



Programování Internetových aplikací

KIV/PIA-E

Bikesharing platform

Pavel Třeštík
A23N0001P

19. ledna 2024

1 Realizace projektu

Projekt se skládá z modulů backendu a frontendu. Navíc jsou použity služby ActiveMQ pro komunikaci klientů a serveru a MySQL databáze.

1.1 Backend

Backend je jeden samostatný modul. Jedná se tedy o monolit, což bylo záměrně zvoleno s ohledem na to, že projekt není příliš rozsáhlý. Struktura modulu je pak dělena na 3 vrstvy: data, business (logic), presentation (communication). Ačkoliv je modul monolit, tak použitá 3 vrstvová architektura stále zaručuje rozumnou strukturu projektu.

Produkt je implementován použitím Java17, Spring Boot 3 a Flyway. K backendu je také potřeba databáze, která využívá MySQL docker image. Celý projekt je pak sestaven nástrojem Maven. Navíc je také použito několik knihoven třetích stran.

1.1.1 Konfigurace

Backend umožňuje konfiguraci důležitých property použitím konfiguračních souborů, které Spring načítá. Jedná se o soubory `bikesharing-backend/src/main/resources/application.properties` a `bikesharing-backend/src/main/resources/application.yaml`. Jednotlivé property, které je možné konfigurovat jsou okomentované v souborech, případně jejich jméno by mělo jasně vypovídat o jejich účelu.

Flyway také používá skripty z adresáře `bikesharing-backend/src/main/resources/db/migration`. Zde jsou 2 skripty, první inicializuje strukturu databáze a druhý inicializuje základní data, která mohou být použita k testování aplikace.

1.2 Frontend

Frontend je také samostatný modul. Protože frontend je implementovaný využitím Angular 17, tak je dodržena struktura projektu, kterou vygeneruje Angular příkaz pro založení projektu. Tato struktura odděluje komponenty (view vrsta z šablony, stylů a příslušné služby), service a model.

Kromě čistého Angular 17 jsou použity knihovny rozšiřujících Angular například o tabulky. Jsou použity i knihovny třetích stran, například Leaflet

pro zobrazování map a markerů.

1.2.1 Konfigurace

Frontend také má konfigurační soubor, pro umožnění dynamického nastavení. Například aby se klient mohl připojit na jiný backend server. Jedná se o soubor `bikesharing-ui/src/environments/environment.prod.ts`.

1.3 Další technologie

Pro databázi je využito MySQL 8. Pro MQ je využito Apache ActiveMQ. Pro build a nasazení práce je použit Docker, resp. docker-compose. Respektive alternativa dockeru Podman, ovšem tyto nástroje jsou téměř zaměnitelné, takže funkčnost by měla být stejná.

Každý modul práce a technologie má vlastní kontajner, takže výsledek jsou tedy 4 kontajnery: `bikesharing-backend`, `bikesharing-ui`, `mysql` a `activemq`.

2 Spuštění

Tento proces je také popsán v README, trochu stručněji.

Build a spuštění by měl být jedním příkazem: **podman-compose up --build** (resp. `docker`). Tento příkaz vybuildí image pro backend a frontend projektu. Poté spustí tyto image navíc s MySQL a ActiveMQ imagi. Spuštění všech kontejnerů do naběhnutého stavu může zabrat několik desítek vteří. Build je v rámci jednotek minut.

Po spuštění by pak měla být aplikace dostupná na adrese `http://localhost:4200`, za předpokladu, že je použita výchozí konfigurace. Je doporučeno zachovat tuto adresu, protože GitHub OAuth2 aplikace vyžaduje adresu, na kterou se má přesměrovat po úspěšné autentikaci. Ačkoliv je možné tuto adresu konfigurovat soubory, tak je adresa vázána na GitHub aplikaci a při změně by tato adresa přestala odpovídat a GitHub by tím pádem zamítl jakýkoliv request. Je ovšem možné vytvořit si vlastní GitHub aplikaci pro autentikaci, kde se zadá jiná adresa aplikace a po změně konfigurace pak bude fungovat nová adresa.

2.1 Problém buildu frontend image

Při vývoji se objevil jeden problém s buildem frontend image. Tento problém nastává při kroku **npm install**, kdy tento krok "běží do nekonečna". Není znám důvod problému a nastává pouze v jedné specifické internetové síti, takže problém by pravděpodobně neměl nastat.

Problém je možné obejít tím, že se **npm install** provede na host stroji. Tedy změnit adresář do **bikesharing-ui** a spustit **npm install**. Předpoklad je, že na host stroji je dostupný nástroj **npm**.

3 Použití

Tento proces je také popsán v README, trochu stručněji. Podobný návod je také na stránce "About" aplikace.

Hlavní stránkou aplikace je mapa. Na této mapě jsou zobrazeny standy pomocí markeru. Světle modré markery s číslem jsou standy, které mají kola a může z nich započít jízda. Oranžové markery jsou standy bez kol a mohou jízdu pouze přijmout.

Při zahájení jízdy je na mapě modře znázorněna cesta jízdy a zelený marker s kolem, který simuluje uživatelskou jízdu po zvýrazněné cestě. Probíhající jízda se zobrazuje všem klientům jako šedý marker s kolem.

Standy a pozorování probíhajících jízd jsou zobrazeny všem uživatelům. Pro správné vyzkoušení je doporučeno otevřít alespoň 2 okna s klientem a alespoň v jednom spustit jízdu.

Spouštění jízdy je prováděno ze stránky z mapou, kdy po přihlášení se uživateli zobrazí 2 select boxy a tlačítko pro spuštění jízdy.

Pokud je uživatel přihlášen, tak si může zobrazit seznam svých jízd na stránce "Past rides".

Pokud je přihlášený uživatel navíc SERVICEMAN, tak může zobrazit seznam kol, které potřebují servisovat a v tomto seznamu je může označit jako servisované.

Pro testování jsou poskytnuti 2 uživatelé **test** a **test2**, který má navíc roli SERVICEMAN. Oba uživatelé mají heslo **test123**.