KMU KOOKMIN UNIVERSITY

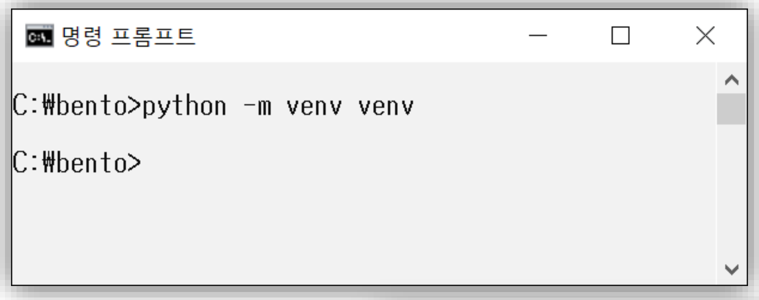
BentoML

BentoML
Inference Platform for Building Fast and Scalable AI Systems
BentoML

소프트웨어융합대학원
진혜진

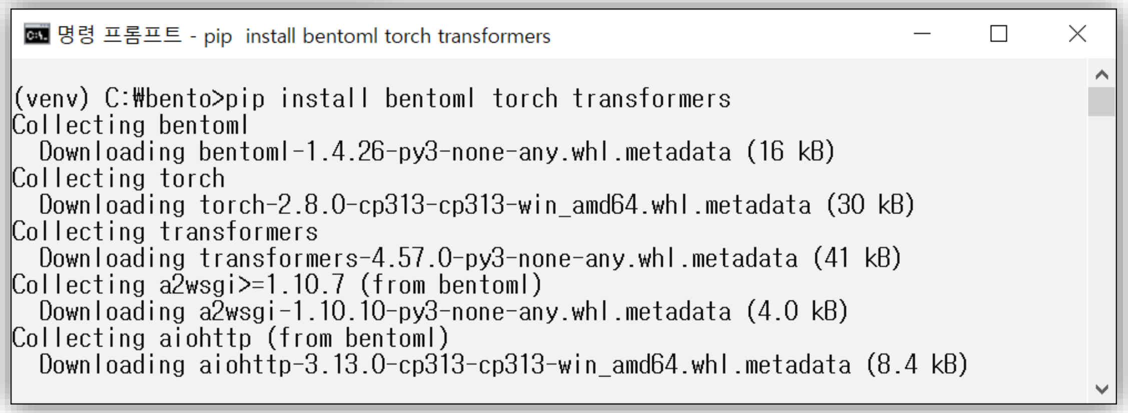
실습(Hugging Face Transformer 모델)

- 가상환경 활성화
  - python -m venv venv
  - venv\Scripts\activate



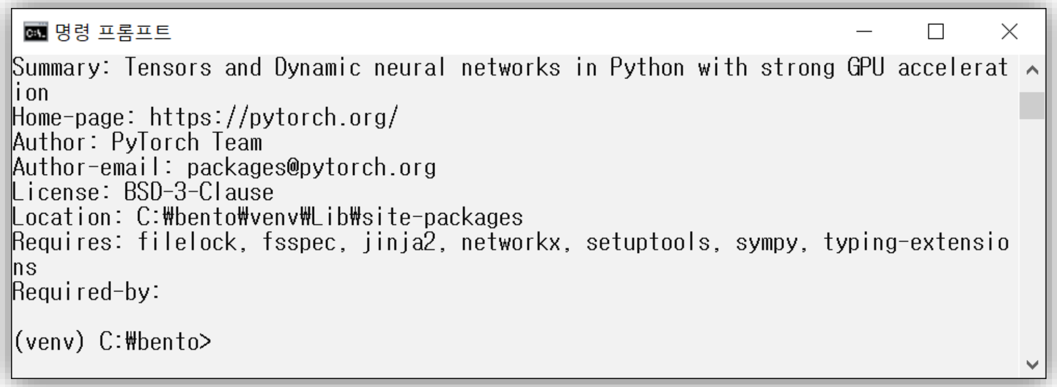
```
C:₩bento>python -m venv venv

C:₩bento>
```

- 종속성 설치
  - pip install bentoml torch transformers



```
(venv) C:₩bento>pip install bentoml torch transformers
Collecting bentoml
  Downloading bentoml-1.4.26-py3-none-any.whl.metadata (16 kB)
Collecting torch
  Downloading torch-2.8.0-cp313-cp313-win_amd64.whl.metadata (30 kB)
Collecting transformers
  Downloading transformers-4.57.0-py3-none-any.whl.metadata (41 kB)
Collecting a2wsgi>=1.10.7 (from bentoml)
  Downloading a2wsgi-1.10.10-py3-none-any.whl.metadata (4.0 kB)
Collecting aiohttp (from bentoml)
  Downloading aiohttp-3.13.0-cp313-cp313-win_amd64.whl.metadata (8.4 kB)
```

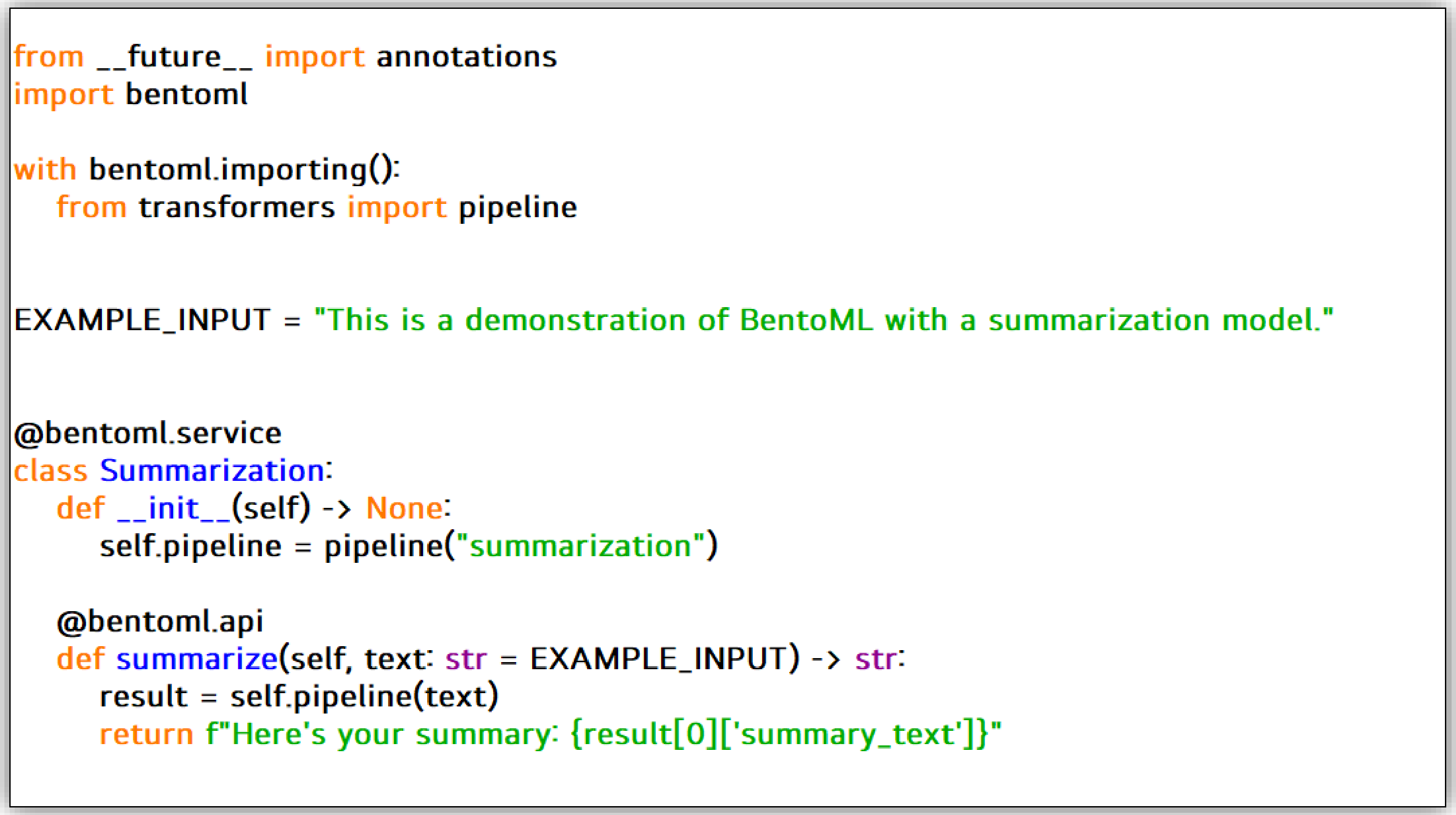
- 설치 확인
  - pip show bentoml
  - pip show torch
  - pip show transformers



```
Summary: Tensors and Dynamic neural networks in Python with strong GPU accelerat
ion
Home-page: https://pytorch.org/
Author: PyTorch Team
Author-email: packages@pytorch.org
License: BSD-3-Clause
Location: C:₩bento₩venv₩Lib₩site-packages
Requires: filelock, fsspec, jinja2, networkx, setuptools, sympy, typing-extensio
ns
Required-by:

(venv) C:₩bento>
```

