

OOP CVIČENIA

Cvičenie 10

10.1 Argumenty

Vytvorte nový Spring projekt s jedným controllerom, ktorý umožní používateľovi zadať svoje meno a vráti mu textový pozdrav. Vyskúšajte si všetky tri možnosti prijímania argumentov - @RequestParam, @PathVariable a @RequestBody. Pre prijímanie argumentov cez telo nepoužite GET request. Implementáciu si overte cez aplikáciu Postman.

10.2 Spring Service & Dependency Injection

Upravte váš controller tak, aby miesto String-u vracal nejaký objekt. Vytvorte vo vašom projekte Service, ktorá bude vytvárať telá odpovedí. Prepojte váš controller a vašu service pomocou dependency injection.

10.3 Jednoduchá databáza

Vytvorte service, ktorá bude fungovať ako jednoduchá databáza. Pre ukladanie záznamov použite triedu Map, ktorej kľúče budete generovať nejakým spôsobom sami. Hodnoty v "databáze" budú nejaké objekty, ktoré si vymyslíte/zvoľte sami - napr. knihy / položky nákupného zoznamu / kontaktné údaje osôb.

Implementujte controller, ktorý umožní vykonávať základné CRUD (Create, Read, Update, Delete) operácie nad vašou "databázou". Pri implementácii dodržte zásady REST architektúry, dbajte predovšetkým na použitie správnych typov HTTP requestov pre správne operácie.

Ak si neviete s návrhom rady, môžete sa inšpirovať nasledovným rozdelením (databáza kníh):

- GET - /book - vráti zoznam všetkých kníh vrátane ich ID
- GET - /book/id - vráti knihu s daným ID
- POST - /book - v tele obsahuje údaje o novej knihe - uloží prijatú knihu do databázy a vráti jej ID
- PUT - /book/id - v tele obsahuje údaje o upravenej knihe - nahradí knihu s daným ID za prijatú knihu a vráti starú hodnotu
- DELETE - /book/id - vymaže knihu s daným ID z databázy a vráti jej hodnotu

10.4 HTTP status codes

Upravte vaše riešenie predchádzajúcej úlohy tak, aby sa pri dopyte na neexistujúce ID vrátil kód 404. Pri implementácii využite triedu ResponseEntity.