Chip and Dale

Dokumentacja do etapu drugiego

Zarys fabularny

Chip i Dale. Bohaterowie jednej z kreskówek wpadli na pomysł, by posiadać aplikację na telefon, która pozwoli im zapisywać i znajdować miejsca, gdzie ukryli smakołyki na czarną godzinę. Chip przezornie chowa po całym mieście sery na czarną godzinę. Niestety Dale nie potrafi namierzyć, gdzie zostały pozostawione.

Architektura

Wstępnie zdecydowaliśmy się na klasyczny podział trójwarstwowy, omawiany na wykładzie.

Web

Zadaniem warstwy web jest obsługiwanie zapytań http. Zapytania są obsługiwane przez klasy opatrzone adnotacją @RestController. Wstępnie planujemy skorzystać z wzorca REST i przesyłać dane pomiędzy serwerem i klientem w formacie JSON.

Service

Warstwa pośredniczy pomiędzy kontrolerem i bazą danych. Zawiera funkcje biznesowe. Reprezentują ją klasy z adnotacją @Service.

Repository

Warstwa zapewniająca wygodny dostęp do bazy danych. Jej interfejsy, oznaczone adnotacją @Repository, rozszerzają CrudRepository, co jest o tyle wygodne, że Spring Data implementuje podstawowe operacje bazodanowe za nas. Najprawdopodobniej nie będziemy musieli napisać w kodzie źródłowym ani jednej linijki SQL.

Funkcjonalnosci

1. Zapisywanie współrzędnych geograficznych schowka ze smakołykami.
2. Zapisywanie opisu schowka, np. jego zawartość.
3. Wyświetlanie na mapie znaczników opisujących lokalizację schowka.
4. Wyznaczanie trasy do najbliższego schowka.
5. Wyznaczanie trasy do wybranego schowka.
6. Oznaczenie smakołyków jako zjedzone.

User Stories

1. Jako użytkownik chcę móc zaznaczyć miejsce ukrycia smakołyku na mapie, żeby inni mogli go potem odnaleźć.
2. Jako użytkownik chcę móc wyświetlić na mapie wszystkie miejsca, w których ktoś ukrył smakołyki, żeby móc zdecydować, który chcę odnaleźć.
3. Jako użytkownik chcę żeby domyślnie GPS kierował mnie do najbliższego smakołyka, żebym mógł szybko odnaleźć smakołyk w razie głodu.
4. Jako użytkownik chcę móc zmienić miejsce do którego kieruje mnie GPS, żeby móc wybrać bardziej smakowity kąsek.
5. Jako użytkownik chcę móc oznaczać smakołyki jako zjedzone, żeby inni nie tracili czasu na odkrywanie splądrowanych kryjówek.

Wykaz technologii

I. Front-End

Bootstrap, AngularJS

II. Back-End

Spring Boot, Spring Framework Web, Spring Data JPA

III. Testy

JUnit

IV. Bazy danych

produkcyjna: MySQL (freemysqlhosting)

testowa: H2 (uruchamiana w pamięci RAM na czas testów)

V. Chmura

OpenShift

Opis zaimplementowanej funkcjonalnosci

Po uruchomieniu strony w przeglądarce, skrypt po stronie klienta wysyła zapytanie do serwera aplikacji ze ścieżką /display. Obiekt klasy StashController wywołuje metodę getAllStashes obiektu warstwy usługowej (StashService). Ten z kolei wywołuje metodę findAll na obiekcie klasy StashRepository, który komunikuje się z produkcyjną bazą danych i zwraca kolekcję obiektów klasy Stash. Kolekcja jest następnie przekazywana z powrotem do kontrolera, który zwraca ją do klienta w formacie JSON. Aplikacja klienta odczytuje dane i wyświetla je na ekranie.

Deploy aplikacji na serwerze

Travis będzie deployował aplikację na serwerze, po każdym mergu na głównej gałęzi repozytorium. Travis wysyła źródła na serwer OpenShift. Na serwerze są one budowane i aplikacja jest uruchamiana. Po tych działaniach można już oglądać nową wersję aplikacji na stronie internetowej.

Testy integracyjne

Testy integracyjne będą sprawdzały poprawność operacji na rekordach produkcyjnej bazy danych - wstawienie rekordu, select na tabeli, usunięcie rekordu i kolejny select.

Poza tym w testach integracyjnych będziemy sprawdzali współpracę aplikacji z Google Maps API.

Podzial zadan

Marta

- Front-End: AngularJS, Bootstrap

Michał

- Konfiguracja chmury: OpenShift

Bartek

- Baza danych: MySQL i testy integracyjne

Julian

- Back-End: Spring i testy jednostkowe

Materialy szkoleniowe

- Spring Framework Reference Documentation
- Spring Data JPA Reference Documentation
- Travis: https://docs.travis-ci.com/
- OpenShift:
    - https://blog.openshift.com/
    - https://docs.travis-ci.com/user/deployment/openshift/
- Google Maps API: https://developers.google.com/maps/tutorials/
- Bootstrap: http://getbootstrap.com/components/