# **Checkpoint 2**

# Meeting Room Reservation System

Verze dokumentu: 2.0.0

**Datum odevzdání:** 08.01.2023

**Název projektu:** Meeting Room Reservation System

**Odkaz na stránky projektu:** B221\_B6B36EAR/rastvdmy

**Řešitele:** Dmytro Rastvorov, Dinmukhammed Yerboluly

**Termín cvičení:** ST – 18:15

**Jméno cvičícího:** Miroslav Holeček

**Meeting Room Reservation System**

**Obsah**

* Téma
* Krátký popis očekávané funkcionality
  + Seznam hlavních funkcí aplikace
  + Komu bude výsledný systém určen
* Druhy uživatelů
  + Uživatel
  + Administrator
* Funkce pro jednotlivé typy uživatelů
  + Uživatel
  + Administrator
* Systémová omezení
* Objektový model (UML class diagram)
* Popis aplikace a její struktury
* Návod, jak aplikaci nainstalovat
* **Téma: Meeting Room Reservation System**
* Systém, který by podporoval správu jednacích místností a jejich rezervaci.
* Měla by také podporovat prioritizaci místností, aby některé místnosti byly k dispozici pouze tehdy, když v daném čase nejsou žádné jiné místnosti.

———————————————————————————————

* **Krátký popis očekávané funkcionality**

1. **Seznam hlavních funkcí aplikace:**
   * Systém správy se bude skládat z uživatelské a administrativní části:

* + - **Uživatel**
    - **Administrator**

1. **Komu bude výsledný systém určen:**

* Tento systém bude určen pro skupinu osob, které plánují schůzky v konferenčních místnostech.

——————————————————————————————

**Pro jaké uživatele (druhy uživatelů) bude systém určen (je nutné mít alespoň 2 funkčně různé uživatele):**

* Systém bude určen pro 2 typy uživatelů: Uživatel a Administrátor.
  + **Uživatel -** je uživatel, který komunikuje s aplikací pro rezervaci konferenčních místností.
  + **Administrator -** je uživatel, který spravuje rezervace konferenčních místností.

——————————————————————————————

**Jaké funkce bude plnit (k čemu bude sloužit pro jednotlivé typy uživatelů).**

* Systém bude obsahovat následující funkce pro uživatele:  
  + **Uživatel -** má možnost rezervovat místnost, zrušit rezervaci, uvidí stav místnosti a čas, kdy bude rezervována.
  + **Administrator -** má možnost přidávat místnost, odstranit místnost, změnit vlastnosti místnosti a má stejné možnosti, které má Uživatel.

