

## Příklad pro Java/Kotlin Developera

Cílem tvé práce bude vytvořit aplikaci, která bude ukládat data do databáze, získávat data z databáze a tato data následně poskytnout uživateli, který je bude moci upravit a následně opět uložit. Tvá aplikace bude používat technologie, které jsou popsány níže.

Zadání tohoto motivačního příkladu tě povede "krok za krokem", ale pokud máš nápady jak to udělat lépe, klidně je realizuj. Příklad není nijak záladný a nemusíš v něm hledat zbytečnou složitost (jednoduché a funkční řešení je to správné řešení). Pokud chceš, máš možnost se předvést a jít za rámec zadání a dovést aplikaci k produkční kvalitě.

Během realizace se neboj zeptat na cokoli, co ti nebude jasné, ať ze samotného zadání, či použití konkrétní technologie.

Je rok 2024, a svět je plný nástrojů, které Ti vývoj mohou ulehčit. Víťame to, a tím, když nám řekneš nebo dokonce ukážeš jak jsi je použil na nás můžeš udělat dojem.

Přejeme Ti veselé programování 😊

### Nástroje

- Git
  - repozitář nastav jako veřejný s nastaveným právem pro čtení
- Gradle / Maven

### Databáze

- [PostgreSQL](#) (aktuální verze) nebo jiná relační DB

### Backend by měl obsahovat

- Kotlin / Java 17+
- Spring Boot (aktuální verze)
- Hibernate (aktuální verze), Jooq, nebo jiný Tvůj oblíbený přístup k databázi
- REST API pomocí Rest controllerů

### Zadání

#### Cíle

- Vytvoř jednoduchou aplikaci, která bude využívat Spring Boot (<https://spring.io/guides/gs/spring-boot>)
- Aplikace bude implementovat RestController, který bude vracet testovací data
  - pro testování restovek můžeš použít aplikaci "Postman" nebo "Advanced REST client"
- Všechny restovky budou přijímat a vracet data pouze ve formátu JSON (pokud tomu nebrání funkce Controlleru, např. načítání obrázků, což není předmětem tohoto úkolu)

### Fáze 1 (vytvoření databáze)

- Vytvoř si novou databázi a v ní tabulku "users"
  - prozatím stačí atributy *id* a *name*
- Nakonfiguruj databázi pomocí Spring Boot a připrav si to co potřebuješ pro práci s tabulkou "users"
- Do databáze vlož několik testovacích záznamů
- Vytvoř nový RestController pro práci s uživateli
  - přidej novou metodu pro načtení uživatele
  - metoda bude schopna převzít id uživatele z parametru URL

- následně načte data z databáze skrze business (servisní) vrstvu, která využije DAO vrstvu
- data o uživateli vrátí jako response ve formátu JSON
- pokud uživatel není v databázi, tak metoda controlleru vrátí odpovídající response code a chybovou hlášku

## Fáze 2 (práce s daty)

- v RestController naimplementuj metody pro :
  - vytvoření nového uživatele,
  - úpravu již existujícího uživatele
  - získání seznamu všech uživatelů
  - smazání uživatele
- všechna vstupní data budou validována pomocí validátorů (ideálně pomocí anotace *javax.validation.Valid*)
  - zamysli se, co všechno by bylo potřeba validovat

## Fáze 3 (uživatelské přihlašování)

- do tabulky v DB přidej atributy *username* a *password*
- uživatel by měl mít možnost nastavit si a změnit heslo
  - porad' si se situací, kdy si heslo při úpravě změnit nechce
- přidej konfiguraci Spring Security a napoj na uživatele přidané do databáze
  - bude možné se přihlásit na základě nově vytvořeného uživatele
  - zajisti, aby upravit a mazat uživatele mohl jen přihlášený uživatel
  - použij Basic autentizaci
    - na straně UI postačí, při volání metody pro smazání uživatele, pouze přidat hlavičku s parametrem "Authorization": "Basic" + Base64.encode("<username>:<password>")

## Fáze 4 (refaktoring, vylepšení, výhled do budoucna) - nepovinná

- máš možnost se v příkladu předvést a přidat, nebo zdokumentovat cokoli Tě napadne, že by potřebovala, kdyby se jednalo o skutečnou aplikaci
- pravděpodobně se budeme ptát
  - co by se dalo vylepšit ale nezbyl čas, nebo je nad rámec zadání ?
  - instrukce pro Tvé budoucími kolegy, aby si mohli aplikaci rozjet
  - jak by se aplikace nasazovala do produkce, databáze, běhové prostředí, automatizace ...
  - jak by se dala řešit situace, kdy každý uživatel může upravit a smazat jen svůj profil a jen administrátor může upravovat kohokoli ?
- pokud máš tu možnost, vezmi si notebook s příkladem na další kolo, můžeme se bavit přímo nad Tvým kódem v tvém oblíbeném editoru.