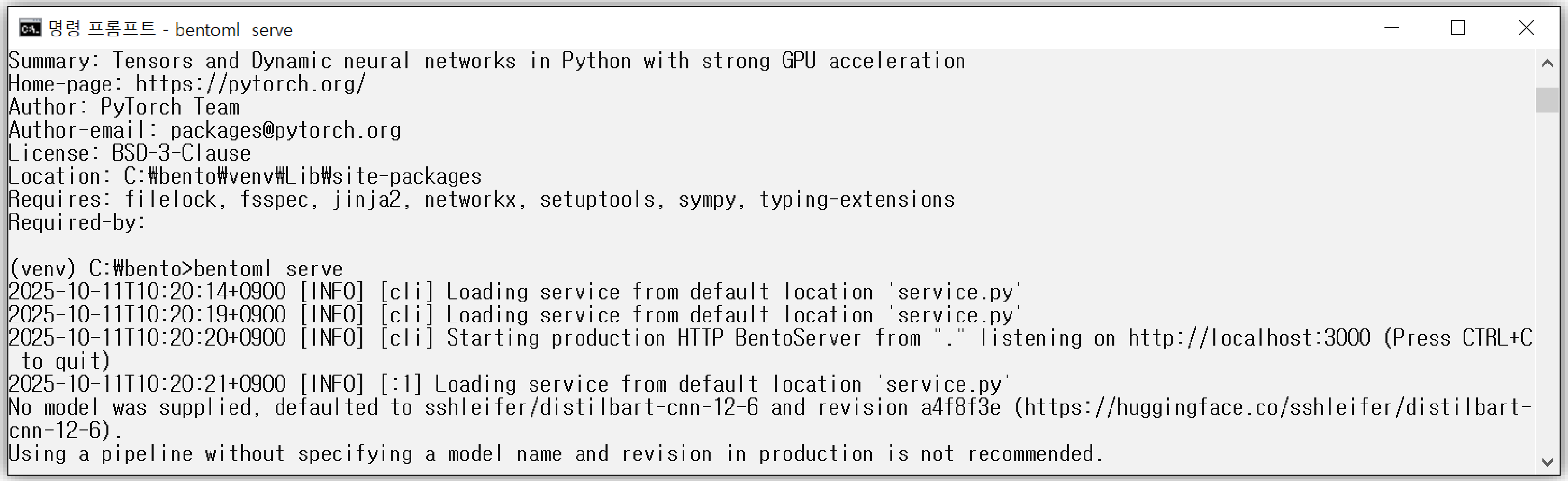
▪ 서비스(API) 만들기

- service.py

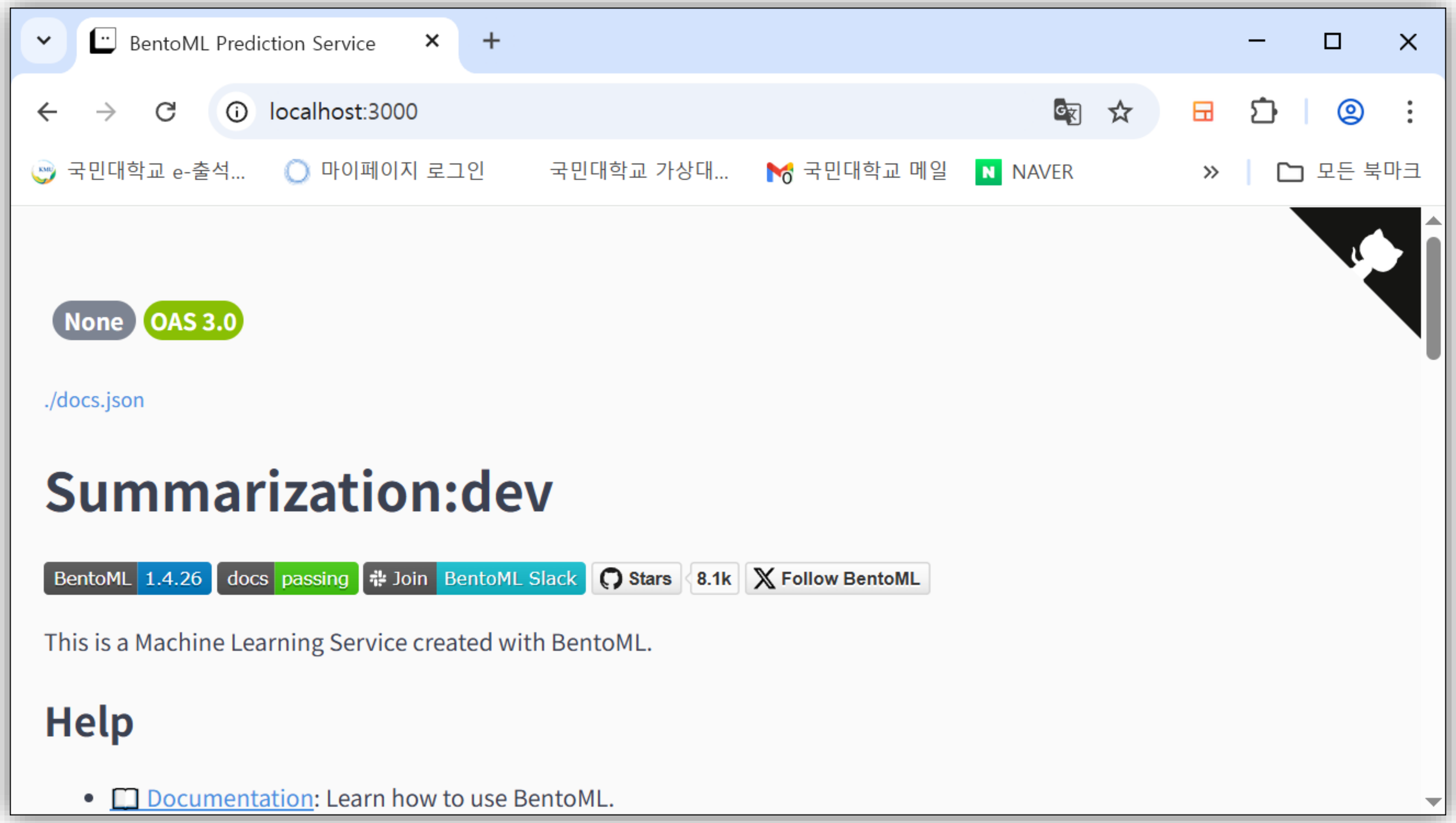


```
from __future__ import annotations
import bentoml

with bentoml.importing():
    from transformers import pipeline


EXAMPLE_INPUT = "This is a demonstration of BentoML with a summarization model."


@bentoml.service
class Summarization:
    def __init__(self) -> None:
        self.pipeline = pipeline("summarization")

    @bentoml.api
    def summarize(self, text: str = EXAMPLE_INPUT) -> str:
        result = self.pipeline(text)
        return f"Here's your summary: {result[0]['summary_text']}"
```

