TravelDiary

Databazove systemy

# Popis aplikacie

Ucelom aplikacie TravelDiary je automatizovane vytvaranie cestovatelskych dennikoch, ktore archivuje, analyzuje a zdiela. Aplikacia sa sklada z dvoch hlavnych casti:

1. Aplikacne rozhranie (dalej uz len API) – sucastou projektu je vypracovanie mobilnej aplikacie, ktora na server posiela data (pouzivatel pri cestovani zadava jednotlive body, ktore sa potom posielaju na server)
2. Webove rozhranie (dalej uz len FRONTEND) – Pouzivatel ma k dispozicii webove rozhranie urcite na spravu prehlad vyletov, spravu zariadeni a zdielanie obsahu na socialnych sietach.

## Priklad vyuzitia

1. Pouzivatel vytvori vylet pomocou weboveho rozhrania alebo mobilu:
   1. Vie upravit zdielanie
   2. Moze pridat spolucestovatelov
   3. Nastavuje aktualny stav vyletu
   4. Zakladne informacie o vylete (datum prichodu a odchodu, destinacia, popis, nazov)
2. Pomocou mobilnej aplikacie pocas cesty zadava body ktore navstivy. Tieto zaznamy mozu pridavat vsetcia ucastnici vyletu podla toho ako boli zadany pri jeho vytvarani
   1. Automaticke zapisanie GPS suradnic
   2. Moze pridat fotku
   3. Zadava sa typ zaznamu (kempovanie, bivak, stopnute auto, zaujimavost, poznamka)
3. Za predpokladu ze sa zaznamy vzdy synchronizuju tak sa zobrazuju na webovou rozhreni zakreslene do mapy

# Technicka realizacia

Cele serverova cast (API a FRONTED) je napisana na PHP frameworku Symfony2. Na abstrakciu od databazovej vrstvy sa pouziva ORM Doctrine2. Klientska cast je napisana na Materialize SCSS frameworku.

Projekt je rozdeleny na 3 casti (bundles):

1. ApiBundle – obsahuje spravanie sa aplikacneho rozhrania
2. CoreBundle – obsahuje konfiguraciu ORM mapovania, databazove entity a objekty formularov
3. InterfaceBundle – aplikacna logika weboveho rozhrania

## Aplikacne rozhranie

Aplikacne rozhranie je napisane podla navrhoveho vzoru REST a vymiena data pomocou JSON formatovania dat. Autorizacia je riesena pomocou OAuth mechanizmu.

Fotografie sa posielaju pomocou BASE64 kodovania.

Kompletna dokumentacia k API rozhraniu je publikovana tu: <http://docs.traveldiaryapi.apiary.io/>.

### API Monitor

Aby sme zlepsili prehlad na aktivitou aplikacneho rozhrania, naprogramovali sme jednoduchy API monitor (realizacia cez Redis NoSQL). Ukladame cas, metodu a URI vykonanje poziadavky. Ukazujeme aktivitu za posledny tyzden a poslednych 20 dotazov za aktualny den.

## Webove rozhranie

Hlavna cast weboveho rozhrania je poskytnutie prehladu o vyletoch a jeho zaznamoch. Pri vykreslovani mapy sa vyuziva Google API (konkretne Google Maps Javascript API – mapa samotna a Google Geolocation API – preklada suradnice na adresy).

Rozhranie tiez poskytuje CRUD operacie nad vyletmi, pouzivatelmi (registracia) a zariadeniami. Vytvaranie zaznamov o vylete nie je sucastou weboveho rozhrania.

## Databaza

### Relacna databaza

Pri navrhu a implementacii aplikacie sa pocita s pouzitim MariaDB server v 10 verzii. Databazova struktura je opisana diagramom v prilohe (\_\_docs/EERv5.png). Dump sa nachadza v priecinku \_db. Informacie o polohe sa ukladaju do dateveho typu POINT aby v buducnosti nebol problem s vyhladavanim na zaklade pozicie.

### Redis

V projekte pouzivame aj nerelacnu Redis databazu. Vyuzitie sme nasli hlavne v oblasti cachovania zaznamov.

Pre pracu s Redis na Symfony frameworku pouzivame [SncRedisBundle](https://github.com/snc/SncRedisBundle), ktory pouziva klientsku PHP kniznicu [Predis](https://github.com/nrk/predis).

#### Cache

Redis pouzivame na ukladanie ORM mapovania pre Doctrine modely a na ukladanie PHP Sessions. Tuto funkcionalitu nam poskytuje SncRedisBundle.

Dalej sme doprogramovali cachovanie vysledkov z [Google Geolocation API](https://developers.google.com/maps/documentation/geolocation/intro), ktore zobrazujeme v detailoch vyletu. Znizujeme tak pocet poziadaviek na Google API. Jedna sa o klasicku key-value tabulku. Z GPS suradnic vytvorime MD5 hash, ktory sa pouzije ako kluc.

*Geolocation:<MD5(lat:lon) = Trolltunga, Norway>*

#### API Monitor

Vsetky data z API Monitoru sa ukladaju do Redis databazi ako list. Zaznamy ukladame na zaklade datumu.

*Api\_requests:<Y-m-d> = Redis list*

# Aktualny stav

Aplikacie je momentalne v stadiu vyvoja a preto nie su spristupneje jej vsetky casti.

1. Aplikacne rozhranie je plne funkcne
2. Webove rozhranie
   * Zoznam vyletov + vyhladavanie
   * Detail vyletu, vykreslovanie trasy na mapu, prekladanie GPS suradnic na citatelny text
   * Device manager: pouzivatel moze spravovat svoje zariadenie pripojene na API
   * API monitor: jednoduchy prehlad na API poziadavkami
   * Autorizacia