

## Zadanie 1

Zadanie je zamerané na vytvorenie prvotnej verzie aplikačného servera s API (End-point), ktorý po zavolaní vráti požadované hodnoty. Zadanie 1 slúži ako základ pre ostatné zadania, ktoré budú potrebné vypracovať počas semestra.

**Termín odovzdania:** 23:59 26.2.2023 pre všetky časti odovzdania (AIS, Github a Nahlasenie linku na docker image cez formulár)

### **Pokyny k odovzdaniu:**

- **AIS** – odovzdávajú sa zdrojové kódy (tak aby bolo možné skompilovať a spustiť projekt) spolu s číslom commitu, ktorý predstavuje finálnu verziu Vášho zadania 1.
- **Github** – vedenie si svojho **privátneho** repozitára v rámci Github classroomu. Názov je vo formáte:
  - o predmet\_a\_rok-priezvisko\_cviciaceho-termín\_cvicenia-priezvisko\_meno\_študenta
  - o napr. dbs23-bencel-stv9-mrkvicka-jan
  - o Github classroom je distribuovaný osobitne pre jednotlivé cvičenia
- **Formulár** – nahlásenie linku na Váš docker image v rámci vášho Github repozitára s príslušným označením podľa pokynov nižšie
  - o link: <https://forms.gle/3hUhiL1JVr5Wv5Zu8>

Odobzdanie je považované za kompletne, ak sú odobzdané všetky časti (AIS, Github, Link na docker image) do termínu odobzdania. V prípade neskorého odobzdania je zadanie považované ako neodobzdané.

### **Popis zadania**

V rámci zadania je potrebné implementovať funkčné RESTful API rozhranie. Je možné použiť ľubovoľný programovací jazyk, tak aby bola splnená požadovaná funkcionality. Riešenie je potrebné mať publikované v rámci Vášho prideleného privátneho repozitára. GitHub repozitár musí obsahovať vytvorený Docker image, ktorá po spustení počúva na TCP porte 8000 a ako vstup berie environment premenné (case sensitive) opísané v tabuľke nižšie, ktoré slúžia na konfiguráciu databázového pripojenia. Príklad takéhoto riešenia je opísaný v repozitári <https://github.com/FIIT-Databases/dbs-python-example>. Súčasťou repozitára je aj CI/CD konfigurácia pre automatické vytváranie Docker image, ktorú jednoduchými obmenami viete adaptovať na Vaše riešenie.

<i>Environment premenná</i>	<i>Popis</i>	<i>Príklad</i>
<code>DATABASE_HOST</code>	IP adresa databázového servera	127.0.0.1
<code>DATABASE_PORT</code>	Port databázového servera	5432
<code>DATABASE_NAME</code>	Názov databázy	dbs
<code>DATABASE_USER</code>	Databázový používateľ	xstudent
<code>DATABASE_PASSWORD</code>	Prihlasovacie heslo	SuperTajneHeslo

Aplikačný server načíta tieto premenné a na základe nich vytvorí pripojenie k databázovému serveru. Databázový server je PostgreSQL.

Aplikačný server implementuje RESTful volanie **GET /v1/status**, ktoré po zavolaní vráti JSON objekt, ktorý obsahuje verziu databázového servera. Pre získanie verzie databázového servera je potrebné zavolať dopyt (query) na databázu. Vrátený parameter je potrebné následne vrátiť v JSON objekte s nasledujúcou štruktúrou:

```
{
    "version": "informácia vrátená databázovým serverom"
}
```

V praxi môže JSON objekt vyzeráť nasledovne:

```
{
    "version": "PostgreSQL 15.1 on aarch64-apple-darwin21.6.0,
compiled by Apple clang version 14.0.0 (clang-1400.0.29.102), 64-bit"
}
```

Pre otestovanie vlastnej implementácie je možné použiť nástroj cURL, ktorým môžete realizovať volanie na server. Príklad volania pre server <http://127.0.0.1:8000> s volaním zo zadania môže vyzeráť nasledovne:

```
curl http://127.0.0.1:8000/v1/status
```

Pre overenie správnosti vášho riešenia pre už vytvorený docker image je možné použiť testovací nástroj na adrese: <https://tester-dbs.fiit.stuba.sk>. Pomocou tohto nástroja bude prebiehať aj finálne overovanie funkčnosti zadania. Do samotného testera sa prihlasuje pomocou údajov z AISu. Následne je potrebné vybrať konkrétne zadanie, ktoré je predmetom testovania a vloženie linky na vytvorený docker image v rámci Vášho privátneho github repozitára.

Linka na github image môže vyzeráť nasledovne:

```
ghcr.io/fiit-databases/dbs-example:1.0.0
```

Príklad nekompletného riešenia:

<https://github.com/FIIT-Databases/dbs-python-example>

- toto riešenie je možné použiť a vychádzať z neho pri implementácii zadania. Obsahuje všetky náležitosti, ktoré sú potrebné pre vytvorenie docker image.

### **Publikovanie zadania v Github**

Každé zadanie musí byť publikované pomocou Github Releases. Pre automatické vytvorenie docker image je potrebné použiť CI/CD skript, ktorý automaticky vytvorí image.

V rámci ukážkového repozitára je vytvorený github workflow `.github/workflows/publish.yml`, ktorý automaticky vytvorí image pre verzie master, main a sémantické verziovanie x.x.x. Link na sémantické verziovanie: <https://semver.org>

Váš finálny release pre zadanie je potrebné označiť podľa semantického verziovania a teda pre zadanie 1 je to označenie ako 1.x.x. Ukážka príkladu release je uvedená v príklade riešenia.

**Programovací jazyk** – nie je obmedzenie – je však potrebné splniť definovanú funkcionality vrátane spustiteľnosti