

Zadanie rekrutacyjne fireTMS

Hej, witamy w procesie rekrutacyjnym do zespołu fireTMS. Mamy dla Ciebie zadanie które pozwoli Ci się sprawdzić z przykładowym problemem jakim mierzymy się w ramach naszych codziennych prac. Zanim przejdziemy do sedna zadania, jeszcze parę informacji. Na rozwiązanie zadania masz 1 tydzień od momentu otrzymania wiadomości. Rozwiązanie prosimy umieścić w repozytorium i przesłać nam adres url.

Zadanie:

W ramach zadania należy stworzyć aplikację, która umożliwi zarządzanie pojazdami oraz naczepami.

Założenia do pojazdów:

- pojazd posiada unikalny numer rejestracyjny,
- każdy pojazd musi mieć wypełniony numer rejestracyjny podczas tworzenia,
- metody służące do tworzenia, edycji i usuwania obiektu,
- mamy możliwość pobrania za pomocą listy pojazdów wraz z przypisanymi naczepami dla miesiąca wstecz od aktualnej daty wraz zakresami czasowymi w których te przypisanie obowiązuje,
- listę możemy sortować i filtrować między innymi po zakresie kiedy przypisana jest naczepa,

Założenia do naczep:

- naczepa posiada unikalny numer rejestracyjny,
- każda naczepa musi mieć wypełniony numer rejestracyjny podczas tworzenia,
- metody służące do tworzenia, edycji i usuwania obiektu

Założenia do przypisywania naczep:

- istnieją metody do zarządzania przypisywaniem naczep do pojazdów,
- początek i koniec okresu przypisania może być określony lub też nie,
- początek i koniec okresu przypisania powinien być reprezentowany przez obiekt typu Instant,
- jedna naczepa w danym momencie może być przypisana tylko do jednego pojazdu,
- operacja przypisania powinna zweryfikować czy w danym momencie dany pojazd nie ma przypisanej naczepy i poinformować użytkownika API w odpowiedni sposób jeśli istnieje przypisanie,
- za pomocą dedykowanej osobnej operacji umożliwiamy wprowadzenie na dany okres naczepy nawet jeśli w tym momencie pojazd ma przypisaną inną naczepę.

Odbywa się to poprzez modyfikację aktualnego przypisania np.:

- skrócenie - gdy aktualne przypisanie zaczyna lub kończy się w nowym zakresie,
- podzielenie na dwie części - gdy nowe przypisanie znajduje się wewnątrz aktualnie trwającego okresu,
- usunięcie - gdy nowy okres w pełni obejmuje aktualnie trwające przypisanie.
- dla dedykowanej operacji możemy określić parametr *boolean* "crossHitch" - który oznacza, przepinkę na krzyż i oznacza, że naczepy będą przepinane pomiędzy

pojazdami na podany okres. *Pomiędzy pojazdami* - oznacza iż jeśli pojazd P1, ma aktualnie przypisaną naczepę N1, a pojazd P2 ma aktualnie przypisaną naczepę N2 to gdy chcemy przełączyć naczepę N2 do pojazdu P1, to naczepa N1 zostanie na ten sam okres przypięta do pojazdu P2 po tej operacji. Brak ustawienia tego parametru spowoduje, iż przy podobnej operacji do pojazdu P1 przypiszemy naczepę N2, jednak do pojazdu P2, nie przypiszemy naczepy N1 z pojazdu P1 (naczepa N1 pozostaje bez przypisania na ten okres).

Dodatkowe założenia do aplikacji:

- aplikacja napisana jest w Javie z użyciem Spring Boot'a,
- wszystkie operacje możemy wywołać przez Rest API które jest udokumentowane za pomocą Swaggera,
- api jest zabezpieczone inMemory userem,
- aplikacja posiada 2 profile:
 - **test** - oparty na bazie h2,
 - **prod** - oparty na bazie postgresSQL
- kod zgodny z zasadami zwanymi "clean code",
- nie musi powstawać żaden UI,
- daty w api wprowadzamy w formacie ISO 8601,