Ukázkový projekt pro Actis s.r.o.

Štěpán Moc

11.04.2023

Zadání

Zadáním je vytvoření jednoduché webové aplikace postavené na frameworku Spring Boot, která umožní pomocí API vypsat záznamy uložené v Postgres databázi.

Jako vývojové prostředí nejlépe použít IDEA (používané v Actis).

Požadavky v bodech:

1. Intellij IDEA - https://www.jetbrains.com/idea/ - studentská licence
2. Git - projekt uložit do git repozitáře (např. gitlab nebo github)
3. SpringBoot - https://spring.io/projects/spring-boot (verze 3)
4. Postgres - https://www.postgresql.org - tabulka s libovolnými záznamy (více atributů - text, číslo, boolean, datum)
5. Automaticky při startu vložit záznamy do db.
6. REST API endpoint - vypsat seznam záznamů z bodu 5. (stačí JSON)
7. Sepsat stručný návod a dokumentaci

Požadavky na instalaci

1) Intellij IDEA
2) SpringBoot
3) PostgreSQL

Instalace aplikace

1) Nainstaloval jsem vývojové prostředí Intellij IDEA se studentskou licencí
2) Vytvořil jsem si spring boot projekt, který jsem nastavil:
    a. Projekt: Maven
    b. Jazyk: Java
    c. Spring boot: 3.0.5
    d. Project Metada:
        i. Packaging: Jar
        ii. Java: 17
3) Nainstaloval jsem si databázový systém PostgreSQL

Stručný návod

1) Otevřít spring boot v Intellij IDEA
2) V application.properties nastavit připojení k databázi pomocí:
    a. Spring.datasource: url, username, password
    b. Nastavit spring.jpa.hibernate.ddl-auto=update, abychom měli v databázi aktuálně přidaná data
    c. Spring.jpa.show-sql=true, abych v konzoli viděli, jestli byly data přidána při spuštění aplikace
3) Vytvořit Třídu Product, která bude vytvářet v konstruktoru, jednotlivé produkty s atributy:
    a. Long id je id produktu a primární klíč. Má anotací @Id, @GeneratedValue (strategy = GenerationType.AUTO) a s názvem sloupce v databázi @Column (name = “product_id”);
    b. String name bude název produktu a bude mít anotaci @Column (name =”product_name”)
    c. To samé, co u Sting name bude platit pro, Integer price(cena produktu), Boolean inStock(Jestli je produkt ve skladu) a LocalDate lastStockDate(Kdy byl produkt naposled naskladněn)
4) Vytvoříme si konstruktor pro třídu Product, ale bez Long id, poté můžeme vytvoří settery a gettery pro všechny proměnné
5) Dále vytvoříme interface ProductRepositry, který bude mít dědičnost JpaRepository a parametry <Product,Long>
    a. Bude sloužit k práci s entitou Product uložené v databázi za pomocí předdefinovaných metod
6) Vytvoříme si třidu ProductServices s anotací @Services. ProductServices bude obsahovat:
    a. Vytvoříme si instanci třídy ProductRepository s anotací @Autowired
        i. @Autowired nám dá možnost využívat metody a atributy třídy ProductRepository
    b. Vytvoříme prázdný konstruktor
    c. Vytvoříme metody “setAllProducts”, která bude typu “void” a bude mít parametr List<Product> productList. Metoda bude ukládat list produktů do databáze za pomocí instance třídy ProductRepository a saveAll();
    d. Vytvoříme další metodu “findAllProducts”, s návratovým typem List<Product>. Metoda nám bude vracet výpis z databáze za pomocí instance třídy ProductRepository a findAll();
    e. s
7) Vytvoříme třídu ProductController s anotací @RestController. Přidáme do třídy @Autowired ProductServices productServices, abychom mohli pracovat s třídou ProductServices a vytvoříme prázdný konstruktor. V třídě budou následující metody:
    a. Metoda pro přidání produktu s návratovým typem “void” a bez parametrů. Metoda bude mít anotaci @PostConstructor, která nám zajistí, že se data

nahrají do databáze při spuštění aplikace. V těle metody bude seznam s produkty do kterého ručně přidáme nějaká data (id, name, price, inStock, date)

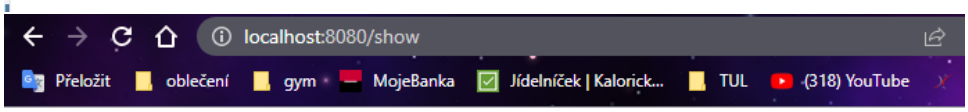
i. Datum typu LocalDate, které budeme přidávat do Productu vyřešíme metodou, kterou nám vytvoří náhodné datum a bude vracet náhodné datum formátu (“yyyy-mm-dd")
ii. Na konci metody zavoláme pomocí instance třídy ProductServices metodu .setAllProducts s parametrem našeho produkt seznamu

b. Metoda getProducts s návratovým typem List<Product> , bez parametru a anotací @GetMapping(“/show”) nám bude vracet všechny data z tabulky
   i. V těle metody bude instance třídy ProductServices, která bude vyvolávat metotodu findAllProducts();

Vzorová ukázka

Data Output | Messages | Notifications

	product_id [PK] bigint	product_in_stock boolean	product_last_stock_date date	product_name character varying (255)	product_price integer
1	1	true	2021-01-03	Mléko	30
2	2	false	2022-11-02	Chleba	50
3	3	false	2023-04-25	Kuřecí prsa	240



[{"id":1,"name":"Mléko","price":30,"inStock":true,"date":"2021-01-03"},
{"id":2,"name":"Chleba","price":50,"inStock":false,"date":"2022-11-02"},{"id":3,"name":"Kuřecí prsa","price":240,"inStock":false,"date":"2023-04-25"}]