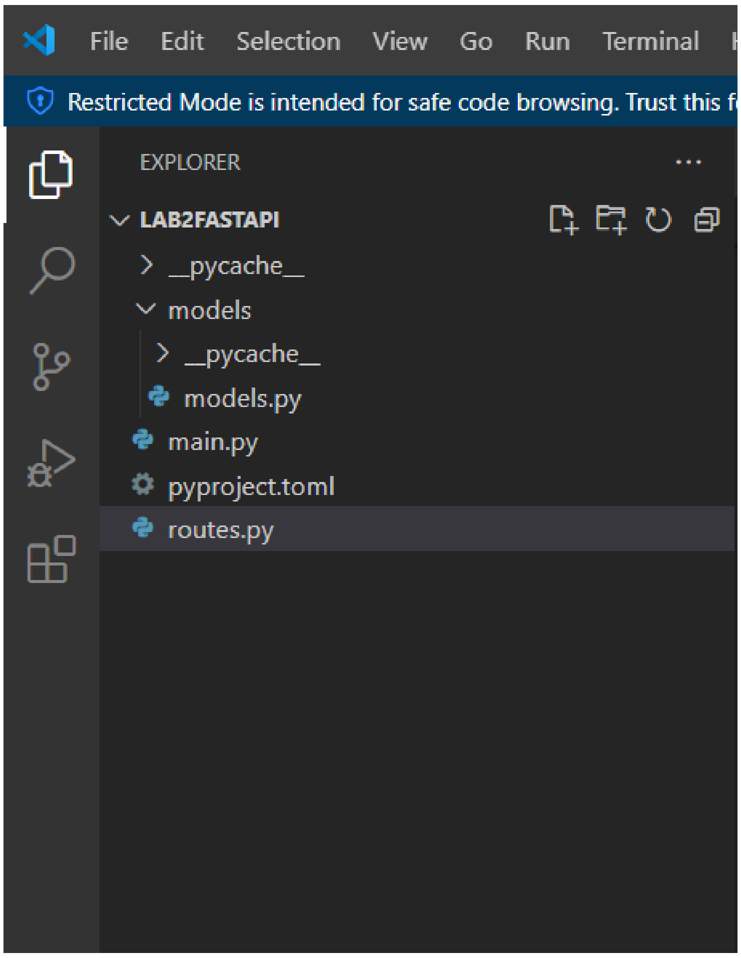
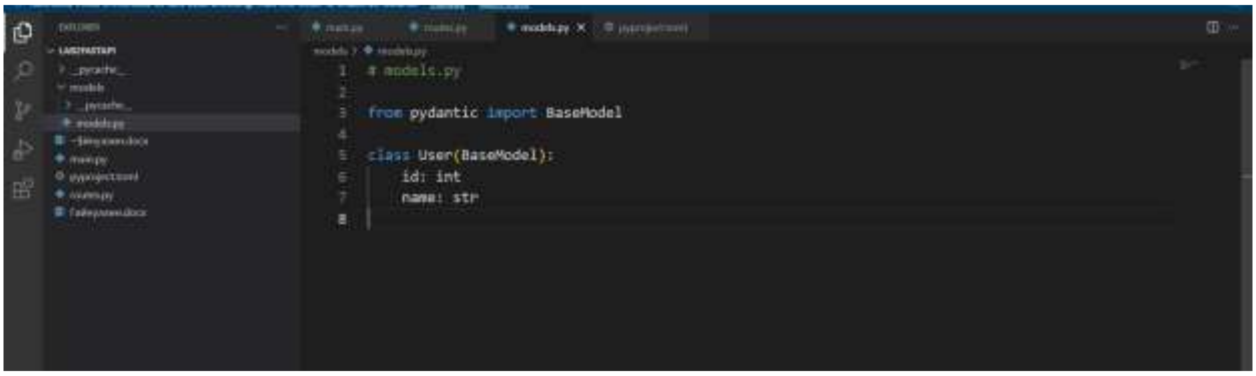
1)



