KOOKMIN UNIVERSITY

실습

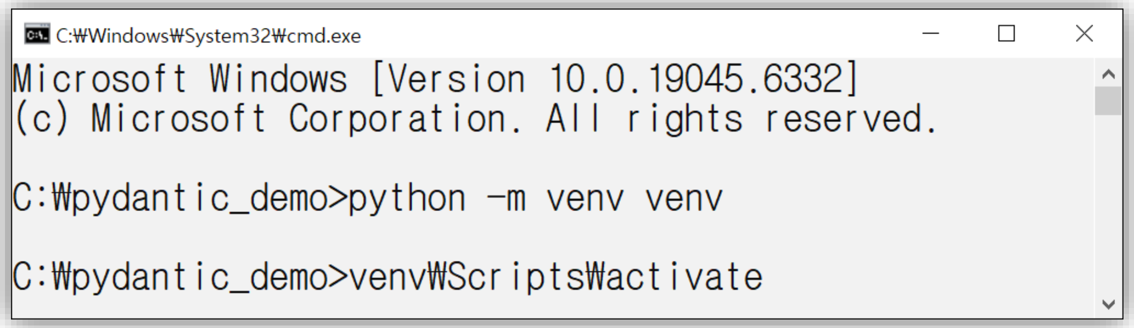
Pydantic

소프트웨어융합대학원
진혜진

실습

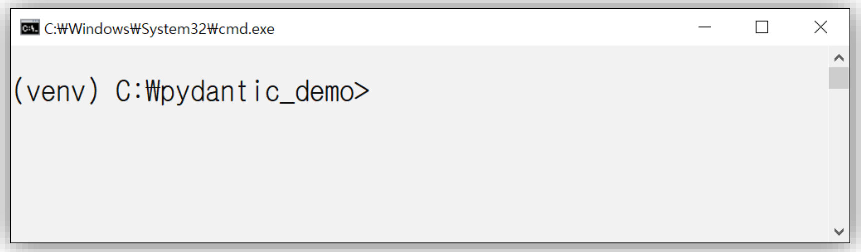
가상 환경 만들기

- 프로젝트 폴더 만들기
- 가상 환경 생성
- 가상 환경 활성화



C:₩Windows₩System32₩cmd.exe

```
Microsoft Windows [Version 10.0.19045.6332]
(c) Microsoft Corporation. All rights reserved.

C:₩pydantic_demo>python -m venv venv

C:₩pydantic_demo>venv₩Scripts₩activate
```



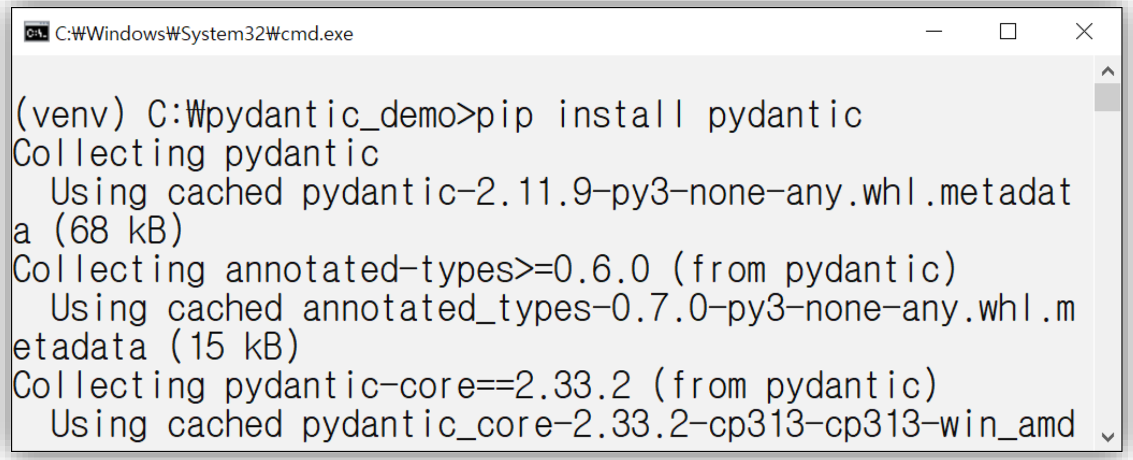
C:₩Windows₩System32₩cmd.exe

```
(venv) C:₩pydantic_demo>
```