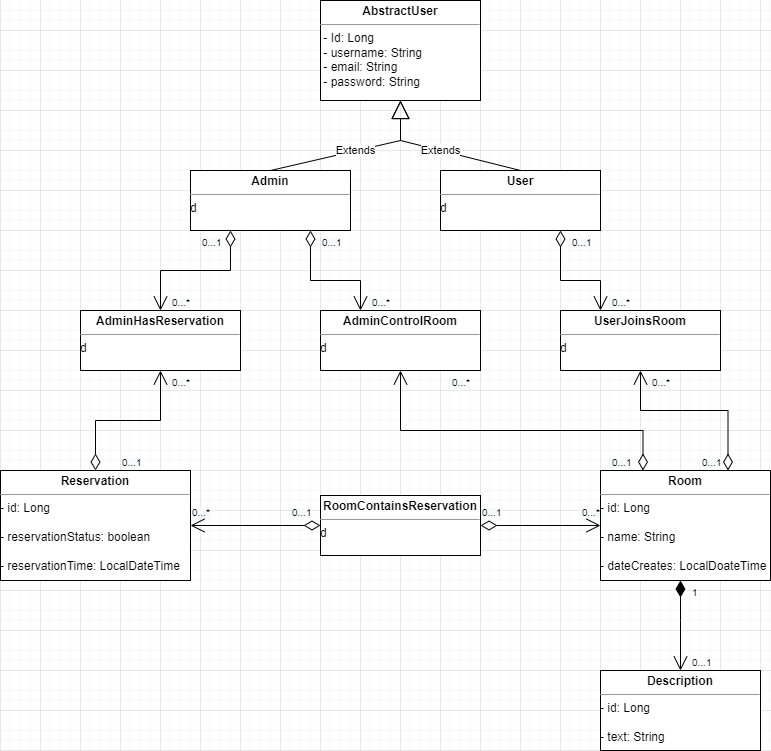
——————————————————————————————

**Jaká omezení bude systém mít (k čemu se nebude používat, i když se to dá očekávat).**

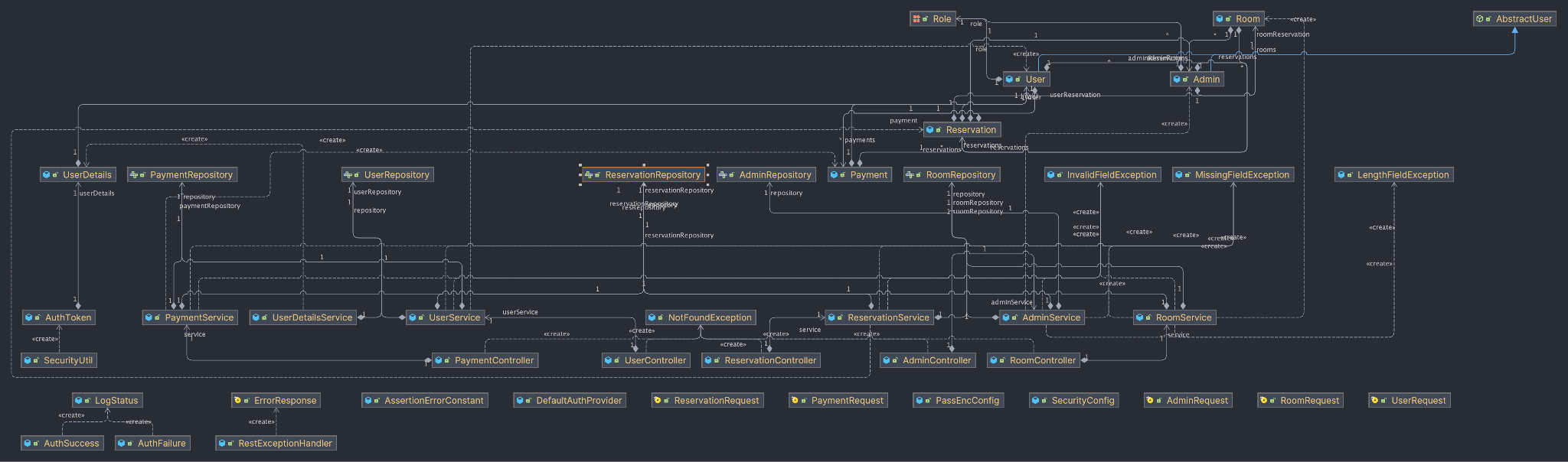
* Systém bude mít pro uživatele omezení.
  + Uživatel si nemůže:
    - Rezervovat více než 1 pokoj.
    - Vytvořit a odstranit místnost.
    - Upravit vlastnosti místnosti.
    - Zrušit jejich výhrady.
    - Zobrazit uživatele, kteří mají rezervované pokoje kromě sebe

——————————————————————————————

**Objektový model (UML diagram vytvořen pomocí** [**draw.io**](https://drawio-app.com)**)**



**Objektový model (UML diagram vytvořen pomocí** [**IntelliJ IDEA**](https://www.jetbrains.com/help/idea/class-diagram.html)**)**



