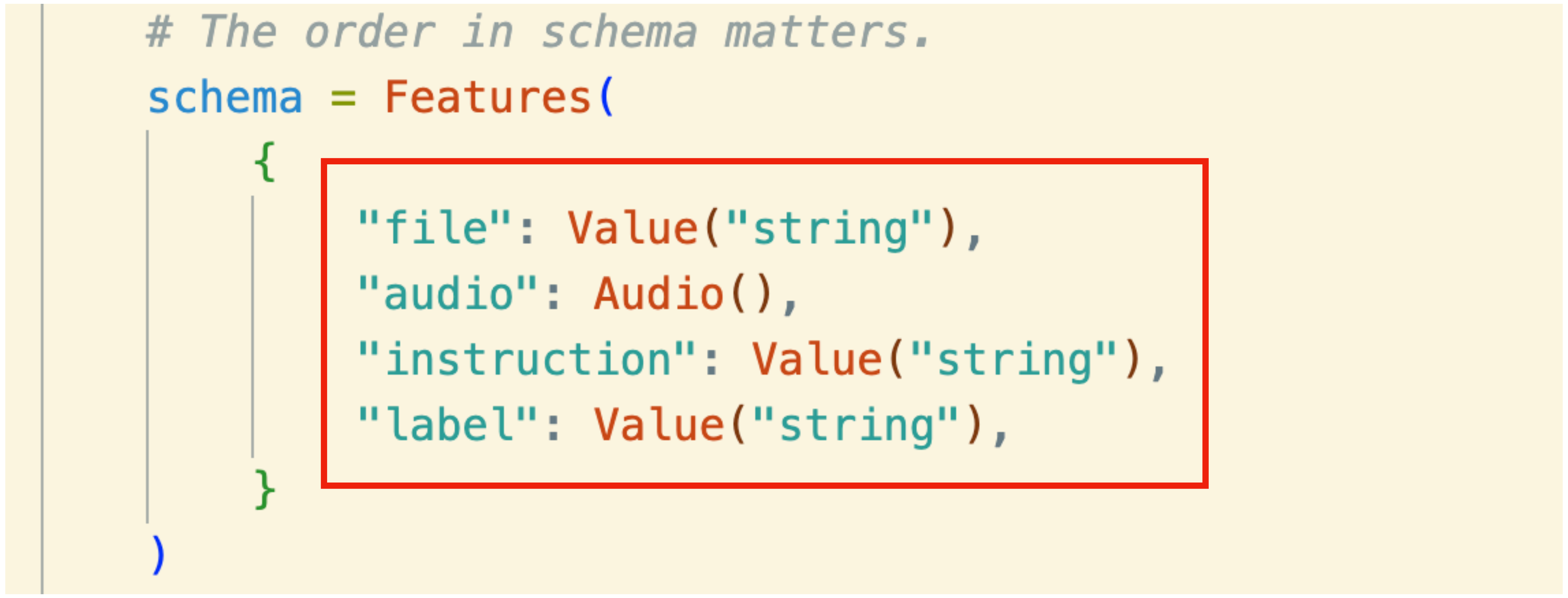
```
python upload_dataset.py --data_dir DATA_DIR --remote_path REMOTE_PATH
```

  - DATA_DIR: The task directory (e.g., my_task).
  - REMOTE_PATH: The repository on Huggingface (e.g., DynamicSuperb/MyTask).
- Modify some parts of code if you added some attributes (refer to the next page).