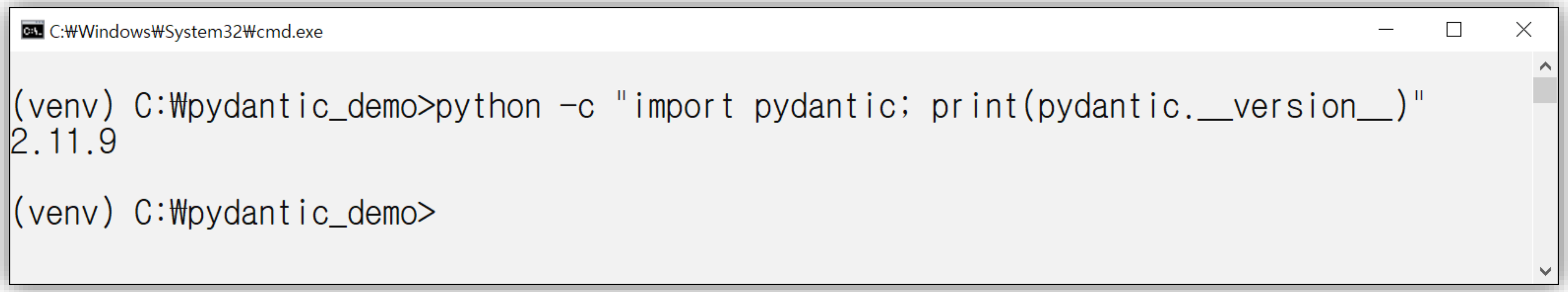
실습

Pydantic 설치

- 가상 환경이 활성화된 상태에서 설치
  - pip install pydantic
- 설치 테스트
  - python -c "import pydantic; print(pydantic.__version__)"



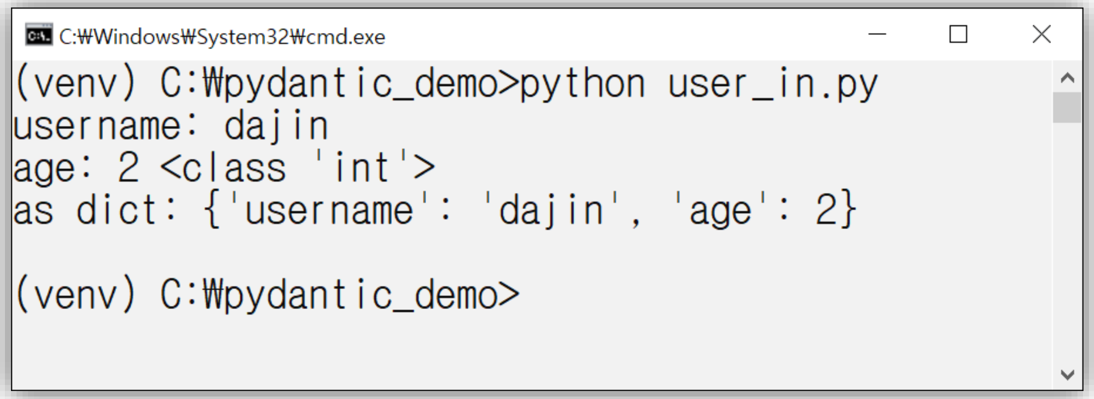
```
(venv) C:₩pydantic_demo>pip install pydantic
Collecting pydantic
  Using cached pydantic-2.11.9-py3-none-any.whl.metadat
a (68 kB)
Collecting annotated-types>=0.6.0 (from pydantic)
  Using cached annotated_types-0.7.0-py3-none-any.whl.m
etadata (15 kB)
Collecting pydantic-core==2.33.2 (from pydantic)
  Using cached pydantic_core-2.33.2-cp313-cp313-win_amd
```



```
(venv) C:₩pydantic_demo>python -c "import pydantic; print(pydantic.__version__)"
2.11.9

(venv) C:₩pydantic_demo>
```

■ 코드 작성 및 실행 (user_in.py)

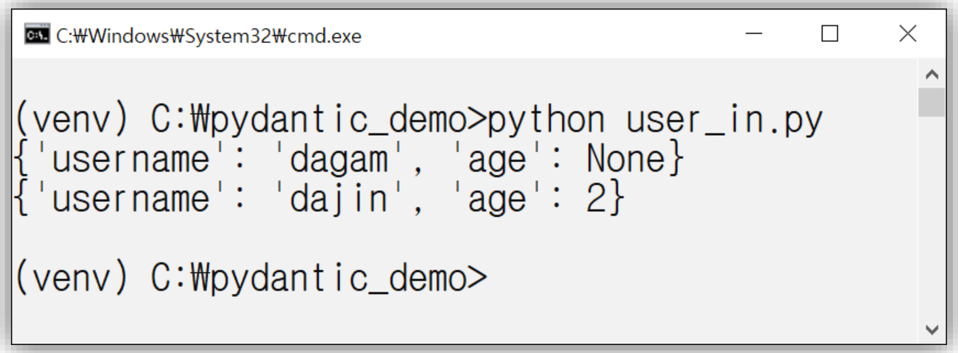
```
from pydantic import BaseModel

class UserIn(BaseModel):
    username: str
    age: int

data = UserIn(username="dajin", age="2")
print("username:", data.username)
print("age:", data.age, type(data.age))

print("as dict:", data.model_dump())
```



```
(venv) C:₩pydantic_demo>python user_in.py
username: dajin
age: 2 <class 'int'>
as dict: {'username': 'dajin', 'age': 2}

(venv) C:₩pydantic_demo>
```