Restricted Mode is intended for safe code browsing. Trust this f

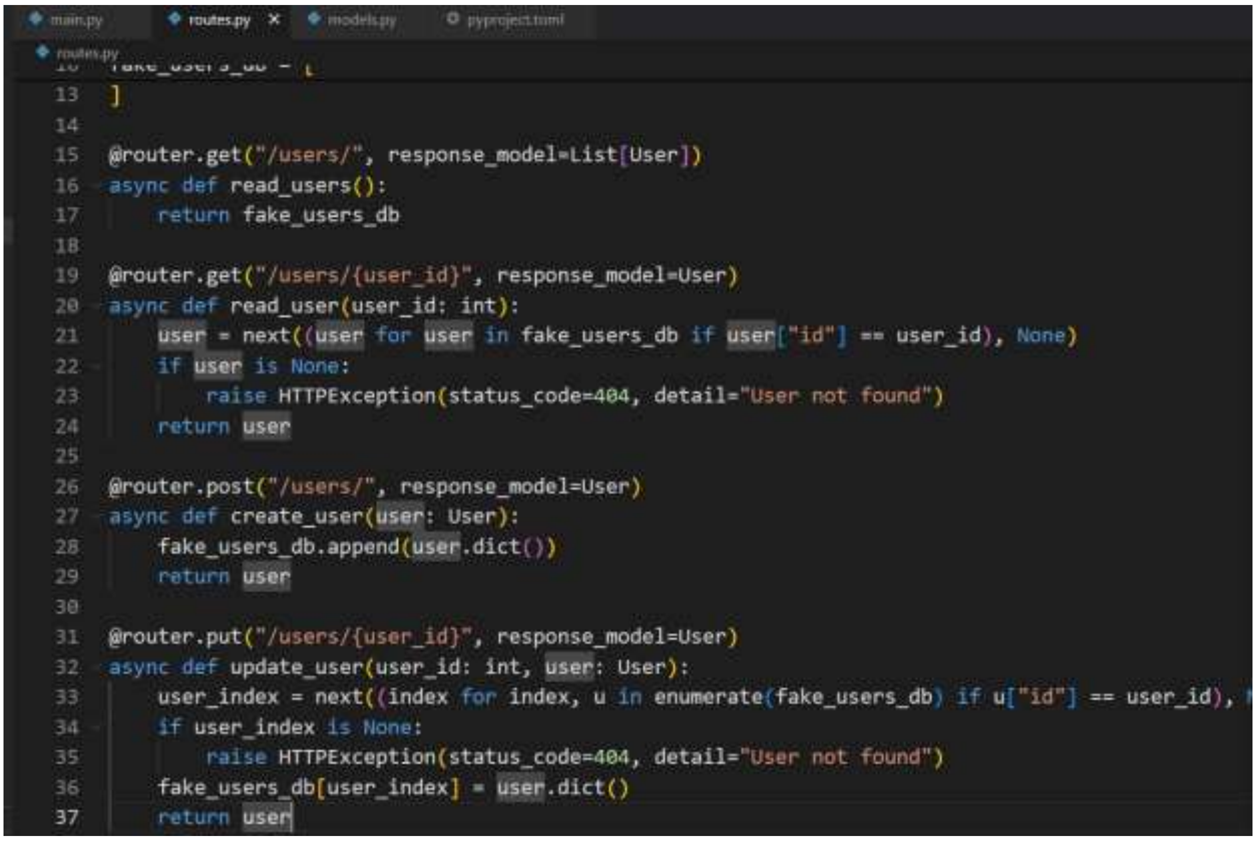
```
LAB2FASTAPI
  > __pycache__
  v models
    > __pycache__
      models.py
    main.py
    pyproject.toml
    routes.py
```

2)



