



**Dokument** | Java zadání

Moravské náměstí 14, 602 00 Brno

[www.inqool.cz](http://www.inqool.cz)

## Úvod

Toto cvičenie je vytvorené pre kandidátov na pozíciu Java vývojára v spoločnosti InQool a.s. Cieľom tohto zadania je ukázať spôsoby, ktorými kandidát pristupuje k riešeniu problémov a schopnosť kandidáta naučiť sa nové technológie. Použitie konkrétnych postupov a technológií (s výnimkou explicitne vyžiadanych technológií) je na kandidátovi, všetky jednoduché kreatívne nápady sú každopádne vítané.

## POPIS PROJEKTU

Vytvorte serverovú aplikáciu na správu rezervácií v tenisovom klube.

Tenisový klub má určité množstvo kurtov, ktoré majú rôzne povrchy. Na jednotlivé druhy povrchov sa viažu rôzne minútové ceny prenájmu. Možné typy povrchov sa môžu v čase meniť, preto je nutné implementovať typ povrchu ako číselník, ktorý môže užívateľ systému spravovať. Rezervácie sa vždy týkajú len jednotlivých kurtov a môžu byť vytvorené na ľubovoľný zmysluplný časový interval, s vopred označením, či sa bude hrať dvojhra alebo štvorhra. Za štvorhru sa cena prenájmu násobí x1.5. Rezervácie sa vždy viažu na jedno telefónne číslo zákazníka, ku ktorému sa ukladá vždy rovnaké meno.

Pri zakladaní rezervácie sa zároveň vytvorí nový užívateľ, pokiaľ v systéme zatiaľ neexistuje užívateľ s daným telefónnym číslom. Operácia delete musí byť implementovaná ako soft delete nad všetkými entitami. Súčasťou externej konfigurácie bude možnosť zapnúť inicializáciu dát. Pokiaľ bude táto možnosť zapnutá, po spustení aplikácie sa automaticky inicializujú dva typy povrchov a štyri kurty.

Výstupom aplikácie budú REST endpoints, umožňujúce:

- CRUD operácie nad kurtami
- RUD operácie nad rezerváciami
- Prístup k zoznamu rezervácii pre konkrétne unikátne číslo kurtu, zoradené podľa dátumu vytvorenia rezervácie
- Prístup k zoznamu rezervácii pre konkrétne telefónne číslo, s možnosťou vyžiadať si iba rezervácie v budúcnosti

- Vznik rezervácie, pre daný kurt, daný typ hry a dané telefónne číslo a meno zákazníka. V odpovedi systém vráti cenu rezervácie (nutná validácia, že daný časový interval sa neprekrýva s inou rezerváciou na rovnakom kurte).

Pozornosť je dobré venovať všeobecným best practices, a to najmä minimálnej duplicite kódu, dobrej dokumentácii a čitateľnosti.

## Požiadavky

- Využitie frameworku Spring boot
- Využitie JPA štandardu ako nástroj pre ORM (odporúčaná implementácia Hibernate)
- Využitie in-memory databázy (napr. H2)
- Unit testy, ktoré pokrývajú aspoň 90 % zdrojového kódu
- **Riešenia s DAO vrstvou implementovanou cez Spring Data JPA nebudú akceptované**
- Všetky endpointy budú začínať prefixom */api*
- Odovzdanie projektu pomocou odkazu na Git repozitár, v ktorom bude aj UML diagram a CLASS diagram
- **Odkaz na Git repozitár pošlite na adresu *halmo@inqool.cz* do 14 dní od obdržania tejto úlohy.**

## Bonusy

- Využitie liquibase knižnice pre správu databázových zmien
- Využitie knižnice lombok pre minimalizovanie boilerplate kódu
- V prípade zlých dotazov (napríklad na vytvorenie nevalidnej rezervácie) vráti REST volanie chybový kód 400
- Zabezpečenie REST endpointov
  - Implementovať správu užívateľov
  - 2 role:
    - USER – má prístup iba k READ operáciám, s výnimkou založenia rezervácie
    - ADMIN – má prístup ku všetkým operáciám
  - Všetky endpointy budú zabezpečené cez JWT token (s výnimkou */api/auth/\**)
  - Prihlásenie bude umožnené cez Basic autentizáciu volaním POST dotazu na */api/auth/login*

- Pokiaľ bude prihlásenie úspešné, systém vráti v odpovedi v autorizačnej hlavičke prístupový token (JWT token)
  - Secret pre JWT token bude súčasťou externej konfigurácie
  - Dĺžka platnosti prístupového tokenu a refreshu bude taktiež súčasťou externej konfigurácie
  - Pokiaľ dotaz neobsahuje autorizačnú hlavičku, prístupovému tokenu v hlavičke vypršala platnosť alebo je prístupový token neplatný, systém vráti status kód 401
  - Pokiaľ budú volané endpointy, na ktoré užívateľ nemá dostatočnú rolu, systém vráti status kód 403