**Java-backend-coding-challenge projekt**

A projekt részét képező anyagokat angolul írom. A pályázat részét képező információkat magyarul.

1. Ha igazi projekt lenne, akkor elkészíteném azokat a dokumentációkat, amikre most csak utalok:
   1. IFA. Ebben definiálnám pontosan a feladatleírásban szereplő adatszerkezetet. Ha van rá fejlesztői kapacitás mindkét rendszer oldalán, akkor az extract és import service-eket egy közös swagger definíció alapján, dinamikusan lehetne konfigurálni. Ez lehetne pl. egy git submodule-ban, amit mindkét projekt elér.
   2. Detail Design, DD
      * Pontosan definiálnám a file exchange-et. Mindkét rendszer által elérhető linux storage, külön alkalmazás user-eken és mount pontokon. Szükséges az sftp, mert a file import service-t kubernetes clusteren belülre tervezem, ahol csak sftp kulccsal lehet authentikálni. Definiálnám az ehhez szükséges kubernetes paramétereket, környezeti változókat, secret-eteket és config map-eket, az sftp privát kulcsot tartalmazó rendszert (pl. azure vault).
      * Liquibase-t használnék. Ha nincs akadálya annak, hogy a java alkalmazás által használt DB alkalmazás usernek DDL jogosultságot adjunk, akkor a liquibase user a default jdbc usert használná.
      * Spring batch-et használnék az adat betöltésre. A DD-ben leírnám a szükséges konfigurációt, pl. a spring batch adattábláinak kezelését, az ütemezési config externalizációját.
   3. A System Manual-be kerülnének az üzemeltetési részletek.
2. Demó projekten belül ezt lehetetlen mind megvalósítani, ezért csak egy leegyszerűsített alkalmazást implementáltam, amit github-ról letöltve és build-elve a feladat kiírásában szereplő file-okat automatikusan betölti egy localban már létező mysql adatbázisba. Részletek a projekt readme-ben.
3. A szükséges, de idő hiányában el nem végzett feladatokról, és definíciós problémákról létrehoztam egy Todo.txt file-t a kód Documentation folderében.