[Semantic Commit Messages (github.com)](https://gist.github.com/joshbuchea/6f47e86d2510bce28f8e7f42ae84c716)

(\*main): **git checkout -b dev**

(\*main): **git push origin feat/crud-album**

* Khi test ở local thì sang branch (**dev** ở local) chạy thử code oke chưa? Nếu oke thì push lên branch (**dev** ở remote)
* Khi branch **dev** ở remote (đã CICD) thì push lên remote test sau (trước mắt là test ở local)
* Branch **dev** sẽ được tạo ra từ branch **main**
* Branch **feat/album-crud** cũngsẽ được tạo ra từ branch
* Link để test: [thinhhuy997/book (github.com)](https://github.com/thinhhuy997/book)

------------------------------------------------------------------------------------------------------------------------

* Về phần source mới (với remote database):
* Copy nội dung từ file “environment.txt” ra file .env
* Tạo file compse: “docker-compose-local.yml”
* Chạy lệnh này: “docker compose -f .\docker-compose-local.yml up”
* Vào đây check: [localhost:8001](http://localhost:8001/)
* Migration:

docker compose -f docker-compose-local.yml exec fastapi\_server alembic revision –autogenerate

103.3.246.128:5854:sun65854:aeu6n

['103.79.142.43', '6497', 'sun16497', 'uiity']

proxy\_1 = proxies\_arr[0]

        proxy\_1 = proxy\_1.split(":")

        proxy\_1 = {"ip": proxy\_1[0], "port": proxy\_1[1],

                   "username": proxy\_1[2], "password": proxy\_1[3]}

        # convert json to proxy schema

        print(IProxyCreate(\*\*proxy\_1))

# NEW FILE

@router.post("/multiple-create")

async def create\_proxies(file: UploadFile = File(...),

                         current\_user: User = Depends(

        deps.get\_current\_user(required\_roles=[IRoleEnum.admin, IRoleEnum.manager])

    ),

                         ):

    if not file:

        return {"message": "No upload file sent"}

    else:

        print('----------------------------------')

        proxies\_str = str(file.file.read())

        proxies\_str = proxies\_str.strip()

        proxies\_str = proxies\_str.strip("b'")

        proxies\_arr = proxies\_str.split("\\r\\n")

        # proxy\_1 = IProxyCreate(proxies\_arr[0])

        proxy\_1 = proxies\_arr[0]

        proxy\_1 = proxy\_1.split(":")

        proxy\_1 = {"ip": proxy\_1[0], "port": proxy\_1[1],

                   "username": proxy\_1[2], "password": proxy\_1[3]}

        proxy\_2 = proxies\_arr[0]

        proxy\_2 = proxy\_2.split(":")

        proxy\_2 = {"ip": proxy\_2[0], "port": proxy\_2[1],

                   "username": proxy\_2[2], "password": proxy\_2[3]}

        # convert json to proxy schema

        # print(IProxyCreate(\*\*proxy\_1))

        proxy\_1 = IProxyCreate(\*\*proxy\_1)

        proxy\_2 = IProxyCreate(\*\*proxy\_2)

        created\_proxies = await crud.proxy.multiple\_create(objs\_in=[proxy\_1, proxy\_2], created\_by\_id=current\_user.id)

        return {"file\_name": file.filename}

# NEW FILE

@router.post("/multiple-create")

async def create\_proxies(file: UploadFile = File(...),

        current\_user: User = Depends(

        deps.get\_current\_user(required\_roles=[IRoleEnum.admin, IRoleEnum.manager])),):

    if not file:

        return {"message": "No upload file sent"}

    else:

        all\_proxies\_str = str(file.file.read())

        all\_proxies\_str = all\_proxies\_str.strip()

        all\_proxies\_str = all\_proxies\_str.strip("b'")

        # Convert to the string array

        all\_proxies\_str\_arr = all\_proxies\_str.split("\\r\\n")

        proxies = []

        # Loop and data to list of json

        for line in all\_proxies\_str\_arr:

            proxy\_info = line.split(":")

            proxy\_json = {"ip": proxy\_info[0], "port": proxy\_info[1],

                    "username": proxy\_info[2], "password": proxy\_info[3]}

            # convert to ICreateProxy Schema

            proxies.append(IProxyCreate(\*\*proxy\_json))

        # Create multiple proxies

        created\_proxies = await crud.proxy.multiple\_create(objs\_in=proxies, created\_by\_id=current\_user.id)

        return create\_response(data=created\_proxies)

103.79.142.43:6497:sun16497:uiity

* Sai định dạng là thiếu field “IP”, còn mấy field khác thiếu vẫn được

Check 2 dòng đầu của file txt:

* Nếu không đúng định dạng thì raise error luôn