- 서버 실행
  - bentoml serve
- http://localhost:3000



```
Summary: Tensors and Dynamic neural networks in Python with strong GPU acceleration
Home-page: https://pytorch.org/
Author: PyTorch Team
Author-email: packages@pytorch.org
License: BSD-3-Clause
Location: C:₩bento₩venv₩Lib₩site-packages
Requires: filelock, fsspec, jinja2, networkx, setuptools, sympy, typing-extensions
Required-by:

(venv) C:₩bento>bentoml serve
2025-10-11T10:20:14+0900 [INFO] [cli] Loading service from default location 'service.py'
2025-10-11T10:20:19+0900 [INFO] [cli] Loading service from default location 'service.py'
2025-10-11T10:20:20+0900 [INFO] [cli] Starting production HTTP BentoServer from "." listening on http://localhost:3000 (Press CTRL+C
 to quit)
2025-10-11T10:20:21+0900 [INFO] [:1] Loading service from default location 'service.py'
No model was supplied, defaulted to sshleifer/distilbart-cnn-12-6 and revision a4f8f3e (https://huggingface.co/sshleifer/distilbart-
cnn-12-6).
Using a pipeline without specifying a model name and revision in production is not recommended.
```



None OAS 3.0

./docs.json

Summarization:dev

BentoML 1.4.26 | docs passing | Join BentoML Slack | Stars 8.1k | Follow BentoML

This is a Machine Learning Service created with BentoML.

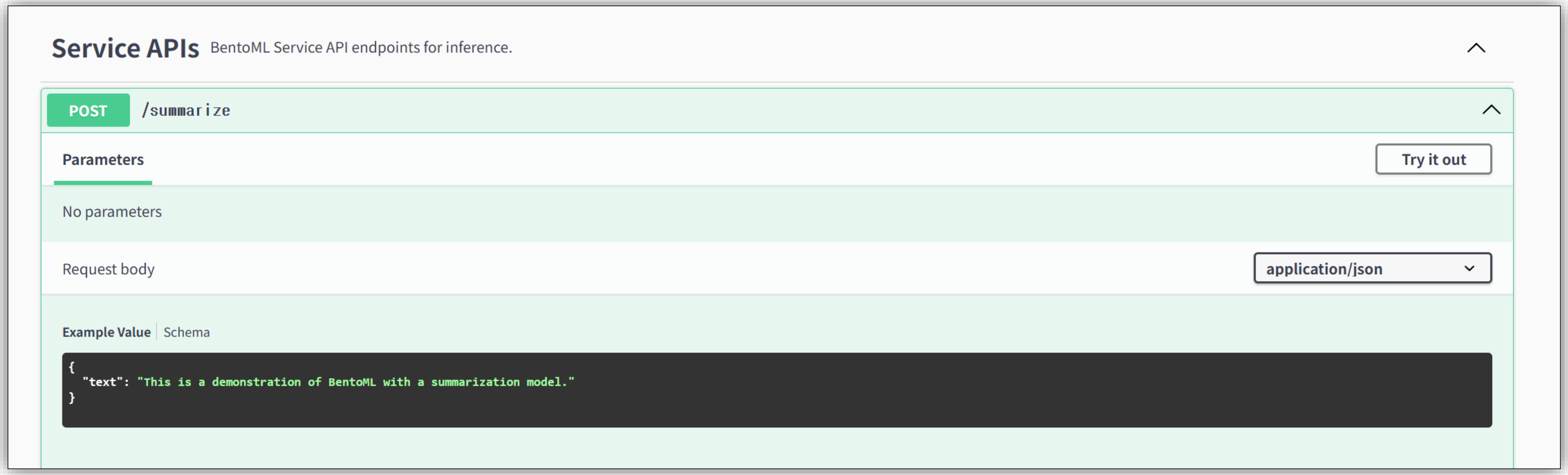
Help

- Documentation: Learn how to use BentoML.

실습

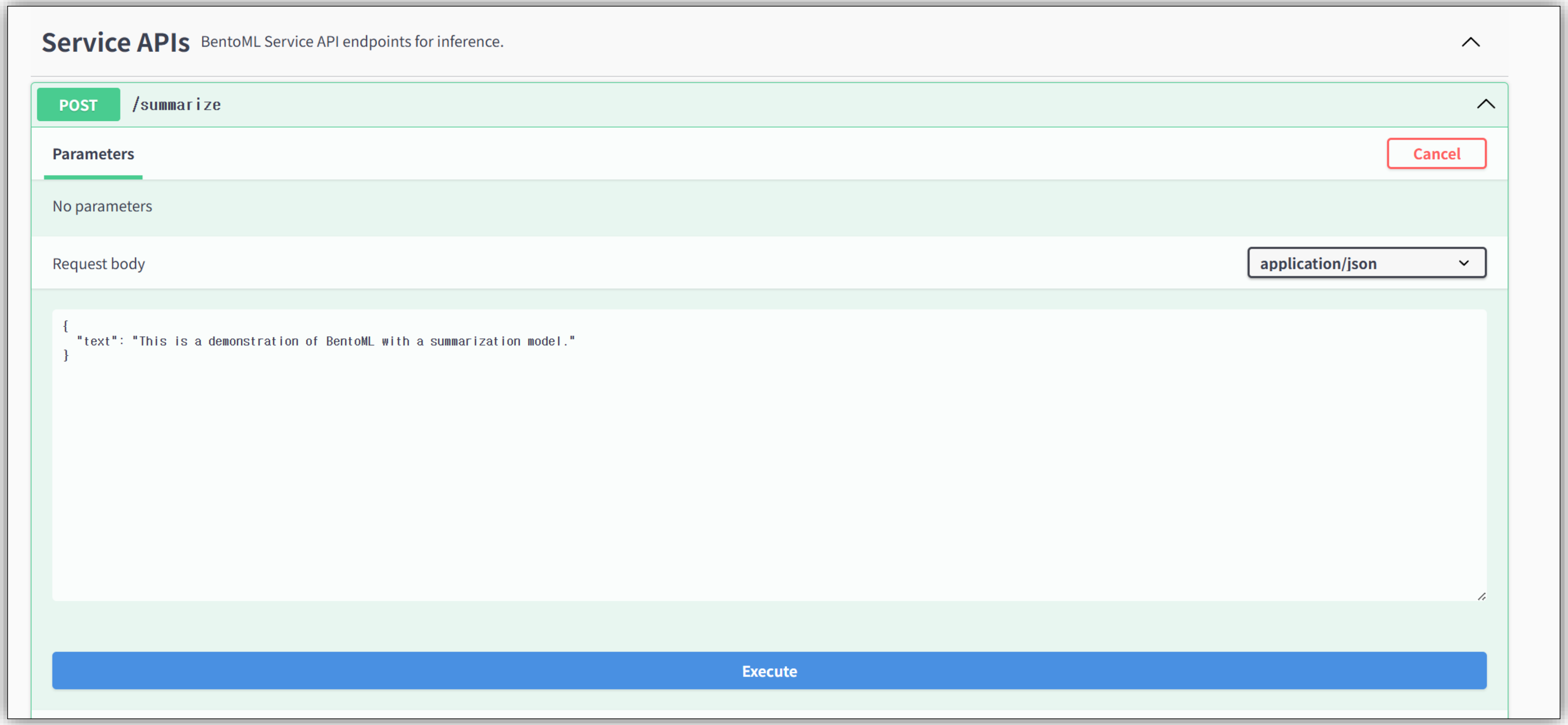
요청 보내기, 응답 확인(Swagger UI)