——————————————————————————————

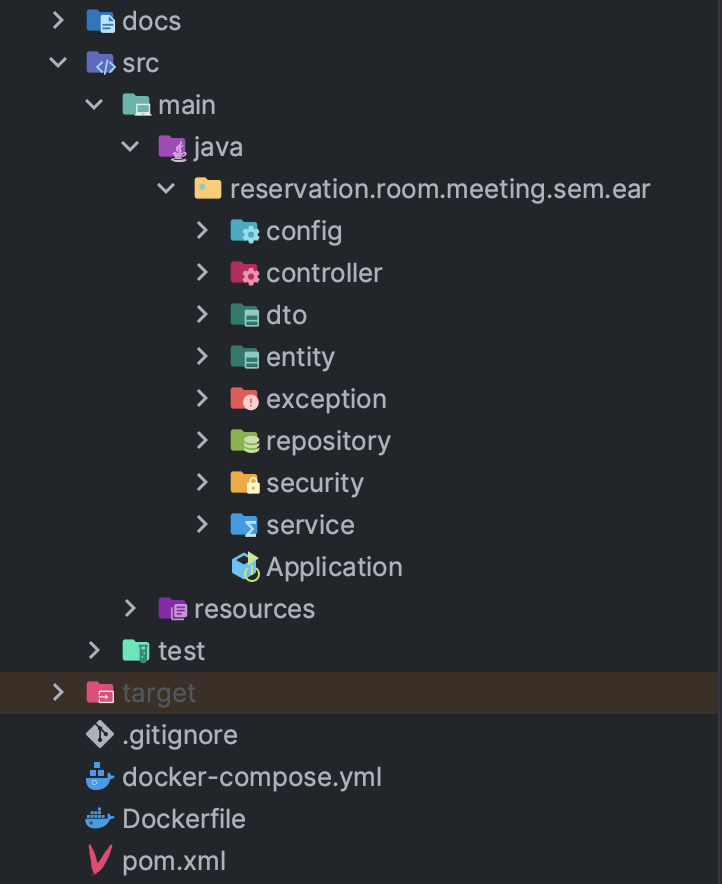
**Popis aplikace a její struktury**

Tato aplikace je severní částí pro rezervační aplikaci, kde uživatel může vytvořit rezervaci pokoje, zobrazit jeho hodinovou cenu a zjistit informace o něm.

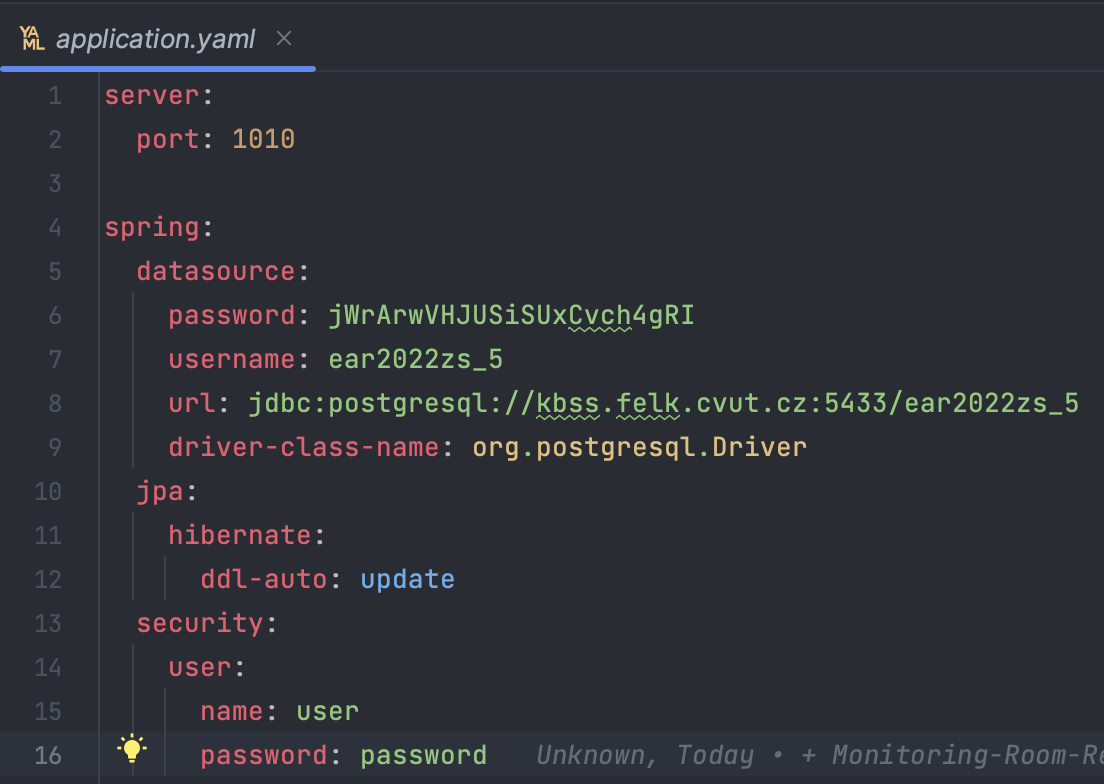
Na druhou stranu má správce mnohem více možností než uživatel. Kromě toho, co může dělat uživatel, má správce plnou volnost při vytváření místností, rezervací, uživatelů a dalších správců a může je také měnit a mazat.

