

# Car Management Tool

## Описание

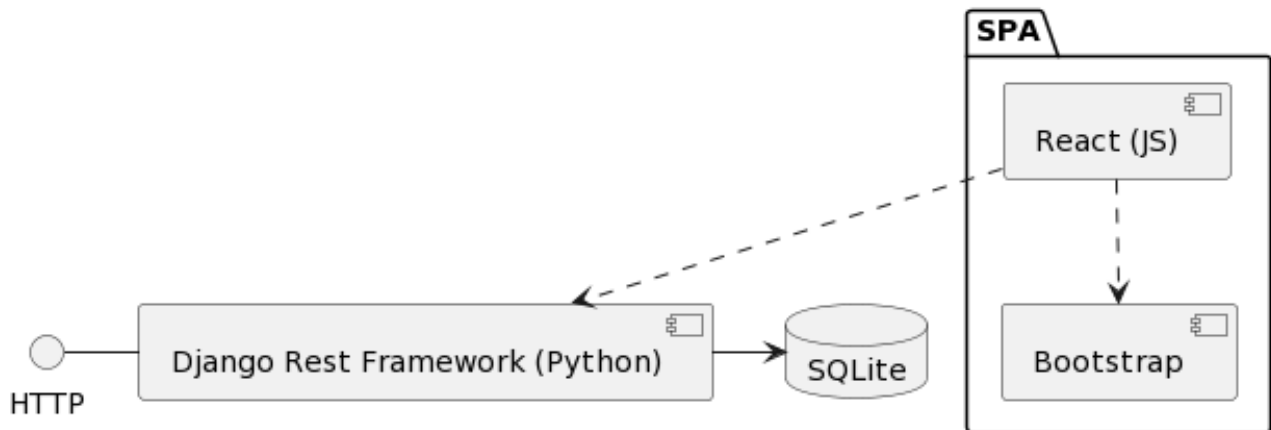
Просто АПИ, чрез което може да се оперира върху 3 модела:

1. 'Car' - [GET], [POST], [PUT], [DELETE].
2. 'CarModel' - [GET], [POST], [PUT], [DELETE].
3. 'CarMake' - [GET], [POST], [PUT], [DELETE].

## Структура на приложението

Приложението е структурирано с Backend написана на **Django & Django Ninja** (**Python**) и с Frontend част написана на **React (JS)** с **Bootstrap**.

Структурата може да бъде видяна на графиката приложена долу.



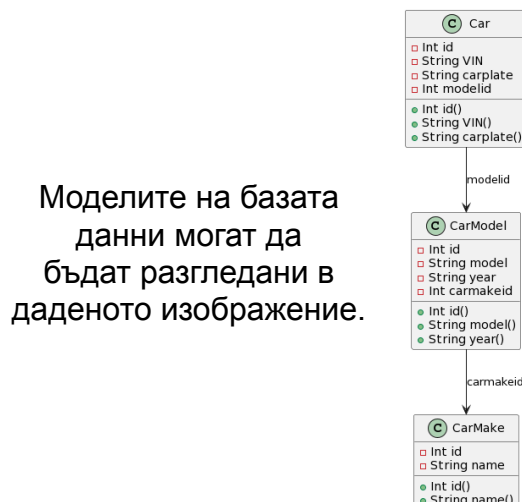
## Класове

Базата данни има три класа:

1. Car (id, VIN, carplate, modelid)
2. CarModel (id, model, year, carmakeid)
3. CarMake (id, name)

Освен това имаме валидация за дадените класове направена както в Backend-a, така и в Frontend-a:

1. VIN - трябва да е точно 10 знака, които да са само цифри.
2. Car's Plate - трябва да е 8 знака, които да са във вида 'AA1234BB'.
3. Всички релационни полета трябва да участват във всички заявки.



## Заявки

### 1. Марки на коли

GET	/api/carmakes	Прави търсене по дадени полета от базата.
POST	/api/carmakes/create/	Създава нова марка по даден шаблон.
DELETE	/api/carmakes/{id}	Трие дадена марка по дадено id.
PUT	/api/carmakes/{id}	Променя дадена марка по даден шаблон и id.
GET	/api/carmakes/{id}	Връща дадена марка по дадено id.
GET	/api/carmakes/getAll/	Връща всички марки в базата.

### 2. Модели на коли

GET	/api/carmodels	Прави търсене по дадени полета от базата.
POST	/api/carmodels/create/	Създава нов модел по даден шаблон.
DELETE	/api/carmodels/{id}	Трие даден модел по дадено id.
PUT	/api/carmodels/{id}	Променя даден модел по даден шаблон и id.
GET	/api/carmodels/{id}	Връща даден модел по дадено id.
GET	/api/carmodels/getAll/	Връща всички модели в базата.

### 3. Коли

GET	/api/cars	Прави търсене по дадени полета от базата.
POST	/api/cars/create/	Създава нова кола по даден шаблон.
DELETE	/api/cars/{id}	Трие дадена кола по дадено id.
PUT	/api/cars/{id}	Променя дадена кола по даден шаблон и id.
GET	/api/cars/{id}	Връща дадена кола по дадено id.

GET

/api/cars/getAll/

Връща всички коли в базата.

За повече яснота: след стартиране може да се види документация във вид "OpenAPI" на адрес <http://localhost:8000/api/docs>.

## Как да стартираме

1. Инсталираме **Python 3** и **Django**.
2. Инсталираме **npm**.
3. Подаваме `npm install` в ReactJS папката.
4. Подаваме `npm start` в ReactJS папката.
5. Подаваме `pip install -r requirements.txt` в Django API папката.
6. Подаваме `python manage.py runserver` в Django API папката.

Ако по някаква причина липсва **db.sqlite3** файлът: Подаваме `python manage.py migrate`, за да създадем базата, от която приложението се нуждае.