- http://localhost:3000
- Service APIs - summarize - Try it out



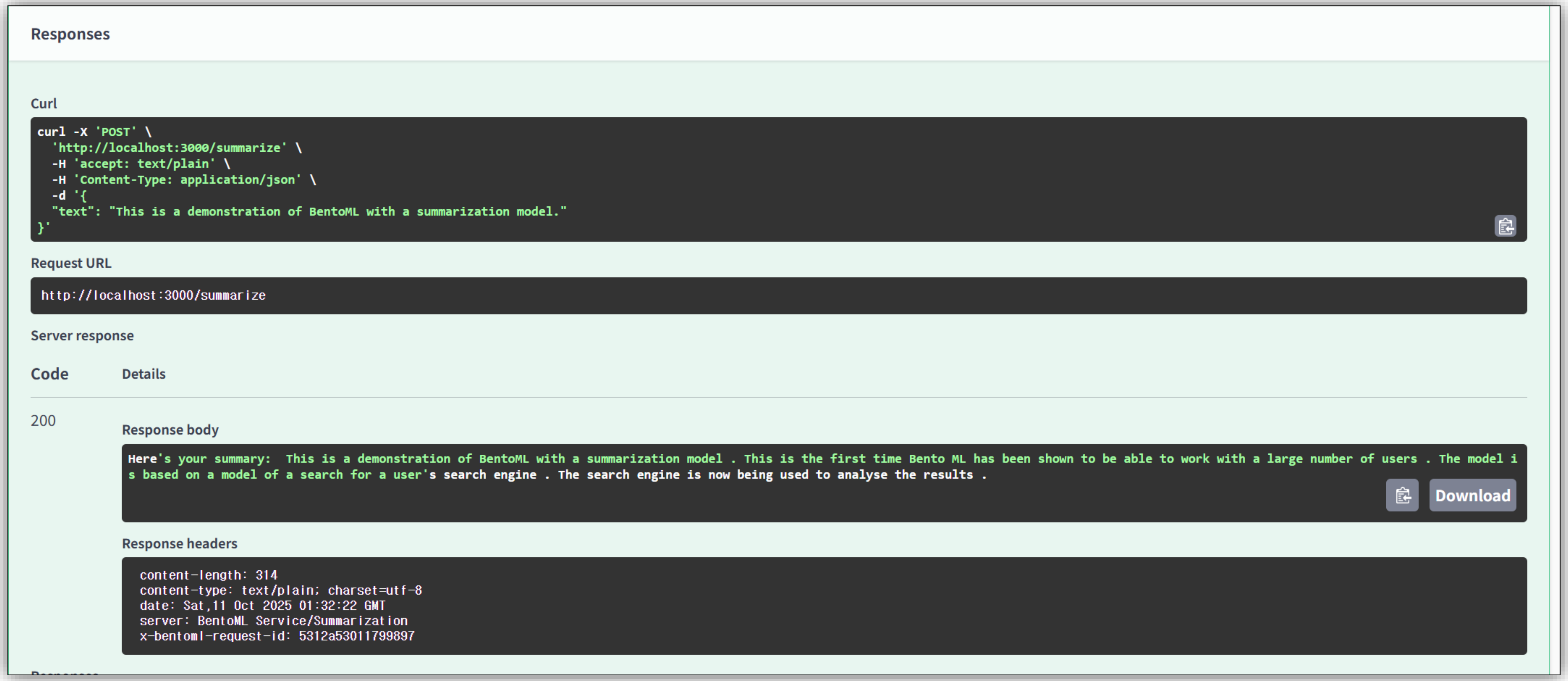
```
{
  "text": "This is a demonstration of BentoML with a summarization model."
}
```

실습



```
{
  "text": "This is a demonstration of BentoML with a summarization model."
}
```

실습



Responses

Curl

```
curl -X 'POST' \
  'http://localhost:3000/summarize' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "text": "This is a demonstration of BentoML with a summarization model."
}'
```

Request URL

```
http://localhost:3000/summarize
```

Server response

Code Details

200

Response body

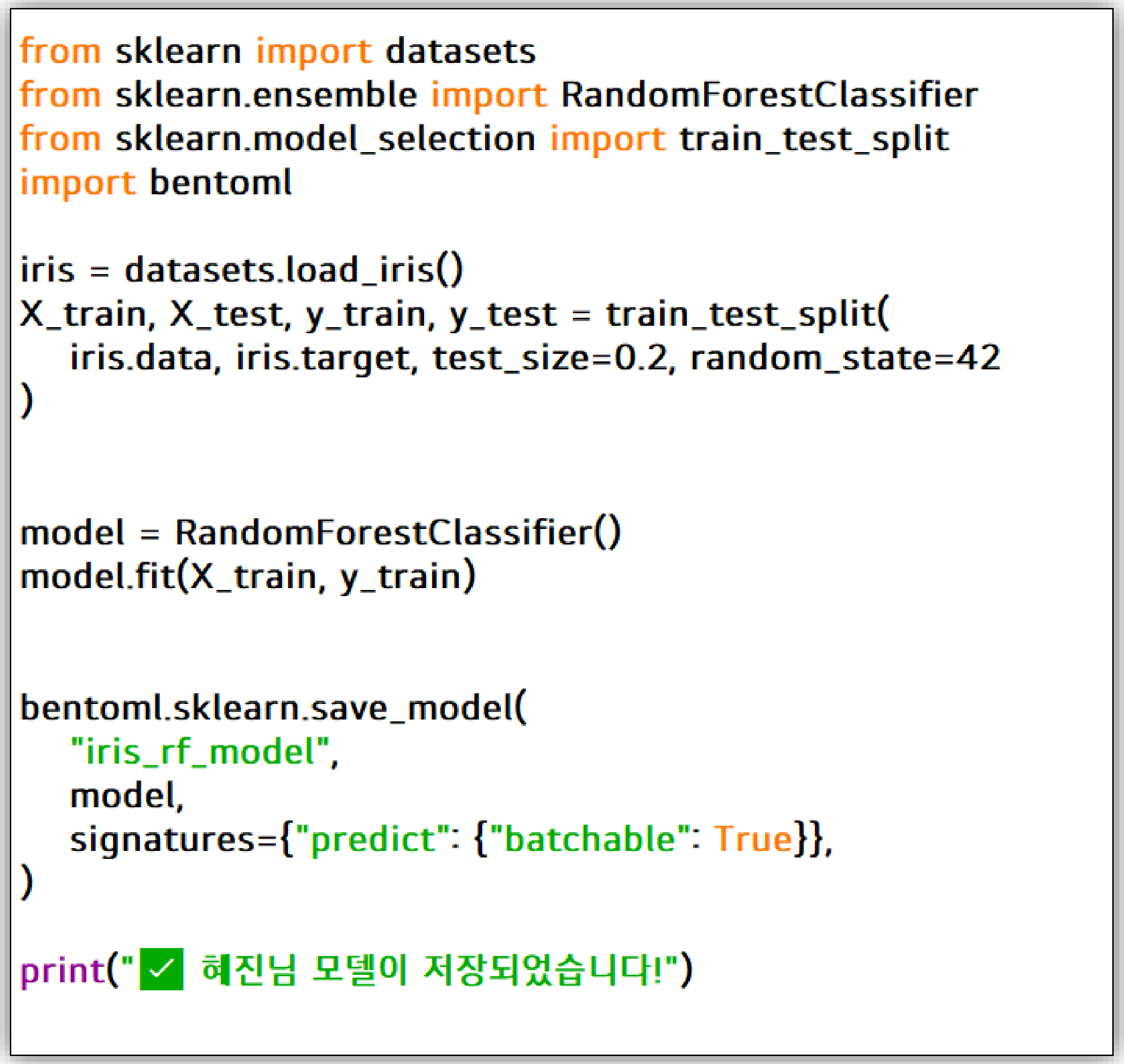
```
Here's your summary:  This is a demonstration of BentoML with a summarization model . This is the first time Bento ML has been shown to be able to work with a large number of users . The model i
s based on a model of a search for a user's search engine . The search engine is now being used to analyse the results .
```

