

Напишете RESTful API, което да следва OpenAPI дефиницията от "Car Management - OpenAPI definition-1.pdf". API-то е backend service за система за управление на автомобили. Frontend, който използва това REST API, можете да намерите във "car-management-frontend.zip". Този frontend **трябва** да може да работи с Вашето API **без модификации**.

API-то трябва да предоставя набор от функционалности, които покриват управлението на сервиси, автомобили и заявки за поддръжка. Ето какво се изисква:

Управление на сервиси

Системата трябва да позволява създаването, изтриването и обновяването на информация за сервизите. Тези операции ще осигуряват възможност за добавяне на нови сервиси, редактиране на данни за съществуващи сервиси и премахването им, когато вече не са необходими. Освен това потребителите трябва да могат да извличат информация за конкретен сервиз чрез неговия уникален идентификатор (ID), което ще им даде достъп до детайлна информация за съответния обект. При необходимост от обобщена информация за всички сервиси, системата трябва да осигури извличане на данни за всички налични сервиси с възможност за филтриране по град. Допълнително, системата трябва да поддържа генериране на справки, които включват филтър за конкретен сервиз, диапазон от дати и статистика, показваща за всяка дата колко заявки има и какъв е свободният капацитет на съответния сервиз.

Управление на автомобили

Системата трябва да осигурява възможност за създаване, изтриване и обновяване на данни за автомобили. Това включва добавяне на нови автомобили към базата данни, редактиране на информация за съществуващи автомобили и изтриване на такива, които вече не са необходими. Потребителите трябва да могат да извличат информация за конкретен автомобил по неговия уникален идентификатор (ID), както и да получават обобщена информация за всички автомобили. За извличане на списък с автомобили трябва да се поддържат независими филтри по марка, обслужващ сервиз и диапазон от години на производство. Освен това всяка кола може да бъде регистрирана в няколко обслужващи сервиза, което позволява по-гъвкаво управление на връзките между автомобили и сервиси.

Управление на заявки за поддръжка

Системата трябва да поддържа създаване и обновяване на заявки за поддръжка, като при създаването или промяната на заявка се проверява наличието на свободни места в съответния сервиз за избраната дата и да връща грешка ако няма свободни места. Това ще предотврати дублиране на заявки или пренасочване към препълнени сервиси. Освен това трябва да има функционалност за изтриване на заявки, което ще позволи премахването на ненужни или неправилно въведени заявки. Потребителите трябва да могат да извличат информация за конкретна заявка по нейния уникален идентификатор (ID) и да получават списък с всички заявки с поддръжка на филтри по кола, сервиз и диапазон от дати. Допълнително, системата трябва да поддържа генериране на справки за броя на заявките по месеци, като предоставя филтри по сервиз и диапазон от месец и година вкл. Справката трябва да връща за всеки месец в

зададения период, колко поръчки са направени. **Имайте в предвид, че месеците в които няма заявки също трябва да присъстват в резултата от справката и да имат брой заявки 0.**

Тези функционалности ще осигурят ефективно управление на сервизи, автомобили и заявки за поддръжка, като същевременно предоставят полезни справки и статистики за потребителите на системата.

Всеки endpoint трябва да прави валидация на данните и да връща грешка 400 при неправилни данни в заявката. Също така при липсващ ресурс, status code 404 трябва да бъде върнат. При успешна операция status code 200.

Всички дати са във "yyyy-mm-dd" формат а месеците в "yy-mm" формат.

За да стартирате frontend-а, следвайте следните стъпки:

- Свалете файловете от https://drive.google.com/drive/folders/1Ugv4oI_al9mtdpb6CtzOc7tMXX1RkTln?usp=sharing
- Инсталирайте Node.js от <https://nodejs.org>.
- Разархивирайте car-management-frontend.zip.
- От директорията, в която сте разархивирали car-management-frontend.zip,
- изпълнете командата npm install serve за да инсталирате serve командата.
- За стартиране на frontend-а, използвайте командата **serve**.
- Frontend-ът ще бъде достъпен на <http://localhost:3000>.

Самият frontend ще се свързва към REST API-то на endpoint <http://127.0.0.1:8088>.

Всички необходими файлове може да намерите в следния Google Drive: https://drive.google.com/drive/folders/1Ugv4oI_al9mtdpb6CtzOc7tMXX1RkTln?usp=sharing или да ги свалите от заданието.

Добавен е и json файл с open-api документацията, който може да ползвате в PostMan ако желаете да тествате Вашето REST API.

За реализация на проекта, можете да използвате език или платформа по Ваш избор.

Реализираният проект трябва да бъде качен в GitHub като изходен код и споделен в Classroom заданието, след като бъде завършен. Също така проекта трябва да има поне **три** смислени **commits** в repository-то в GitHub. Няма да се приемат архиви в classroom-а, както и качени архивни файлове в GitHub.

Успех,
гл. ас. МоноВ