Žádost se drží předchozích dvou CP, které byly sepsány a nebyly změněny.

Projekt používá **Ordering, NamedQuery**, kaskádové persist/update/merge/delete, existují také komplexní klíče.

Projekt je rozdělen do určitých složek, přičemž v každé složce je jiná část. 

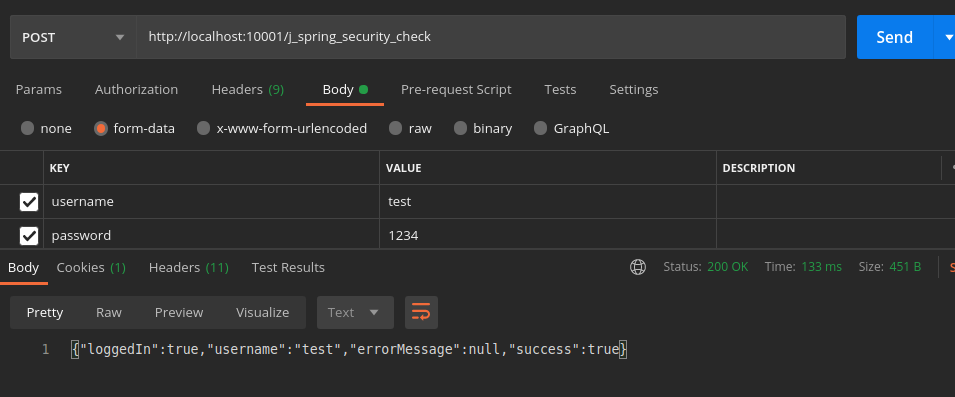
Nastavení propojení s databází lze najít ve složce **resources/application.yaml**:



Projekt má plně funkční **CRUD**, který pokrývá všechny funkce programu.