Response headers

```
 content-length: 314
 content-type: text/plain; charset=utf-8
 date: Sat,11 Oct 2025 01:32:22 GMT
 server: BentoML Service/Summarization
 x-bentoml-request-id: 5312a53011799897
```

실습(scikit-learn Iris 분류 모델)

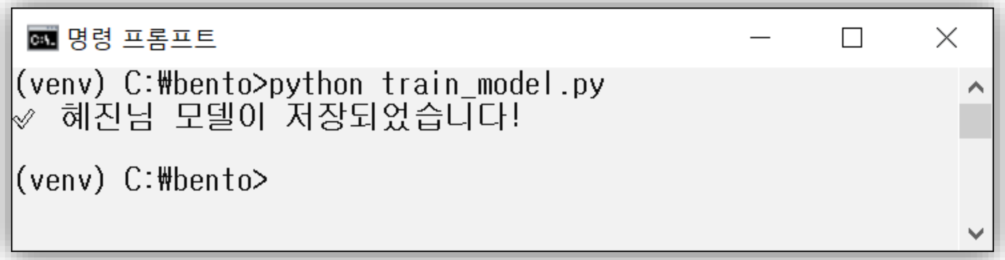
- train_model.py



```python
from sklearn import datasets
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
import bentoml

iris = datasets.load_iris()
X_train, X_test, y_train, y_test = train_test_split(
    iris.data, iris.target, test_size=0.2, random_state=42
)


model = RandomForestClassifier()
model.fit(X_train, y_train)


bentoml.sklearn.save_model(
    "iris_rf_model",
    model,
    signatures={"predict": {"batchable": True}},
)

print("✅ 혜진님 모델이 저장되었습니다!")
```

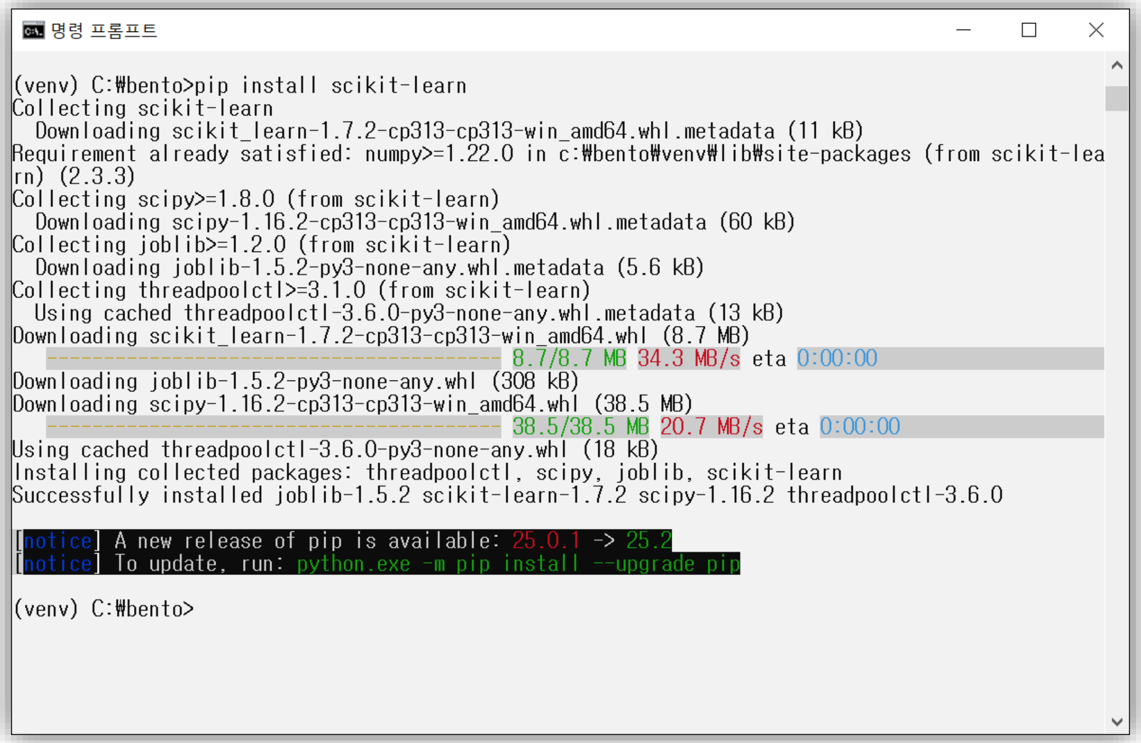
실습

- 실행
  - python train_model.py



```
(venv) C:₩bento>python train_model.py
✓ 혜진님 모델이 저장되었습니다!

(venv) C:₩bento>
```

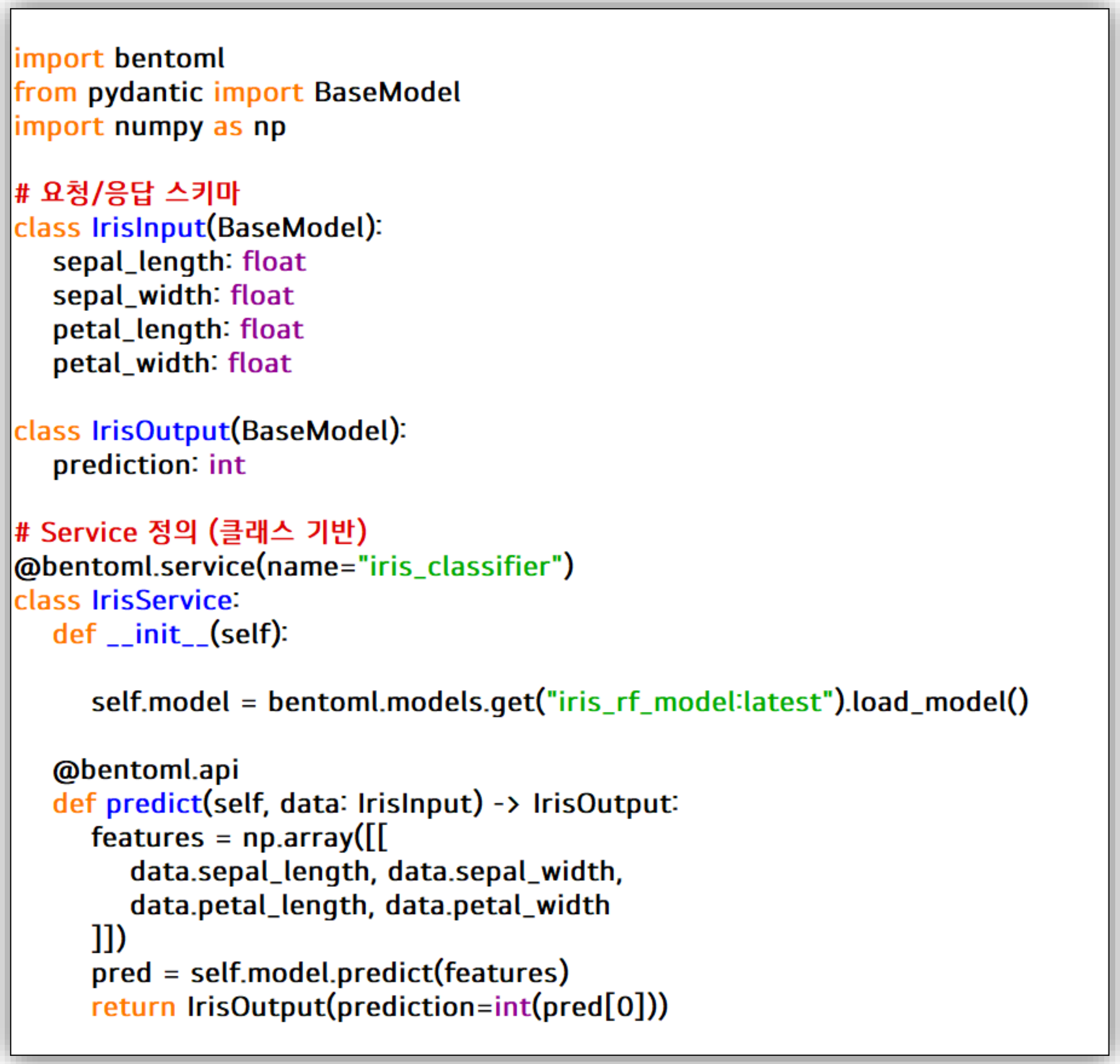
  - pip install scikit-learn



```
(venv) C:₩bento>pip install scikit-learn
Collecting scikit-learn
  Downloading scikit_learn-1.7.2-cp313-cp313-win_amd64.whl.metadata (11 kB)
Requirement already satisfied: numpy>=1.22.0 in c:₩bento₩venv₩lib₩site-packages (from scikit-lea
rn) (2.3.3)
Collecting scipy>=1.8.0 (from scikit-learn)
  Downloading scipy-1.16.2-cp313-cp313-win_amd64.whl.metadata (60 kB)
Collecting joblib>=1.2.0 (from scikit-learn)
  Downloading joblib-1.5.2-py3-none-any.whl.metadata (5.6 kB)
Collecting threadpoolctl>=3.1.0 (from scikit-learn)
  Using cached threadpoolctl-3.6.0-py3-none-any.whl.metadata (13 kB)
Downloading scikit_learn-1.7.2-cp313-cp313-win_amd64.whl (8.7 MB)
   ---------------------------------------- 8.7/8.7 MB 34.3 MB/s eta 0:00:00
Downloading joblib-1.5.2-py3-none-any.whl (308 kB)
Downloading scipy-1.16.2-cp313-cp313-win_amd64.whl (38.5 MB)
   ---------------------------------------- 38.5/38.5 MB 20.7 MB/s eta 0:00:00
Using cached threadpoolctl-3.6.0-py3-none-any.whl (18 kB)
Installing collected packages: threadpoolctl, scipy, joblib, scikit-learn
Successfully installed joblib-1.5.2 scikit-learn-1.7.2 scipy-1.16.2 threadpoolctl-3.6.0

[notice] A new release of pip is available: 25.0.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip

(venv) C:₩bento>
```

