

# Pripojenie na databázový server z Docker prostredia

**Termín odovzdania:** 23:59 25.2.2024 pre všetky časti odovzdania (AIS, Github a nahlásenie linku na Docker image cez formulár)

## Popis zadania

Zadanie je zamerané na vytvorenie prvej verzie RESTful aplikačného servera s endpointom, ktorý po zavolaní vráti požadované hodnoty. Tento aplikačný server je následne distribuovaný ako Docker image (Docker image je vstup pre automatický DBS tester, ktorý budete používať počas semestra). Zadanie 1 slúži ako základ pre ostatné zadania, ktoré budú potrebné vypracovať počas semestra.

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, ktorý 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 FIIT-Databases/dbs-python-example. Súčasťou repozitára je aj CI/CD konfigurácia (GitHub Actions) pre automatické vytváranie Docker image, ktorú jednoduchými obmenami viete adaptovať na Vaše riešenie.

Environment variable	Popis	Príklad
DATABASE_HOST	IP adresa databázového servera	127.0.0.1
DATABASE_PORT	Port databázového servera	5432
DATABASE_NAME	Názov databázy	dbs
DATABASE_USER	Databázový používateľ	xstudent
DATABASE_PASSWORD	Prihlasovacie heslo	SuperTajneHeslo

Table 1: Environment variables

Aplikačný server načíta premenné z tabuľky XY 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:

```
1 {  
2   "version": "informácia vrátená databázovým serverom"  
3 }
```

V praxi môže JSON objekt vyzeráť nasledovne:

```

1 {
2   "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"
3 }

```

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:

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

Pre overenie správnosti vášho riešenia pre už vytvorený docker image je možné použiť testovací nástroj `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`

Ako príklad vám môže poslúžiť repozitár `FIIT-Databases/dbs-python-example`. Obsahom je implementácia Python REST servera, 'Dockerfile' a konfigurácia GitHub Actions na publikovania Docker image. S obsahom repozitára môžete manipulovať na základe MIT licencie a použiť vo svojich zadaniach. Adaptácia pre rôzne programovacie jazyky a technológie by mala byť jednoduchá.

## Pokyny k odovzdaniu

Každé zadanie musí byť publikované ako Github Release. 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`.

Za účelom testovania zadania nemusíte vytvárať nové releases. Tester akceptuje aj images pre 'master' alebo 'main'.

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

Pre úspešné odovzdanie zadania, musí byť splnené všetky body uvedené nižšie:

- Zdrojové kódy (tak aby bolo možné skompilovať a spustiť projekt) spolu s hashom commitu (ktorý predstavuje finálnu verziu Vášho zadania 1.) boli odovzdané do **AIS**.
- Na **GitHube** je vedený privátny repozitár v rámci GitHub Classroom v nasledovnom formáte.
  - `predmet_a_rok-priezvisko_cviciaceho-termín_cvicenia-priezvisko_meno_študenta`
  - napr. `dbs24-bencel-stv9-mrkvicka-jan`
  - Github Classroom je distribuovaný osobitne pre jednotlivé cvičenia
- Adresa vášho Docker Image bola nahlásená pomocou **Google Formulára** <https://forms.gle/Rq5CeUX25qzNy9Kt7>

Odovzdanie je považované za kompletne, ak sú odovzdané všetky časti (AIS, Github, Link na docker image) do termínu odovzdania. V prípade neskorého odovzdania je zadanie považované ako neodovzdané.