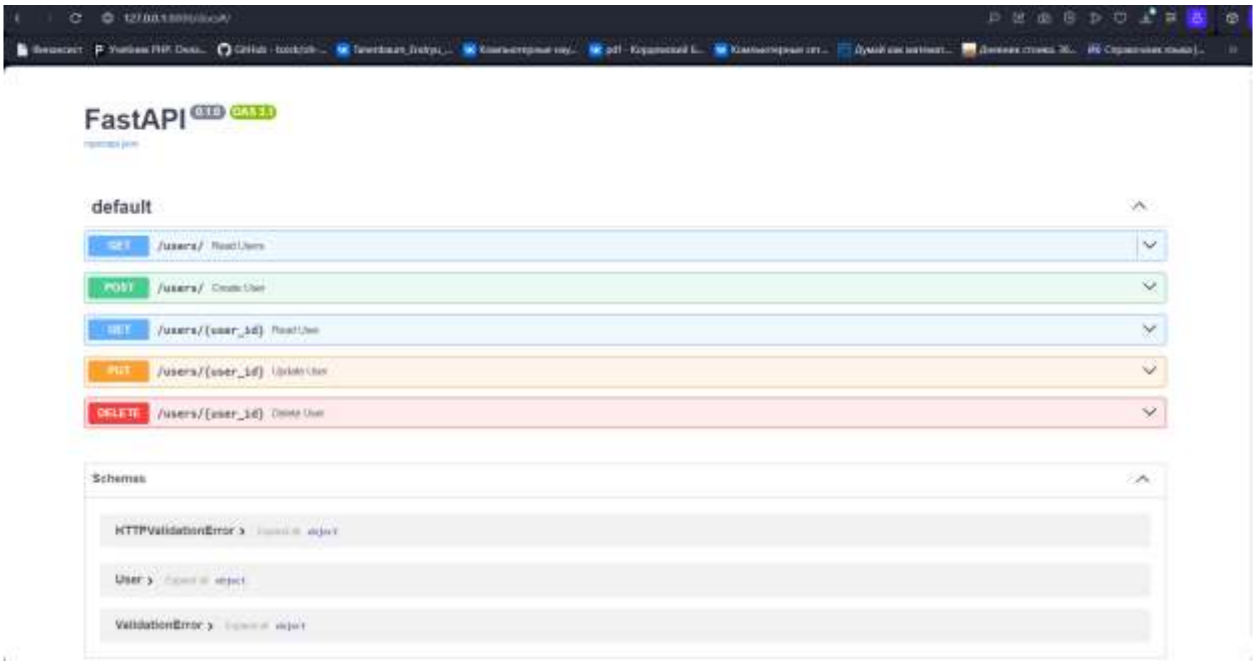
```python
# models.py

from pydantic import BaseModel

class User(BaseModel):
    id: int
    name: str

```

3)



```python
fake_users_db = [
]

@router.get("/users/", response_model=List[User])
async def read_users():
    return fake_users_db

@router.get("/users/{user_id}", response_model=User)
async def read_user(user_id: int):
    user = next((user for user in fake_users_db if user["id"] == user_id), None)
    if user is None:
        raise HTTPException(status_code=404, detail="User not found")
    return user

@router.post("/users/", response_model=User)
async def create_user(user: User):
    fake_users_db.append(user.dict())
    return user

@router.put("/users/{user_id}", response_model=User)
async def update_user(user_id: int, user: User):
    user_index = next((index for index, u in enumerate(fake_users_db) if u["id"] == user_id),
    if user_index is None:
        raise HTTPException(status_code=404, detail="User not found")
    fake_users_db[user_index] = user.dict()
    return user
```

4)