실습

- 서비스(API) 만들기

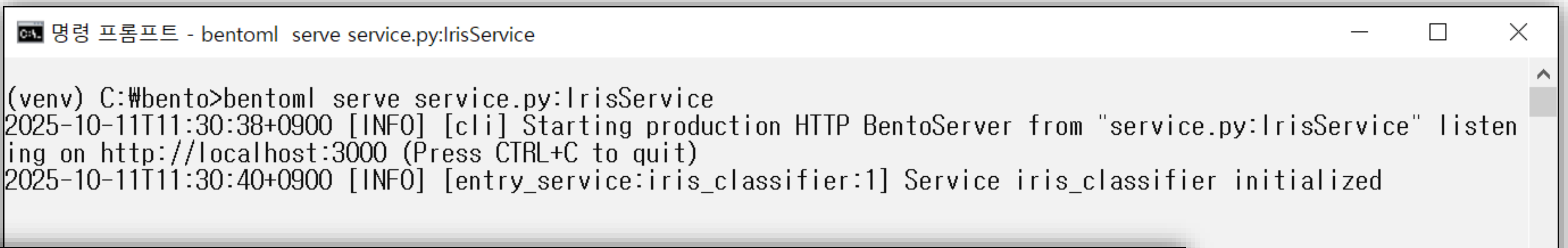


```
import bentoml
from pydantic import BaseModel
import numpy as np

# 요청/응답 스키마
class IrisInput(BaseModel):
    sepal_length: float
    sepal_width: float
    petal_length: float
    petal_width: float

class IrisOutput(BaseModel):
    prediction: int

# Service 정의 (클래스 기반)
@bentoml.service(name="iris_classifier")
class IrisService:
    def __init__(self):

        self.model = bentoml.models.get("iris_rf_model:latest").load_model()

    @bentoml.api
    def predict(self, data: IrisInput) -> IrisOutput:
        features = np.array([[
            data.sepal_length, data.sepal_width,
            data.petal_length, data.petal_width
        ]])
        pred = self.model.predict(features)
        return IrisOutput(prediction=int(pred[0]))
```

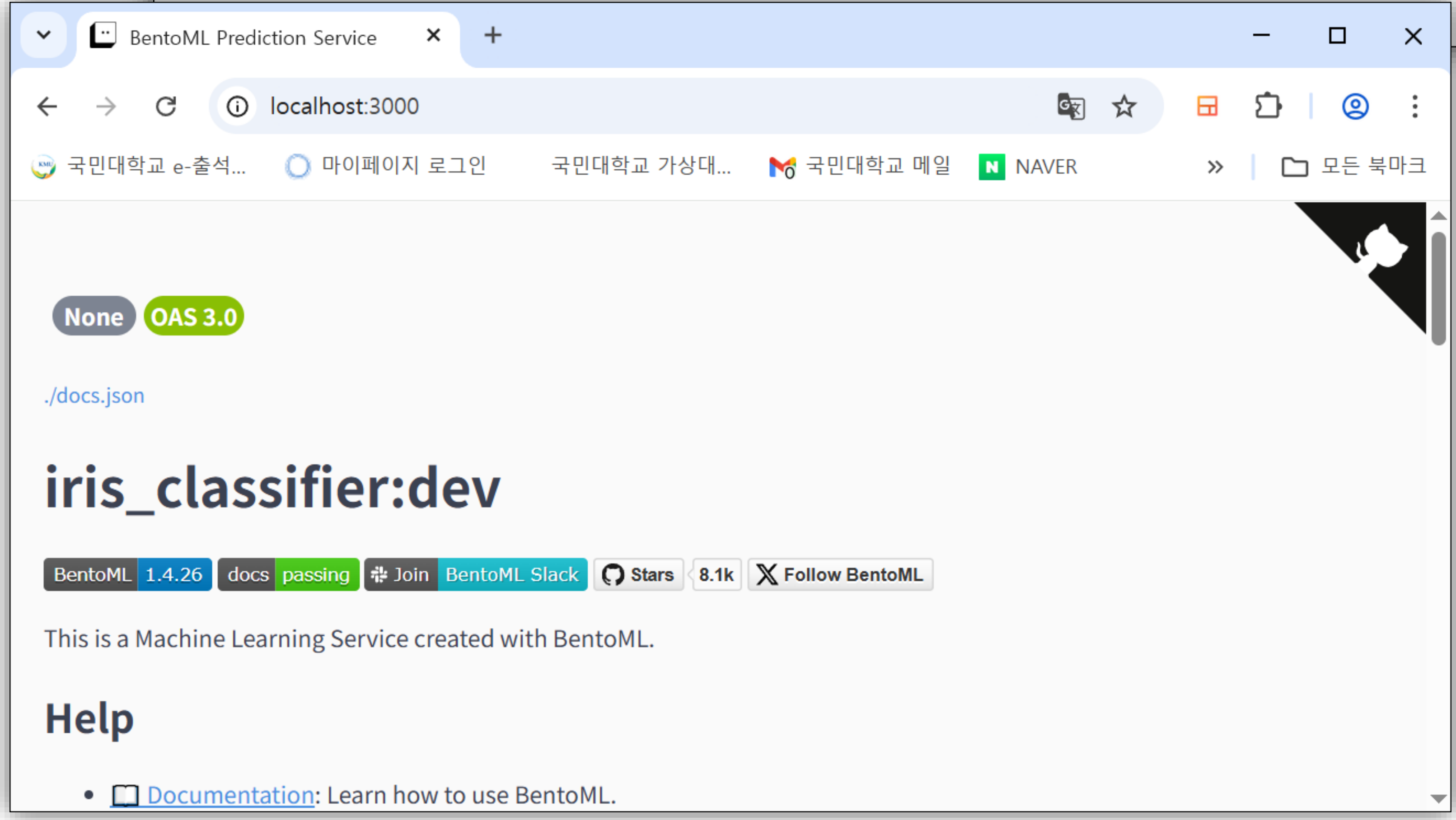
실습

KOOKMIN UNIVERSITY

■ 서비스 실행



```
(venv) C:₩bento>bentoml serve service.py:IrisService
2025-10-11T11:30:38+0900 [INFO] [cli] Starting production HTTP BentoServer from "service.py:IrisService" listen
ing on http://localhost:3000 (Press CTRL+C to quit)
2025-10-11T11:30:40+0900 [INFO] [entry_service:iris_classifier:1] Service iris_classifier initialized
```



