Step 1: Activate Python Environment and Set API Key

```
(base) ➜  fine_tuning source myenv/bin/activate
```

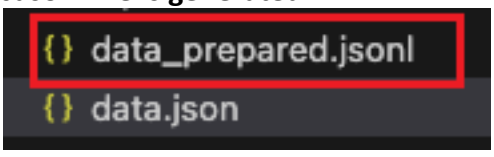
```
(myenv) (base) ➜  fine_tuning export OPENAI_API_KEY=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
```

Step 2: Create JSONL File

Step 3: Analyze and Prepare Data

```
(myenv) (base) ➜  fine_tuning openai tools fine_tunes.prepare_data -f data.json

Analyzing...

- Your file contains 3 prompt-completion pairs. In general, we recommend having at least a few hundred examples.
 We've found that performance tends to linearly increase for every doubling of the number of examples
- All prompts end with suffix `?`
- Your data does not contain a common ending at the end of your completions. Having a common ending string appen
ded to the end of the completion makes it clearer to the fine-tuned model where the completion should end. See h
ttps://platform.openai.com/docs/guides/fine-tuning/preparing-your-dataset for more detail and examples.
- The completion should start with a whitespace character (` `). This tends to produce better results due to the
 tokenization we use. See https://platform.openai.com/docs/guides/fine-tuning/preparing-your-dataset for more de
tails

Based on the analysis we will perform the following actions:
- [Recommended] Add a suffix ending `\n` to all completions [Y/n]: Y
- [Recommended] Add a whitespace character to the beginning of the completion [Y/n]: Y


Your data will be written to a new JSONL file. Proceed [Y/n]: Y

Wrote modified file to `data_prepared.jsonl`
Feel free to take a look!

Now use that file when fine-tuning:
> openai api fine_tunes.create -t "data_prepared.jsonl"

After you've fine-tuned a model, remember that your prompt has to end with the indicator string `?` for the mode
l to start generating completions, rather than continuing with the prompt. Make sure to include `stop=["\n"]` so
 that the generated texts ends at the expected place.
Once your model starts training, it'll approximately take 2.48 minutes to train a `curie` model, and less for `a
da` and `babbage`. Queue will approximately take half an hour per job ahead of you.
```

Jasonl file is generated:



Step 4: Fine-Tune the Model

```
$ openai api fine_tunes.create -t "data_prepared.jsonl" -m curie
```

```
Upload progress: 100%| | 417/417 [00:00<00:00, 639kit/s]
Uploaded file from data_prepared.jsonl: file-98FxK3uKP31TYP7B71FOULDa
Created fine-tune: ft-bqghwGxxcg7Z4RKV6cFmEYJq
Streaming events until fine-tuning is complete...

(Ctrl-C will interrupt the stream, but not cancel the fine-tune)
[2023-11-20 22:29:39] Created fine-tune: ft-bqghwGxxcg7Z4RKV6cFmEYJq
[2023-11-20 22:29:41] Fine-tune costs $0.00
[2023-11-20 22:29:42] Fine-tune enqueued. Queue number: 0
```

Step 5: List Fine-Tuned Models

```
$ openai api fine_tunes.list
```


```
{
  "object": "list",
  "data": [
    {
      "object": "fine-tune",
      "id": "ft-bqghwGxxcg7Z4RKV6cFmEYJq",
      "hyperparams": {
        "n_epochs": 4,
        "batch_size": 1,
        "prompt_loss_weight": 0.01,
        "learning_rate_multiplier": 0.1
      },
      "organization_id": "org-5Wp0wasioO9MFN8F2b7ki4ap",
      "model": "curie",
      "training_files": [
        {
          "object": "file",
          "id": "file-98FxK3uKP31TYP7B71FOULDa",
          "purpose": "fine-tune",
          "filename": "data_prepared.jsonl",
          "bytes": 417,
          "created_at": 1700548178,
          "status": "processed",
          "status_details": null
        }
      ],
      "validation_files": [],
      "result_files": [],
      "created_at": 1700548179,
      "updated_at": 1700548182,
      "status": "pending",
      "fine_tuned_model": null
```