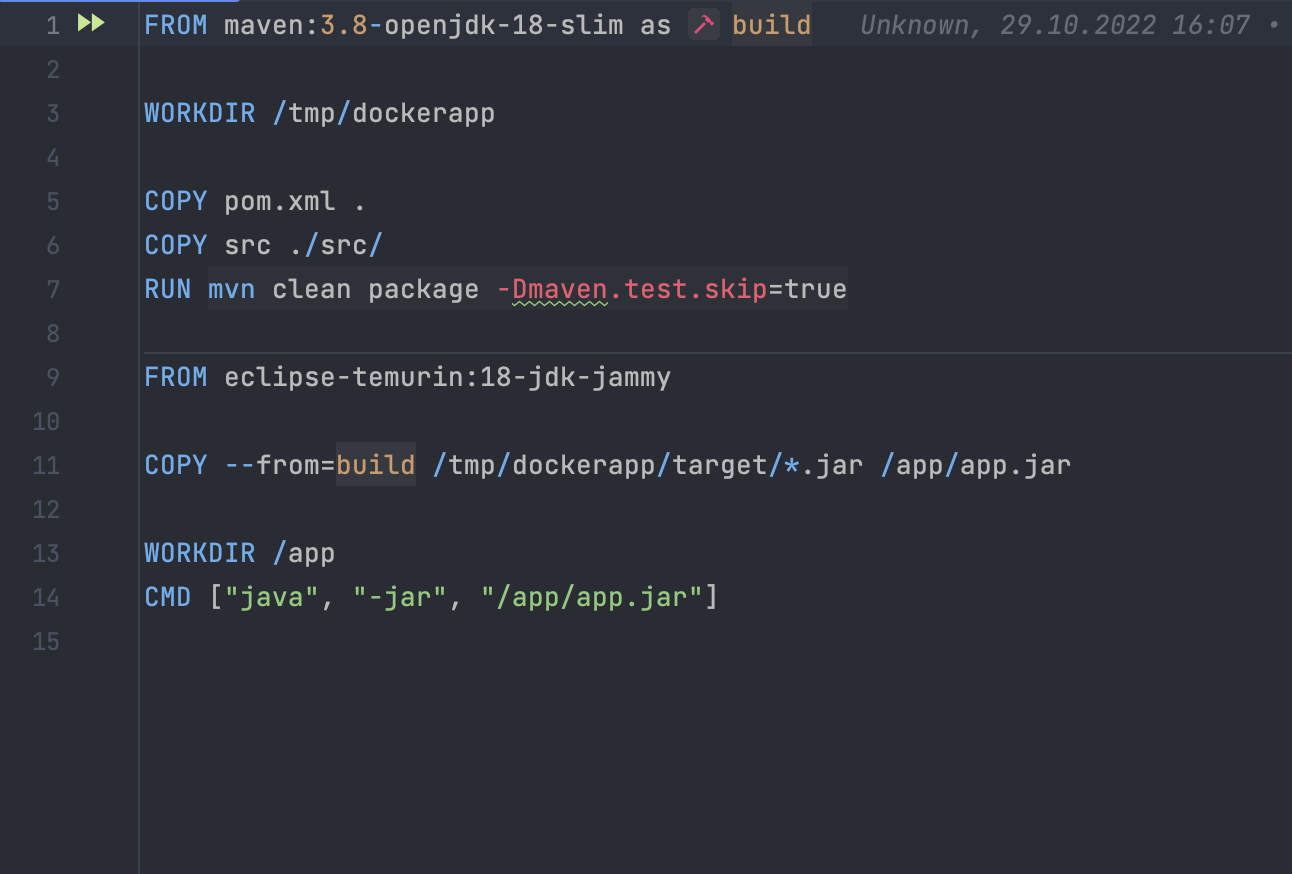
Projekt také využívá Security podpora. Aby mohl uživatel program používat, musí si vytvořit vlastní profil.

Naproti tomu správce v našem projektu již byl vytvořen a máte k němu plný přístup: uživatelské jméno: **test**, heslo: **1234**. Můžete vytvořit další správce a již máte plný přístup k funkcím:

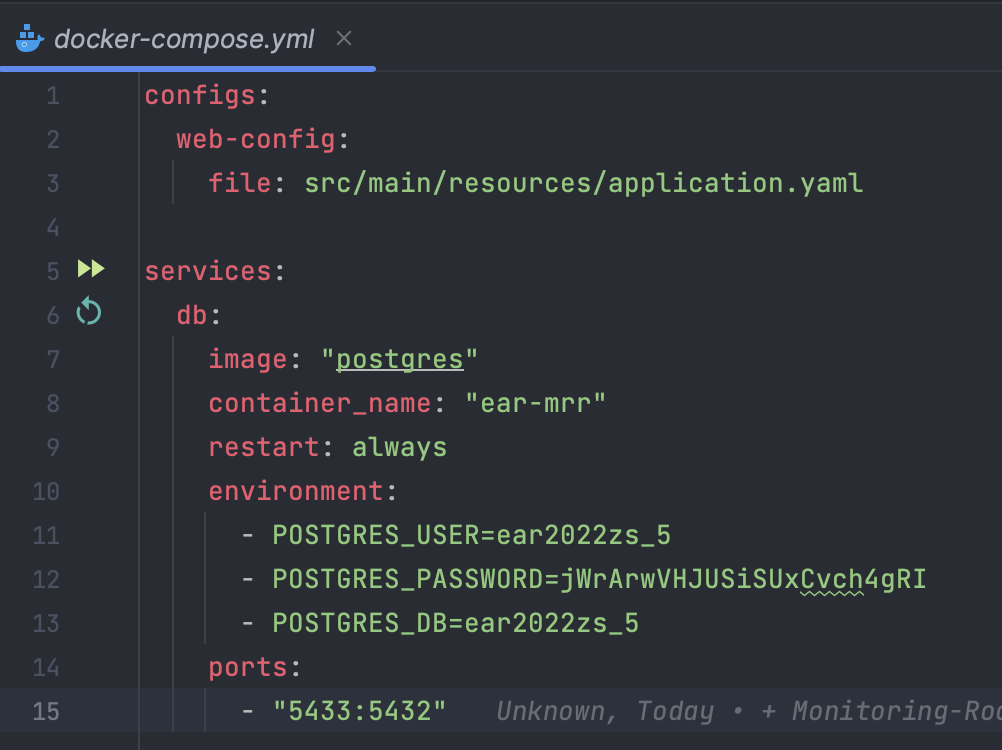


Ke kontrole požadavků REST HTTP používáme **Postman**. Můžeme jej použít k testování funkčnosti projektu a mnoha dalším věcem. Projekt využívá také **Prezentační vrstvy**.

Jako bonus byl přidán **Docker**, takže se můžeme připojit k databázi a také vygenerovat soubor Docker.



docker-compose.yml pro propojení dockeru s databází.

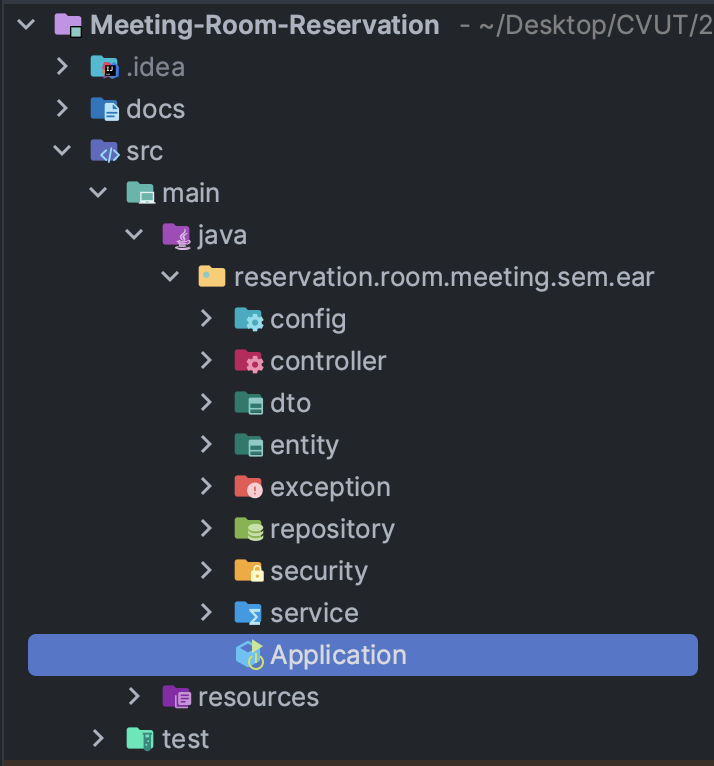
****

——————————————————————————————

**Návod, jak aplikaci nainstalovat**

**!! DŮLEŽITÉ:** *Projekt funguje ve spojení s JDK 18. Před instalací a spuštěním projektu se ujistěte, že je vaše verze aktuální.*

Program pracuje s pomocí nástroje Maven. Chcete-li projekt spustit, zkopírujte odkaz git, otevřete terminál počítače a napište **git clone** (odkaz SSH). Poté otevřete soubor projektu. Otevřete **Maven**, přejděte na položku Životní cyklus, rozbalte ji a klikněte na položku balíček. Tím se projekt vygeneruje, projde všemi potřebnými operacemi a bude připraven ke spuštění. Po dokončení tohoto procesu přejděte do složky **src/main/java/reservation/room/meeting/sem/ear** a vyberte soubor **Application.java**.



Poté klikněte na šipku spuštění a projekt se začne kompilovat. Nezapomeňte nainstalovat Postman a zkontrolovat, zda projekt funguje. Pomocí něj můžete zkontrolovat, zda projekt funguje. Projekt je nyní spuštěn a lze jej používat.

Děkuji vám za pozornost! 😌