./docs.json

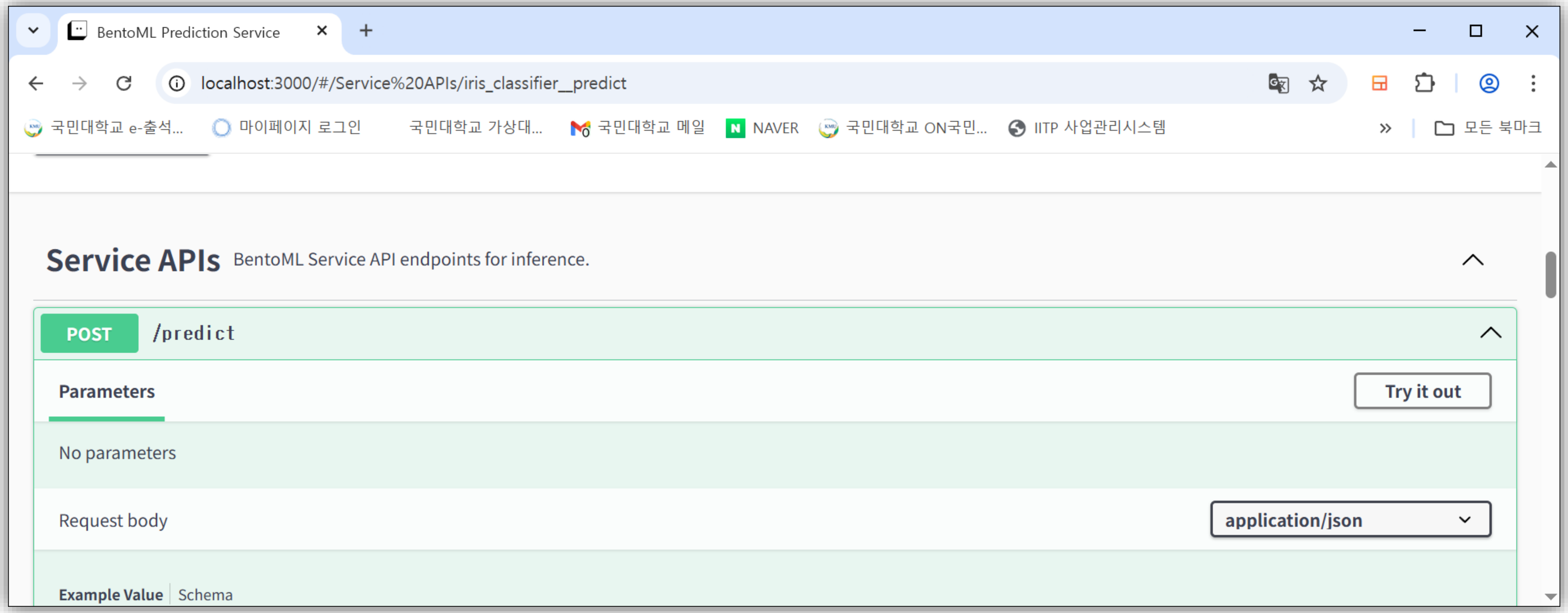
iris_classifier:dev

This is a Machine Learning Service created with BentoML.

Help

- Documentation: Learn how to use BentoML.

■ 테스트



Service APIs BentoML Service API endpoints for inference.