■ 선택적 필드, 기본값

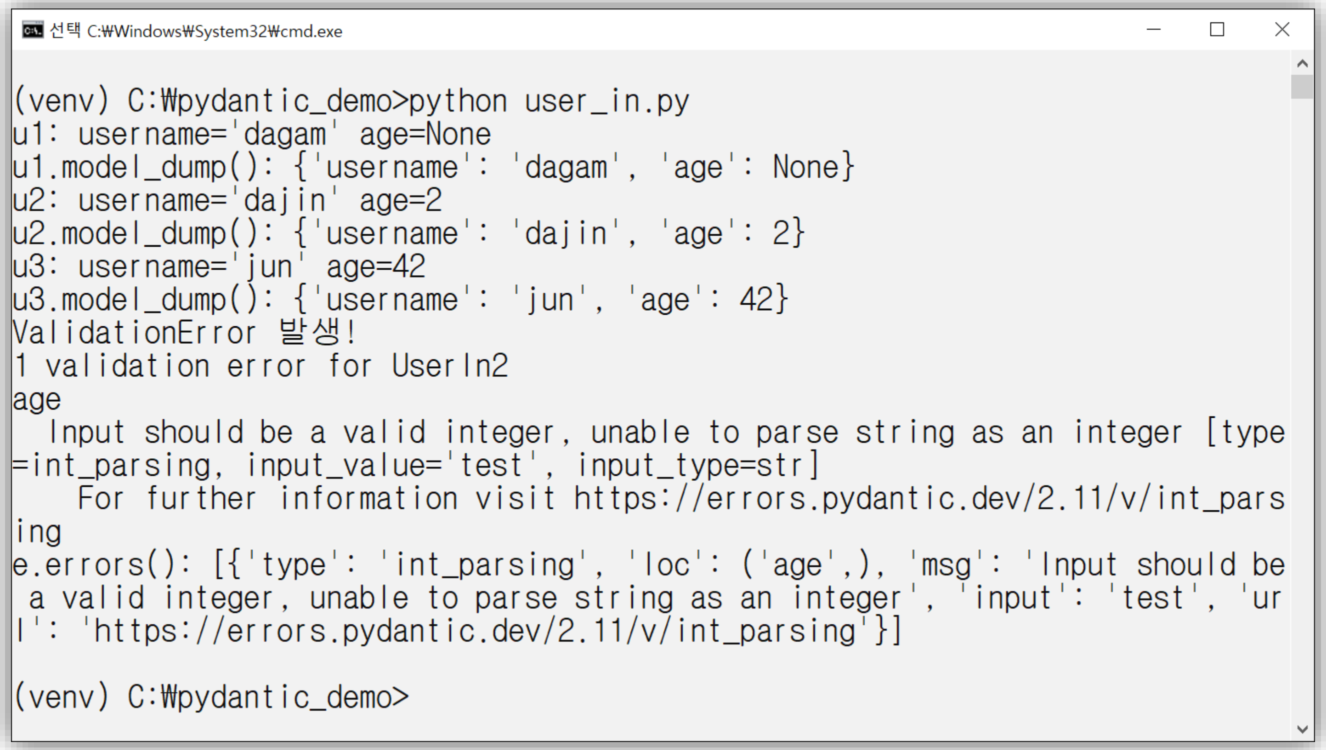
```
from typing import Optional
from pydantic import BaseModel

class UserIn2(BaseModel):
    username: str
    age: Optional[int] = None


u1 = UserIn2(username="dagam")
print(u1.model_dump())


u2 = UserIn2(username="dajin", age="2")
print(u2.model_dump())
```



```
(venv) C:₩pydantic_demo>python user_in.py
{'username': 'dagam', 'age': None}
{'username': 'dajin', 'age': 2}

(venv) C:₩pydantic_demo>
```

실습

```python
from typing import Optional
from pydantic import BaseModel, ValidationError

class UserIn2(BaseModel):
    username: str
    age: Optional[int] = None


u1 = UserIn2(username="dagam")
print("u1:", u1)
print("u1.model_dump():", u1.model_dump())


u2 = UserIn2(username="dajin", age=2)
print("u2:", u2)
print("u2.model_dump():", u2.model_dump())


u3 = UserIn2(username="jun", age="42")
print("u3:", u3)
print("u3.model_dump():", u3.model_dump())


try:
    u_err = UserIn2(username="jin", age="test")
    print("u_err:", u_err)
    print("u_err.model_dump():", u_err.model_dump())
except ValidationError as e:
    print("ValidationError 발생!")
    print(e)

    print("e.errors():", e.errors())
```