Step 6: Resume Fine-Tuning

```
$ openai api fine_tunes.follow -i ft-eIt0cXubeWlMeLtlJ4Sh7Lml
```

```
[2023-11-20 22:39:09] Created fine-tune: ft-eIt0cXubeWlMeLtlJ4Sh7Lml
[2023-11-20 22:59:50] Fine-tune costs $0.00
[2023-11-20 22:59:50] Fine-tune enqueued. Queue number: 0
[2023-11-20 22:59:51] Fine-tune started
[2023-11-20 23:00:52] Completed epoch 1/4
[2023-11-20 23:00:52] Completed epoch 2/4
[2023-11-20 23:00:53] Completed epoch 3/4
[2023-11-20 23:00:54] Completed epoch 4/4
[2023-11-20 23:01:13] Uploaded model: curie:ft-personal-2023-11-21-07-01-13
[2023-11-20 23:01:14] Uploaded result file: file-mro92uGsLRAX5QCkM1L6yxly
[2023-11-20 23:01:14] Fine-tune succeeded

Job complete! Status: succeeded 🎉
Try out your fine-tuned model:

openai api completions.create -m curie:ft-personal-2023-11-21-07-01-13 -p <YOUR_PROMPT>
```

Step7:Use the Fine-Tuned Model

```
$ export FINE_TUNED_MODEL="<FINE_TUNED_MODEL>"
$ openai api completions.create -m $FINE_TUNED_MODEL -p <YOUR_PROMPT>
```

Step 8: Use python

```python
import openai
import os
from dotenv import load_dotenv, find_dotenv
_ = load_dotenv(find_dotenv()) # read local .env file

openai.api_key  = os.environ['OPENAI_API_KEY']

FINE_TUNED_MODEL="curie:ft-personal-2023-11-21-07-01-13"
YOUR_PROMPT="What is the remote for?"

response = openai.Completion.create(
    model=FINE_TUNED_MODEL,
    prompt=YOUR_PROMPT
    # additional parameters
    # temperature,
    # frequency_penalty,
    # presence_penalty
    # ..etc
)
print(response)
```

Result:

```
└─$ python3 test.py
{
  "id": "cmpl-8NFFX0TayawYZZIw48FjRec153EVr",
  "object": "text_completion",
  "created": 1700550855,
  "model": "curie:ft-personal-2023-11-21-07-01-13",
  "choices": [
    {
      "text": "\n\n\\u003cp\\u003eRCA plugs into the wall",
      "index": 0,
      "logprobs": null,
      "finish_reason": "length"
    }
  ],
  "usage": {
    "prompt_tokens": 6,
    "completion_tokens": 16,
    "total_tokens": 22
  }
}
```

Step 9：Analyze Fine-Tuned Model

```
┌──(sharoncao0920㉿Sharon)-[~/Desktop/Fine-tune]
└─$     openai api fine_tunes.results -i ft-eIt0cXubeWlMeLtlJ4Sh7Lml
step,elapsed_tokens,elapsed_examples,training_loss,training_sequence_accuracy,training_token_accuracy
1,25,1,1.6863485659162203,0.0,0.36363636363636365
2,58,2,1.4617992158653215,0.0,0.55
3,91,3,1.7533674666797743,0.0,0.23529411764705882
4,124,4,1.312146593441721,0.0,0.55
5,157,5,1.601872600489296,0.0,0.29411764705882354
6,182,6,1.434608019342025,0.0,0.45454545454545453
7,207,7,1.3744844387223323,0.0,0.45454545454545453
8,240,8,1.065065179914236,0.0,0.65
9,273,9,1.3830768561922013,0.0,0.35294117647058826
10,306,10,1.3652678412059323,0.0,0.35294117647058826
11,339,11,1.0061288996529765,0.0,0.65
12,364,12,1.2389785048365591,0.0,0.45454545454545453
13,397,13,1.0058308023260907,0.0,0.65
```