POST /predict

Parameters

Try it out

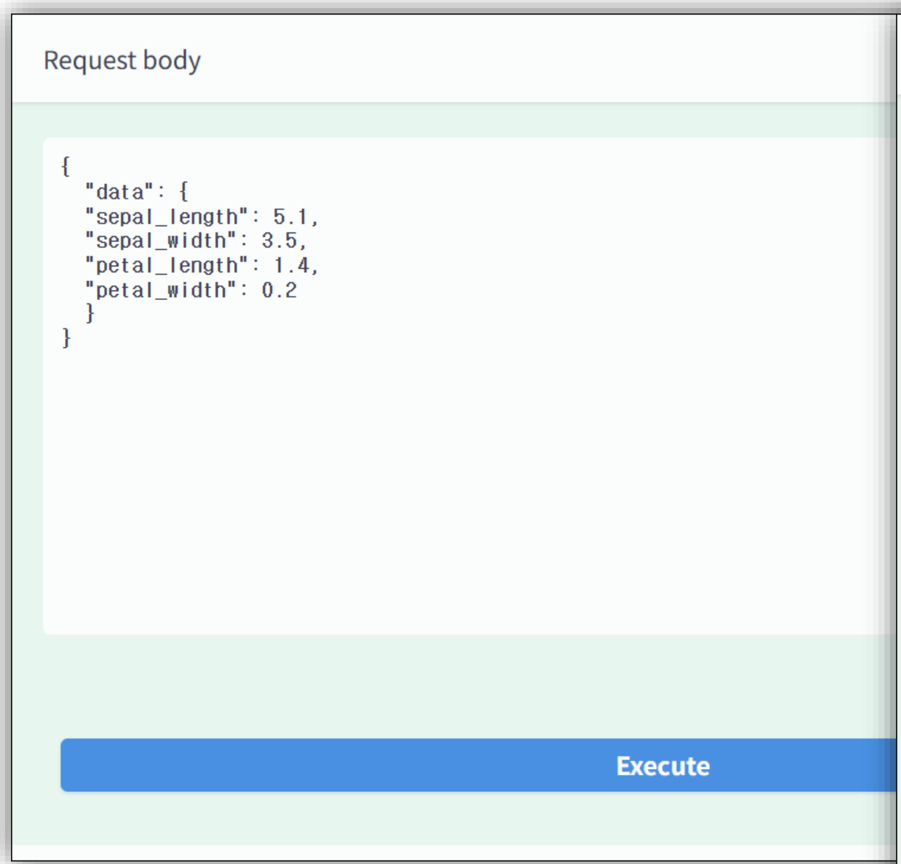
No parameters

Request body

application/json

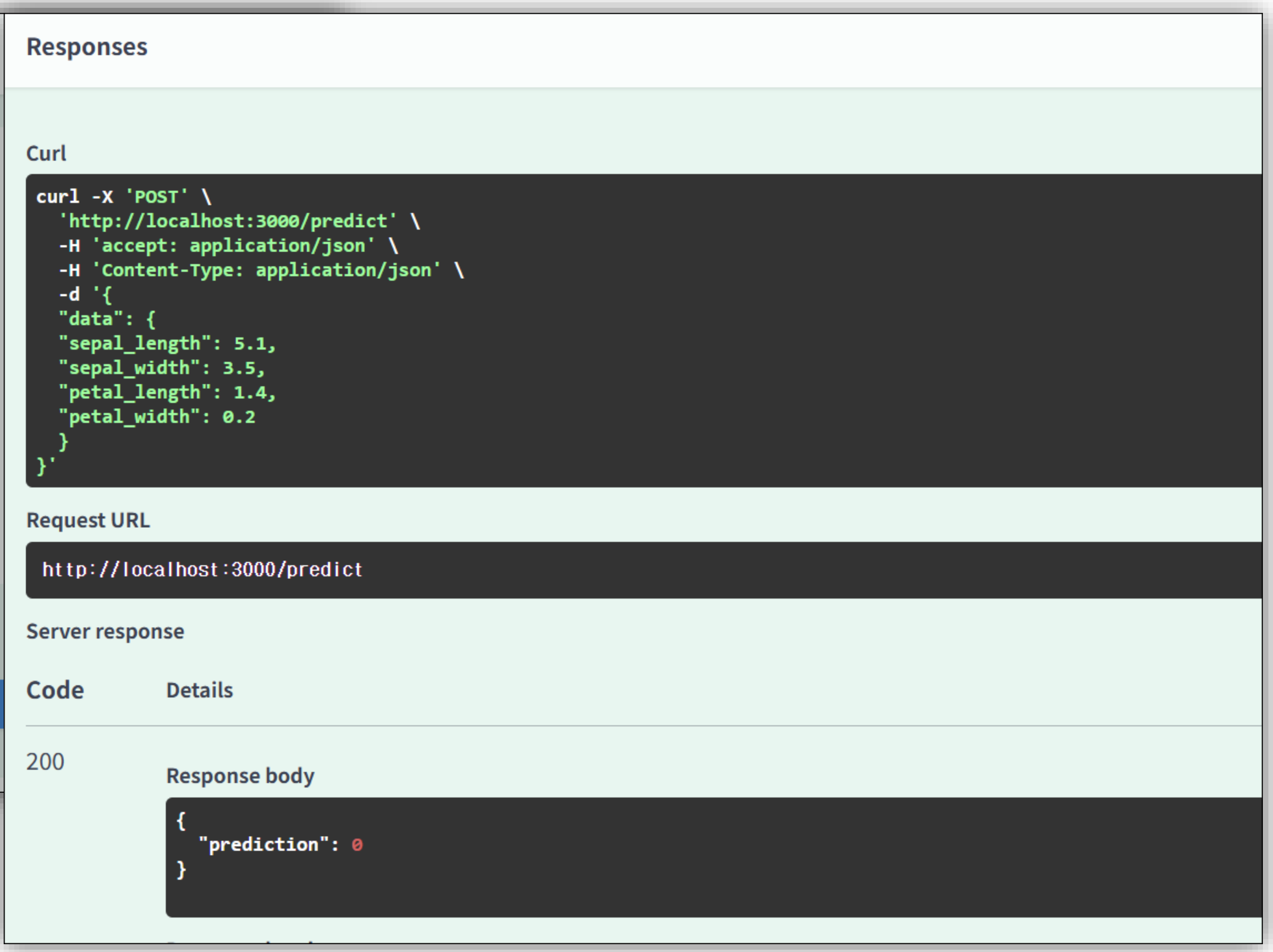
Example Value | Schema

실습



Request body

```
{
  "data": {
  "sepal_length": 5.1,
  "sepal_width": 3.5,
  "petal_length": 1.4,
  "petal_width": 0.2
  }
}
```



Responses

Curl

```
curl -X 'POST' \
  'http://localhost:3000/predict' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "data": {
  "sepal_length": 5.1,
  "sepal_width": 3.5,
  "petal_length": 1.4,
  "petal_width": 0.2
  }
}'
```

Request URL

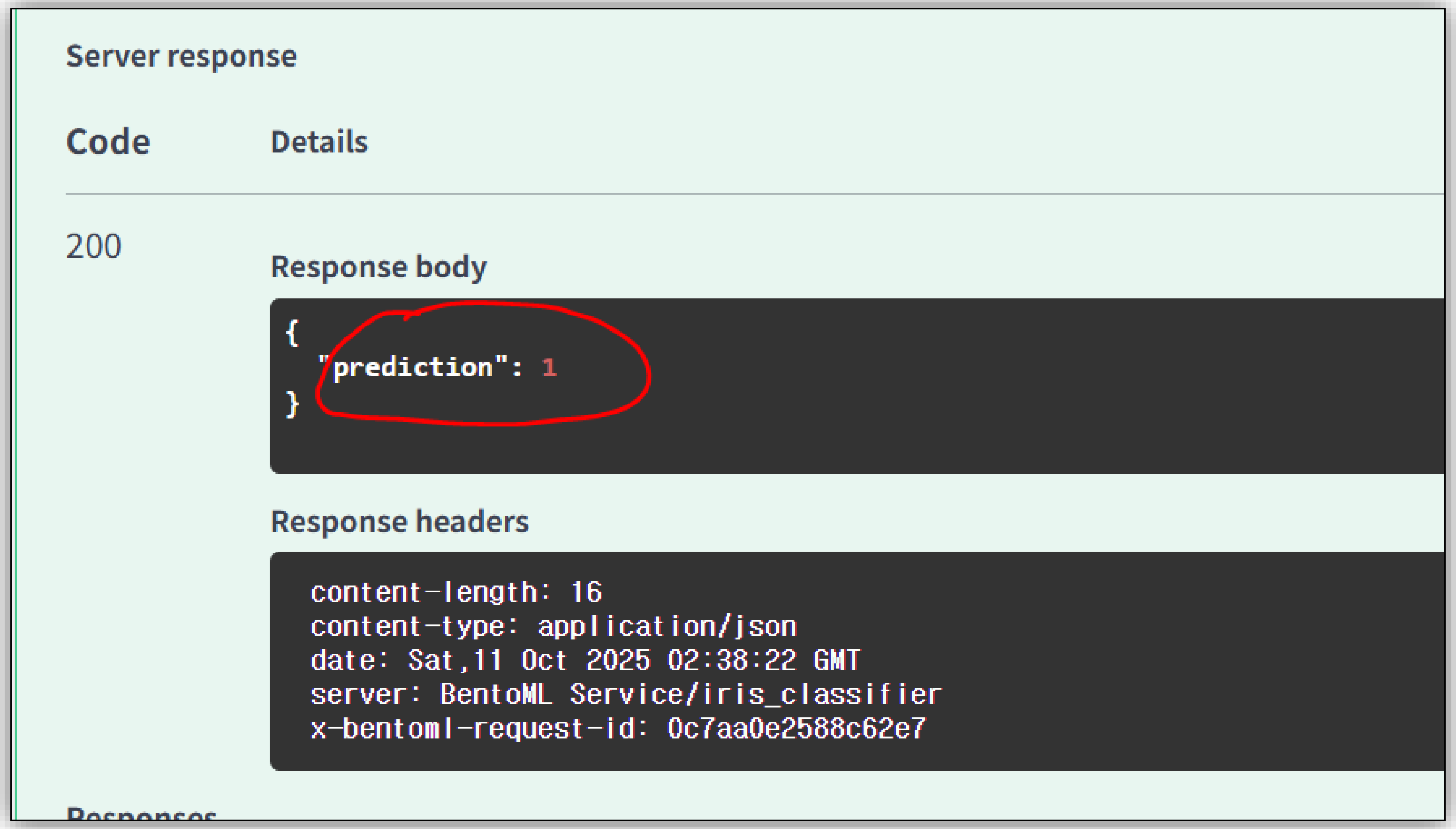
```
http://localhost:3000/predict
```

Server response

Code	Details
200	Response body

```
{
  "prediction": 0
}
```

실습



Server response

Code Details

200

Response body

```
{
  "prediction": 1
}
```

Response headers

```
 content-length: 16
 content-type: application/json
 date: Sat,11 Oct 2025 02:38:22 GMT
 server: BentoML Service/iris_classifier
 x-bentoml-request-id: 0c7aa0e2588c62e7
```

Responses