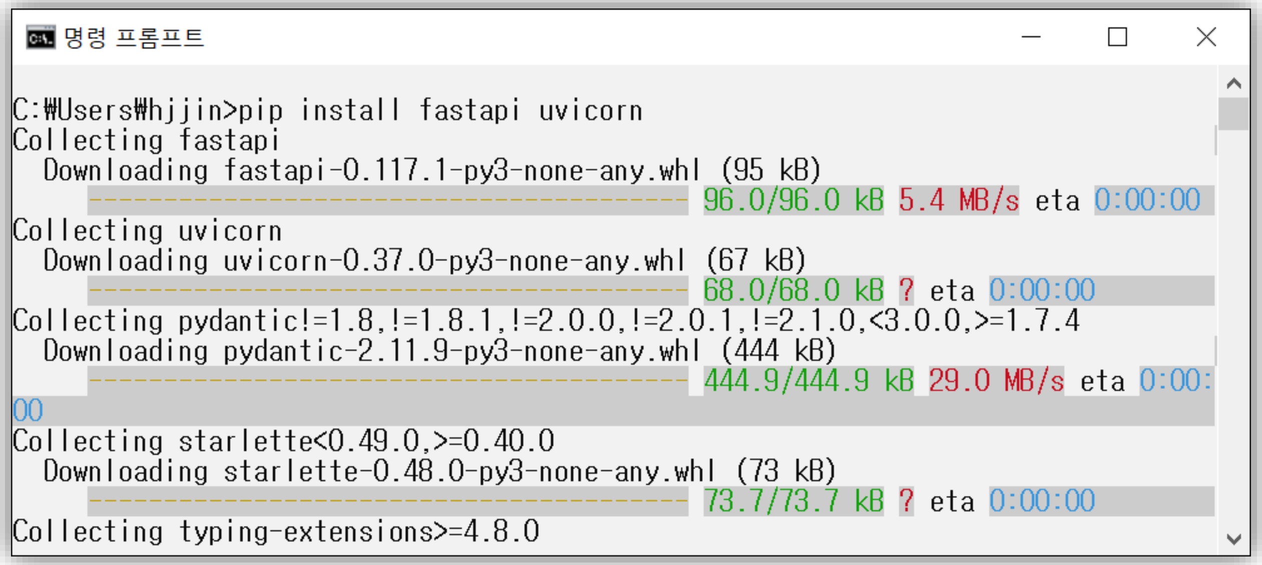
```
(venv) C:₩pydantic_demo>python user_in.py
u1: username='dagam' age=None
u1.model_dump(): {'username': 'dagam', 'age': None}
u2: username='dajin' age=2
u2.model_dump(): {'username': 'dajin', 'age': 2}
u3: username='jun' age=42
u3.model_dump(): {'username': 'jun', 'age': 42}
ValidationError 발생!
1 validation error for UserIn2
age
  Input should be a valid integer, unable to parse string as an integer [type
=int_parsing, input_value='test', input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/int_pars
ing
e.errors(): [{'type': 'int_parsing', 'loc': ('age',), 'msg': 'Input should be
 a valid integer, unable to parse string as an integer', 'input': 'test', 'ur
l': 'https://errors.pydantic.dev/2.11/v/int_parsing'}]

(venv) C:₩pydantic_demo>
```

실습

FastAPI + Pydantic

패키지 설치

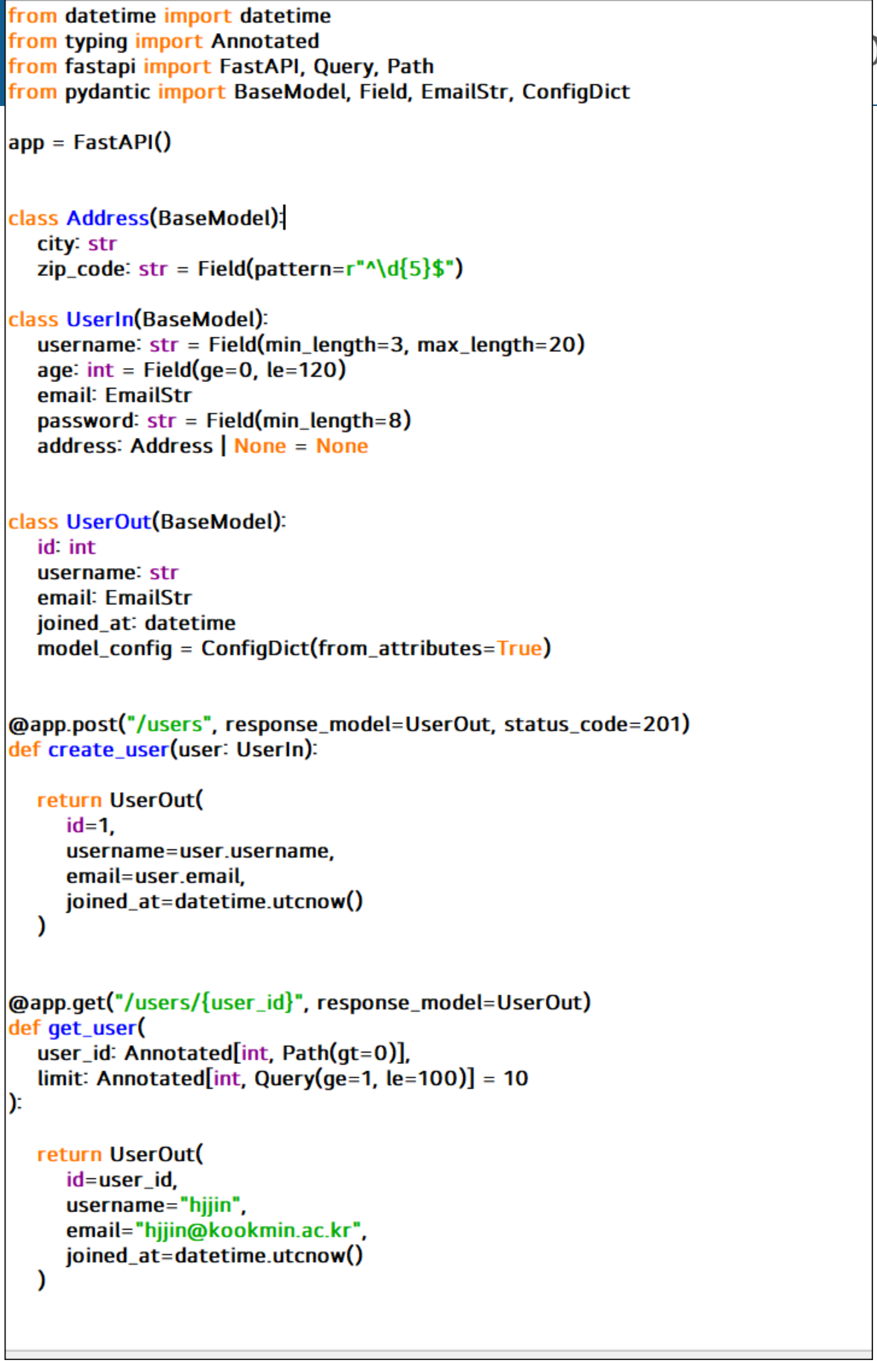
- pip install fastapi "uvicorn[standard]"



```
C:₩Users₩hjjin>pip install fastapi uvicorn
Collecting fastapi
  Downloading fastapi-0.117.1-py3-none-any.whl (95 kB)
     ---------------------------------------- 96.0/96.0 kB 5.4 MB/s eta 0:00:00
Collecting uvicorn
  Downloading uvicorn-0.37.0-py3-none-any.whl (67 kB)
     ---------------------------------------- 68.0/68.0 kB ? eta 0:00:00
Collecting pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4
  Downloading pydantic-2.11.9-py3-none-any.whl (444 kB)
     ---------------------------------------- 444.9/444.9 kB 29.0 MB/s eta 0:00:
00
Collecting starlette<0.49.0,>=0.40.0
  Downloading starlette-0.48.0-py3-none-any.whl (73 kB)
     ---------------------------------------- 73.7/73.7 kB ? eta 0:00:00
Collecting typing-extensions>=4.8.0
```

