Házi Feladat

Beadási határidő:

2025.01.12 23:59:59

A házifeladat megoldásait külön-külön fájlban mentsétek el, zippelve az alábbi mail címre küldhetitek el: vince.dulicz@gmail.com

(+akinek van már GitHub mehet oda is, ezzel később fogunk foglalkozni)

1. Felhasználói adatok szinkronizálása

- Használj egy API-t például a httpbin-t.
- Készíts egy osztályt UserSync néven, amely a felhasználói adatok küldéséért és lekéréséért felelős egy szerverről.
- Az osztály tartalmazza az alábbi metódusokat:
    - `create_user(self, user_data: dict)`: Egy új felhasználó adatainak küldése (POST kérés).
    - `get_user(self, user_id: int)`: Egy adott felhasználó adatainak lekérése (GET kérés).
    - `update_user(self, user_id: int, update_data: dict)`: Felhasználó adatainak módosítása (PUT vagy PATCH kérés).
    - `delete_user(self, user_id: int)`: Felhasználó törlése a szerverről (DELETE kérés).
- Minden metódus a szerver válaszát JSON formátumban adja vissza.

Extra:

- Használj logikát a válaszok kezelésére, pl. státuszkód alapján különböző kivételeket dobj (raise_for_status).

---

2. Weboldal státuszának monitorozása

- Hozz létre egy WebsiteMonitor nevű osztályt, amely az alábbi metódusokat tartalmazza:
    - `check_status(self, url: str)`: Küldj egy GET kérést az adott URL-re, és add vissza a státuszkódot, valamint a válaszidőt.
    - `is_online(self, url: str)`: Döntsd el, hogy az oldal elérhető-e (200-as státuszkód esetén elérhető).
    - `get_headers(self, url: str)`: Szerezd meg az oldal válaszának fejlécét.
- Implementálj egy naplózási funkciót is, amely minden vizsgálat eredményét elmenti egy fájlba (monitor.log).

Extra:

- Adj hozzá időzített ellenőrzéseket (pl. 1 percenként).
- Több weboldal támogatása egyszerre