

1. Základní informace

- Název projektu: CatParadiseHotel
 - Popis: Webová aplikace pro hotel pro kočky a jejich majitele – informuje o službě, umožňuje online rezervaci a správu rezervací přes uživatelský účet.
 - Technologie: Frontend (HTML/CSS/JavaScript), Backend (Java Spring Boot), Databáze (MySQL)
 - Verze a autor: Eliska Hrda, snapshot 0.0.1 (27.7.2025/10.8.2025)
-

2. Prostředí a spuštění projektu

- Frontend:
 - Spouští se na lokálním Node serveru příkazem
 - npm dev run
 - Backend:
 - Spouští se pomocí Maven příkazu
 - mvn spring-boot:run
 - Databáze:
 - MySQL Workbench, připojení pomocí tvého uživatelského jména a hesla
 - Propojení:
 - Frontend komunikuje s backendem přes API, volání řeší JavaScriptová logika
-

3. Backend & API

- Použitý framework: Java Spring Boot
 - Hlavní endpointy API: (např. /api/auth/login, /api/reservations, /api/users atd.)
 - Provoz: Projekt běží zatím pouze lokálně.
-

4. Databáze

- Typ: MySQL
 - Spuštění: Lokálně v aplikaci MySQL Workbench
 - Přístup: Přes vlastní user/heslo k databázi
-

5. Frontend

- Spuštění: npm dev run

- Propojení s backendem: Pomocí JavaScriptu (fetch/AJAX volání na Spring Boot REST API)
 - Poznámka: Případné změny endpointů je třeba opravit v JS kódu (např. URL API volání).
-

6. Údržba

- Aktualizace závislostí:
 - Frontend:
 - npm install
 - Backend:
 - mvn clean install
 - Řešení chyb:
 - Backend: Sleduj výstup ve Spring Boot konzoli (logy), případně restartuj server příkazem
 - mvn spring-boot:run
 - Frontend: Zkontroluj JS konzoli v prohlížeči nebo výstup v Node CLI
 - Obnova projektu: Pokud dojde k problému, zkontroluj závislosti (npm install, mvn clean install), spusť backend i frontend znovu, ověř konfiguraci DB připojení.
-

7. Bezpečnost

- Současný stav: Projekt zatím nechrání API klíče – veškeré přístupy a data jsou dostupné všem uživatelům stejně. Hesla jsou chráněna hashováním v databázi
 - Doporučení: Pro školní účely OK, při případné produkci doplnit autentizaci, autorizaci a šifrování dat/hesel.
-

8. Kontakty a verze

- Autor: Eliska Hrda
 - Verze: snapshot 0.0.1
 - Datum: poslední úprava 10.8.2025
-

- Výpis všech endpointů API
- <http://localhost:8080/api/auth/login> POST Přihlášení uživatele (login) – ověření údajů a získání přístupových práv
- <http://localhost:8080/api/auth/register> POST - Registrace nového uživatele
- <http://localhost:8080/api/cats> GET - Získání seznamu všech koček v systému

- `http://localhost:8080/api/cats` POST - Vytvoření nové kočky (zápis nového záznamu do databáze)
- `http://localhost:8080/api/cats/by-user/:userid` GET - Získání všech koček patřících konkrétnímu uživateli
- `http://localhost:8080/api/cats/:id` GET - Získání detailních informací o kočce podle ID
- `http://localhost:8080/api/cats/:id` DELETE - Smazání kočky podle jejího ID
- `http://localhost:8080/api/cats/:id` PUT - Úprava údajů o kočce dle zadaného ID
- `http://localhost:8080/api/reservations` GET - Získání všech rezervací (přehled rezervací v systému)
- `http://localhost:8080/api/reservations` POST - Vytvoření nové rezervace
- `http://localhost:8080/api/reservations/status/:status` GET - Získání rezervací podle stavu (např. aktivní, dokončené apod.)
- `http://localhost:8080/api/reservations/:id` DELETE - Smazání rezervace podle ID
- `http://localhost:8080/api/reservations/:id` PUT - Úprava existující rezervace podle ID
- `http://localhost:8080/api/reservations/:id` GET - Získání detailu rezervace dle ID
- `http://localhost:8080/api/reservations/:id/status` PUT - Změna stavu rezervace (např. schváleno, zamítnuto, čekající...)
- `http://localhost:8080/api/users` GET - Výpis všech uživatelů systému
- `http://localhost:8080/api/users` POST - Vytvoření nového uživatele
- `http://localhost:8080/api/users/by-email` GET - Získání uživatele podle e-mailu
- `http://localhost:8080/api/users/:id` GET - Získání informací o uživateli podle ID
- `http://localhost:8080/api/users/:id` DELETE - Smazání uživatele podle jeho ID
- `http://localhost:8080/api/users/:id` PUT - Úprava údajů o uživateli podle ID