실습(main.py)

```
from datetime import datetime
from typing import Annotated
from fastapi import FastAPI, Query, Path
from pydantic import BaseModel, Field, EmailStr, ConfigDict

app = FastAPI()


class Address(BaseModel):
    city: str
    zip_code: str = Field(pattern=r"^\d{5}$")

class UserIn(BaseModel):
    username: str = Field(min_length=3, max_length=20)
    age: int = Field(ge=0, le=120)
    email: EmailStr
    password: str = Field(min_length=8)
    address: Address | None = None


class UserOut(BaseModel):
    id: int
    username: str
    email: EmailStr
    joined_at: datetime
    model_config = ConfigDict(from_attributes=True)
```



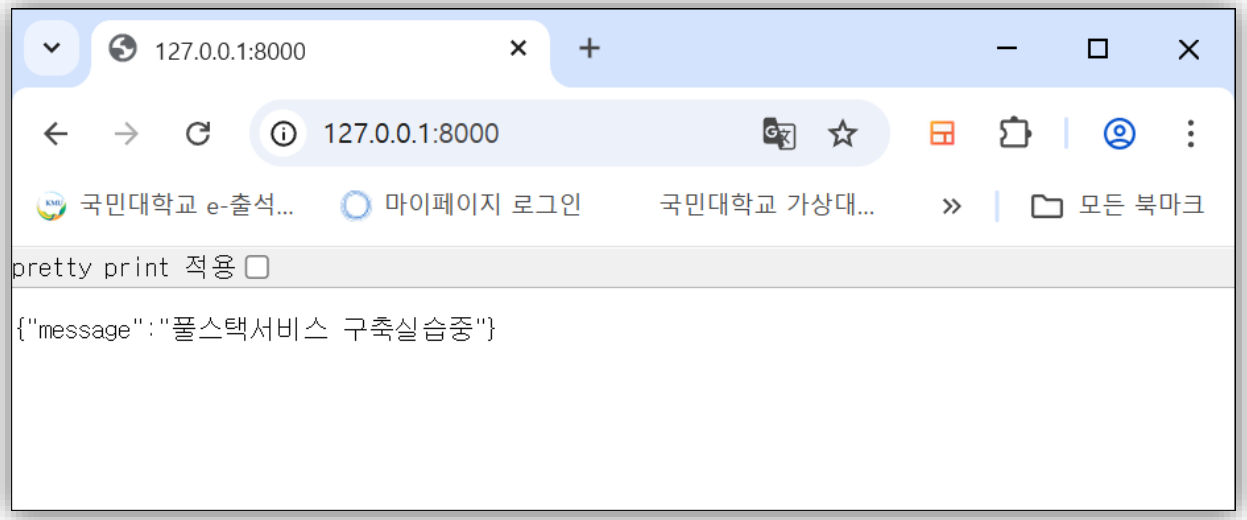
```
from datetime import datetime
from typing import Annotated
from fastapi import FastAPI, Query, Path
from pydantic import BaseModel, Field, EmailStr, ConfigDict

app = FastAPI()


class Address(BaseModel):
    city: str
    zip_code: str = Field(pattern=r"^\d{5}$")

class UserIn(BaseModel):
    username: str = Field(min_length=3, max_length=20)
    age: int = Field(ge=0, le=120)
    email: EmailStr
    password: str = Field(min_length=8)
    address: Address | None = None


class UserOut(BaseModel):
    id: int
    username: str
    email: EmailStr
    joined_at: datetime
    model_config = ConfigDict(from_attributes=True)


@app.post("/users", response_model=UserOut, status_code=201)
def create_user(user: UserIn):

    return UserOut(
        id=1,
        username=user.username,
        email=user.email,
        joined_at=datetime.utcnow()
    )


@app.get("/users/{user_id}", response_model=UserOut)
def get_user(
    user_id: Annotated[int, Path(gt=0)],
    limit: Annotated[int, Query(ge=1, le=100)] = 10
):

    return UserOut(
        id=user_id,
        username="hjjin",
        email="hjjin@kookmin.ac.kr",
        joined_at=datetime.utcnow()
    )
```