Uploading Dataset

Adding Custom Attributes

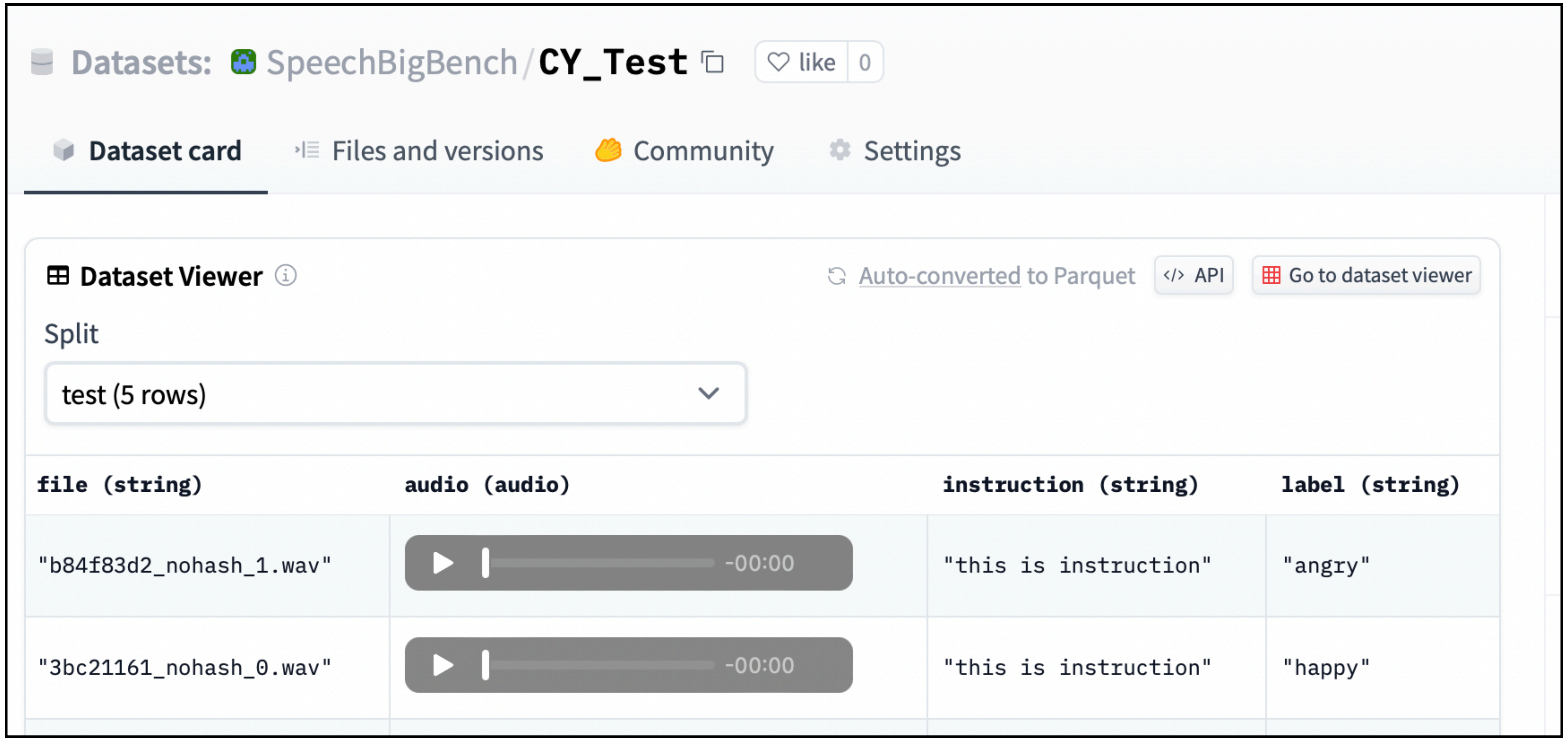
- Append attributes in "**schema**".
- Order matters! (Ensure **file** & **audio** be the first two attributes.)
- No need to include **file_name**.



```
# The order in schema matters.
schema = Features(
    {
        "file": Value("string"),
        "audio": Audio(),
        "instruction": Value("string"),
        "label": Value("string"),
    }
)
```

Check Status

- Don’t forget to check your data on Huggingface.



Datasets: SpeechBigBench / CY_Test

file (string)	audio (audio)	instruction (string)	label (string)
"b84f83d2_nohash_1.wav"	-00:00	"this is instruction"	"angry"
"3bc21161_nohash_0.wav"	-00:00	"this is instruction"	"happy"