

Shapr3D API

Használt technológiák:

Az API pythonban készült [fastAPI](#) segítségével, az adatbázis kezeléshez [SQLAlchemy](#) használtam, ami több adatbázist is támogat én ezek közül az [SQLite](#)-ot választottam. ASGI-nek pedig az [Uvicorn](#)-t használtam. Tesztelésre van Swagger beépítve (ip:port/docs) emellett Insomniát használtam még. Verziókezelőnek git-et. IDE-nek pedig Pycharmot.

Manuális indítás:

Bemegyünk az hometest mappába, hogy egy szinten legyünk a main.py fájlal és kiadjuk terminálba

```
uvicorn main:app --host 0.0.0.0 --workers 12
```

ha meg akarjuk adni a portot is azt a - -Port-al tudjuk a worker szám ahhoz kell hogy hány worker process-t indítson el a uvicorn (minél nagyobb a szám annál több kérést tudunk egyszerre kezelni) viszont ez függ a szerver CPU-tól is az ajánlott worker szám az $2 * \text{number_of_cores} + 1$ ha development módban akarjuk indítani akkor azt a

```
> uvicorn main:app --reload
```

-el tudjuk

ilyenkor bármit változtatunk a kódban az egyből bekerül az API-ba nem kell újraindítani és a localhost:8000-en indul el

felépítés:

main.py:

itt van behívva a fastAPI meg a middleware itt van megadva
meg be van húzva a router az endpoints.py-ból

endpoints.py:

itt vannak megadva az endpointok meg pár alap response

databasehandler.py:

itt van megadva egy függvény ami megnyitja a adatbázis felé
a kapcsolatot majd bezárja azt, ez a függvény később
injektálva lesz az endpointoknál (Session = Depends(get_db))

backgroundTask.py:

ebben van megvalósítva az a szál ami elindítja a convertert és
frissíti az adatbázisban a progress-t

converter.py:

konvertálás imitálása

models.py:

leírja az adatbázis egyik táblájában a modelt

endpointok:

post /convert:

elindítja egy fájl konvertálását a megadott formátumra
visszaadja a konvertált fájl nevét és ID-ját
uploadedFile: a konvertálandó fájl
convertto: a kiterjesztés amire konvertálni szeretnénk

`get /history:`

visszaadja az adatbázisban lévő konvertálásokat és azok állapotát

`get /origin/{file_id}:`

letöltési pontot ad vissza a megadott id-jú konvertálandó fájlhoz

`get /converted/{file_id}:`

letöltési pontot ad egy átkonvertált fájlhoz

Deployment:

[fastAPI dokumentáció](#) szerint bármelyik cloud providert lehet használni. Én személyesen a Porter-t választanám AWS-el. Ehhez található részletes [guideline](#) a Porter oldalán.

Scalability:

Uvicorn A Scalability szempontjából nagyon jó a workerek segítségével. Ha nagyon nagyon sok kérés érkezik ami le terheli a szervert akkor a konvertálást át lehet mozgatni egy külön szerverre ehhez annyit kell változtatni vagy a `backgroundTask`-on vagy a `convert` endpointon hogy ne ugyanazon a gépen indítsa a konvertálást hanem a másikon. Emellett biztosítani kell a konvertálást végző gépet hogy legyen elérése az adatbázishoz. Ha ez se lenne elég be lehet vezetni több konvertálást végző gépet és az API mindig azon indítaná el a konvertálást ami a legkevésbé van leterhelve