실습

```
@app.post("/users", response_model=UserOut, status_code=201)
def create_user(user: UserIn):

    return UserOut(
        id=1,
        username=user.username,
        email=user.email,
        joined_at=datetime.utcnow()
    )


@app.get("/users/{user_id}", response_model=UserOut)
def get_user(
    user_id: Annotated[int, Path(gt=0)],
    limit: Annotated[int, Query(ge=1, le=100)] = 10
):

    return UserOut(
        id=user_id,
        username="hjjin",
        email="hjjin@kookmin.ac.kr",
        joined_at=datetime.utcnow()
    )
```

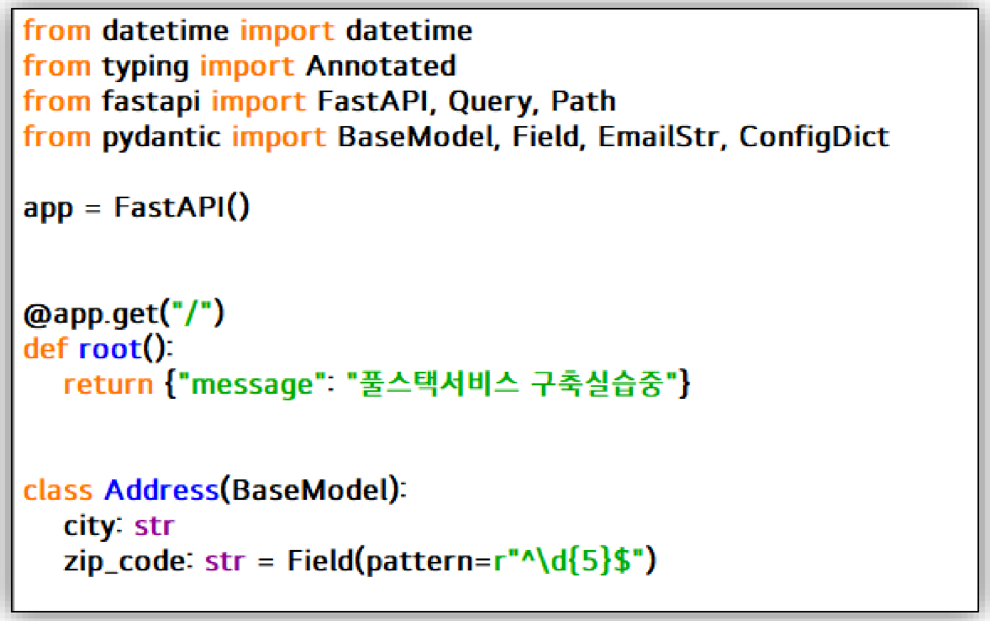
실습

- 서버 실행하기 (uvicorn 사용)
  - uvicorn main:app --reload



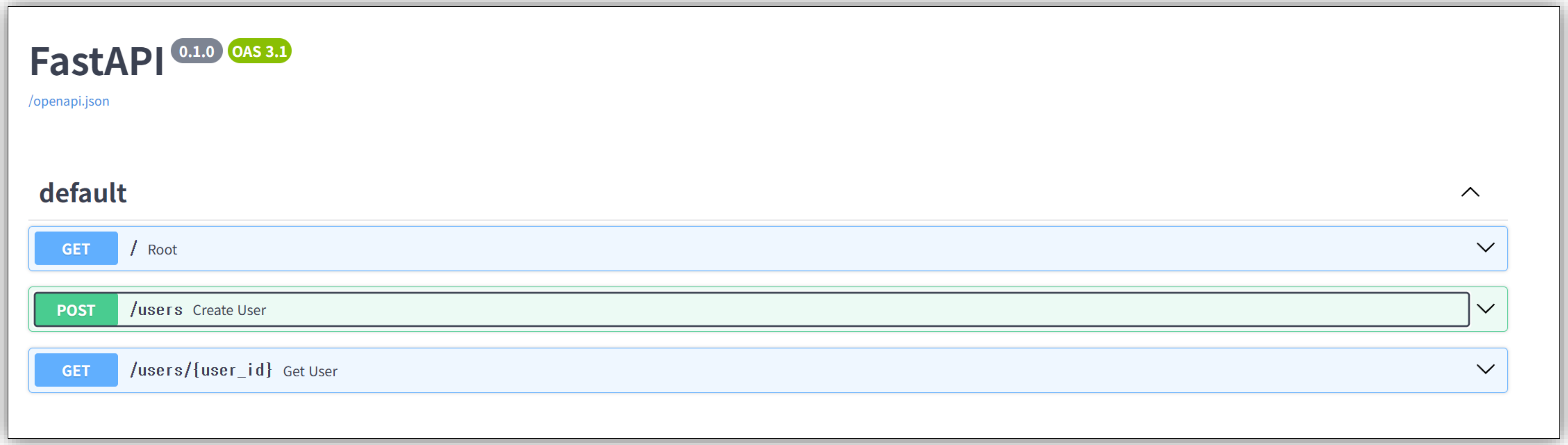
pretty print 적용 □

```
{"message":"풀스택서비스 구축실습중"}
```



```python
from datetime import datetime
from typing import Annotated
from fastapi import FastAPI, Query, Path
from pydantic import BaseModel, Field, EmailStr, ConfigDict

app = FastAPI()


@app.get("/")
def root():
    return {"message": "풀스택서비스 구축실습중"}


class Address(BaseModel):
    city: str
    zip_code: str = Field(pattern=r"^\d{5}$")
```