127.0.0.1:8000/docs#/

FastAPI 0.1.0 OAS 3.1

/openapi.json

default

GET /users/ Read Users

POST /users/ Create User

GET /users/{user_id} Read User

PUT /users/{user_id} Update User

DELETE /users/{user_id} Delete User

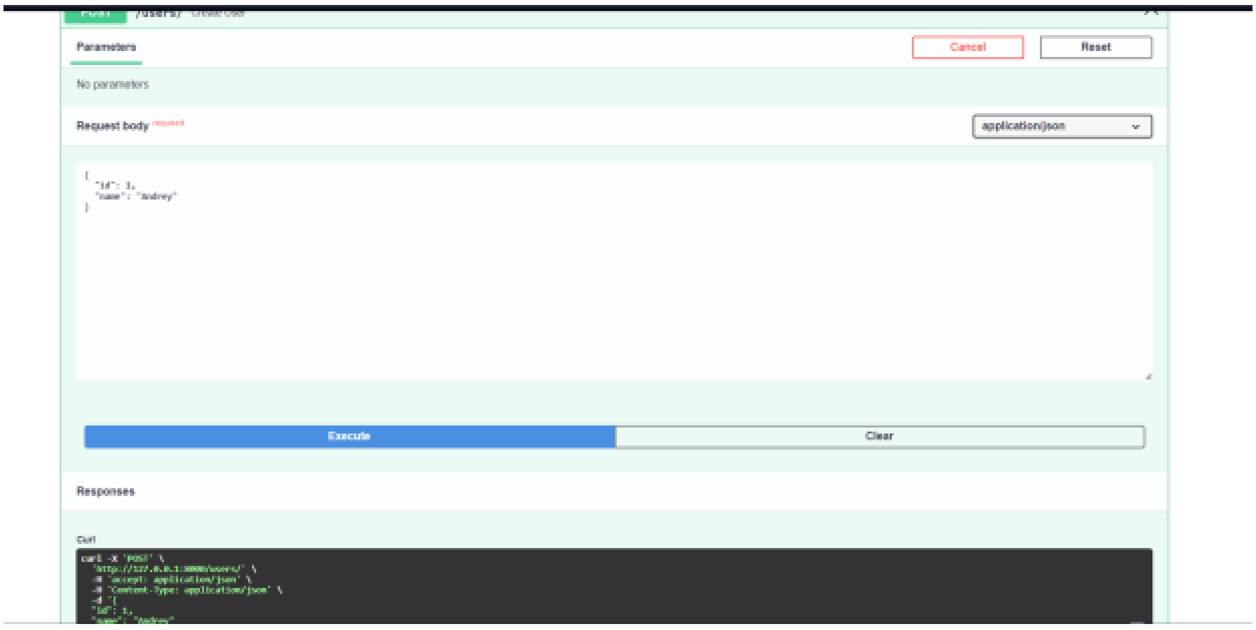
Schemas

HTTPValidationError › Expand all object

User › Expand all object

ValidationError › Expand all object

5)



POST /users/ Create User

Parameters

Cancel Reset

No parameters

Request body required

application/json

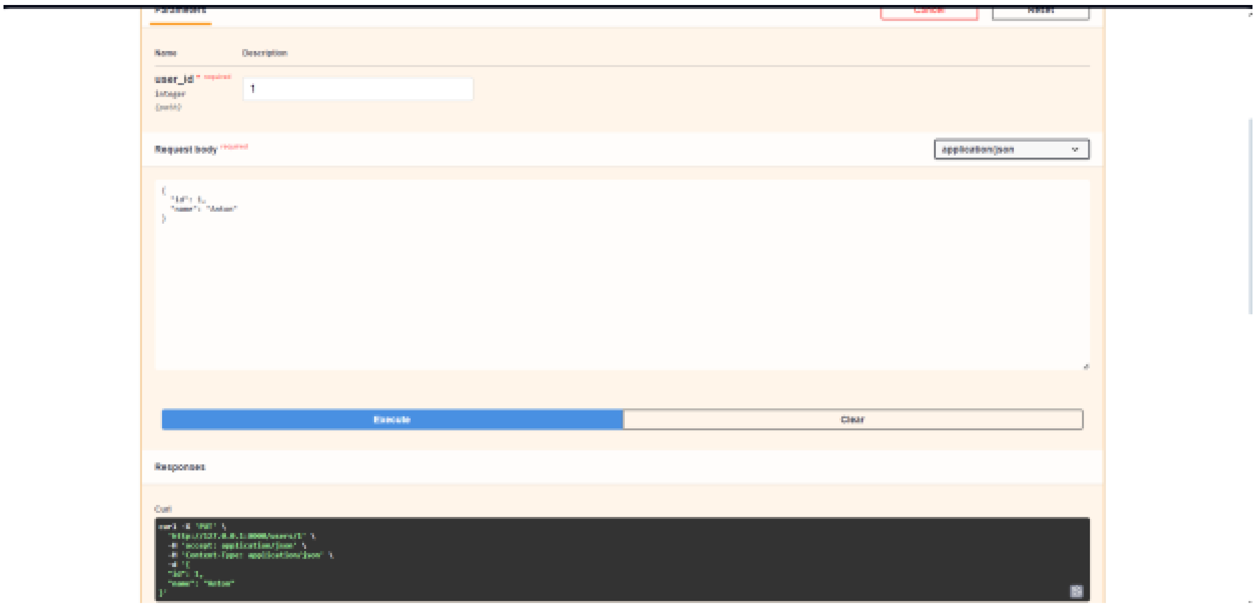
```
{
  "id": 1,
  "name": "Andrey"
}
```

Execute Clear

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/users/' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "id": 1,
  "name": "Andrey"
```



Parameters

Cancel Reset

| Name | Description |
|---|---|
| user_id * required<br>integer<br>(path) | 1 |

Request body required

application/json

```
{
  "id": 1,
  "name": "Anton"
}
```

Execute Clear

Responses

Curl

```
curl -X 'PUT' \
  'http://127.0.0.1:8000/users/1' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "id": 1,
  "name": "Anton"
}'
```