- 동작 확인 – Swagger UI / 테스트 엔드포인트
  - http://127.0.0.1:8000/docs



FastAPI 0.1.0 OAS 3.1

/openapi.json

default

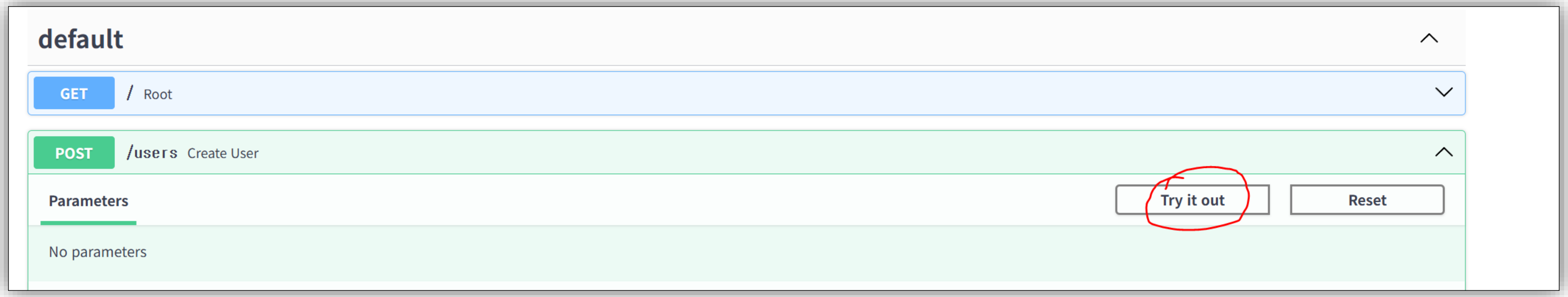
GET / Root

POST /users Create User

GET /users/{user_id} Get User

실습

- 목록에서 POST /users 항목 클릭 → Try it out 버튼 클릭



default

GET / Root

POST /users Create User

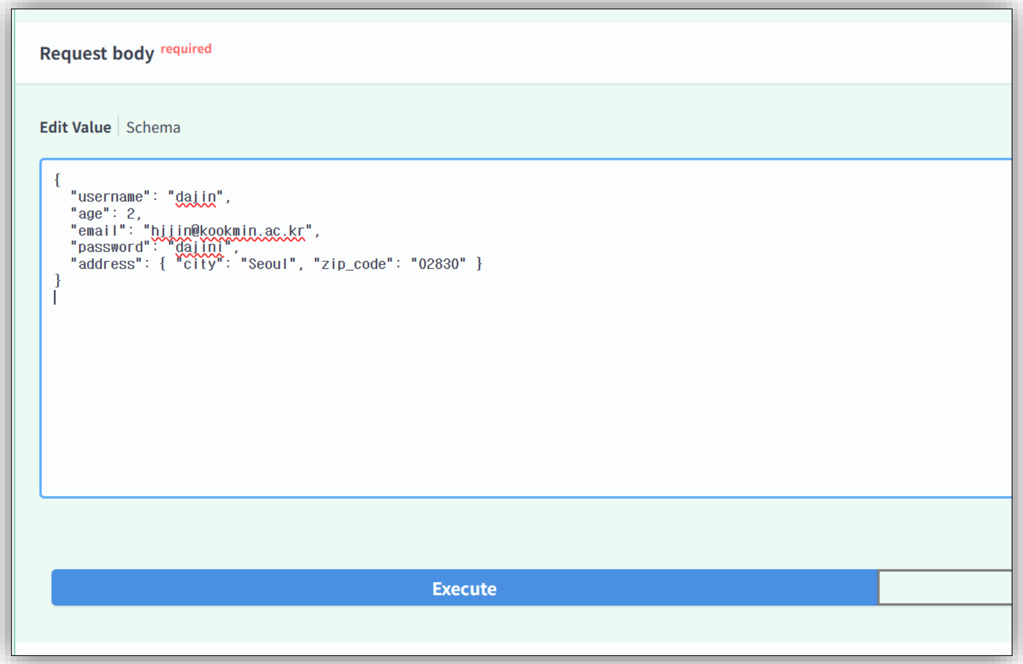
Parameters

No parameters

실습

Request body 입력란

```
{
  "username": "dajin",
  "age": 2,
  "email": "hjjin@kookmin.ac.kr",
  "password": "dajini",
  "address": { "city": "Seoul", "zip_code": "02830" }
}
```

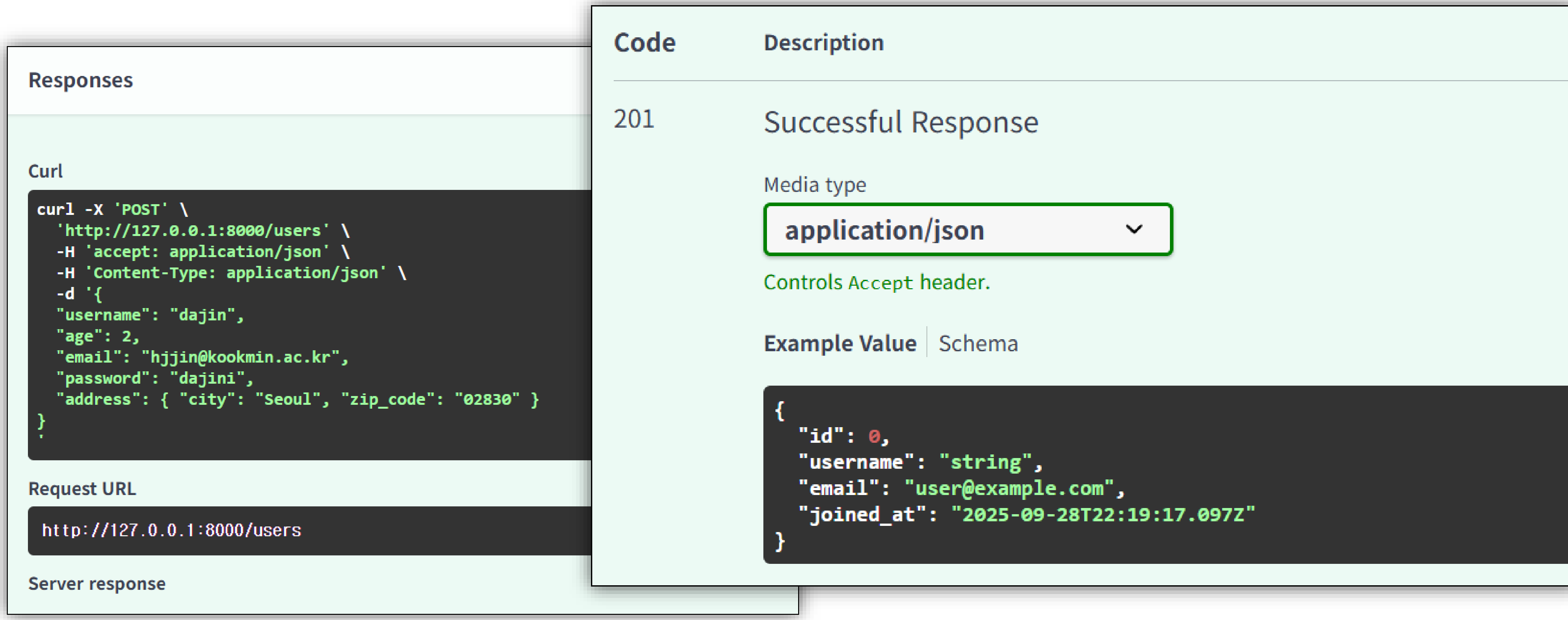


Request body required

Edit Value | Schema

```
{
  "username": "dajin",
  "age": 2,
  "email": "hjjin@kookmin.ac.kr",
  "password": "dajini",
  "address": { "city": "Seoul", "zip_code": "02830" }
}
```

실습



Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/users' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "username": "dajin",
  "age": 2,
  "email": "hjjin@kookmin.ac.kr",
  "password": "dajini",
  "address": { "city": "Seoul", "zip_code": "02830" }
}
'
```

Request URL

```
http://127.0.0.1:8000/users
```

Server response

Code	Description
201	Successful Response

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "id": 0,
  "username": "string",
  "email": "user@example.com",
  "joined_at": "2025-09-28T22:19:17.097Z"
}
```