

# Техническое задание

---

## На разработку простого REST API

Вам необходимо разработать REST API для диллера автомобилей со следующим функционалом:

- У API есть 3 ресурса `/register` , `/stock`, `/auto/<string:mark>`
- **POST** `/register` - позволяет зарегистрировать нового пользователя для API. Завершается кодом **201** и сообщением `{"Message" : "User created. Try to auth"}` в случае, если такого пользователя еще не было в БД. В противном случае завершается кодом **400** и сообщением `{"Error" : "User already exists"}`. Поля пользователей выглядят так:

```
{
  "username" : "Some user",
  "password" : "Some password"
}
```

- **POST** `/auth` - возвращает JWT метку для зарегистрированных пользователей.
- Все методы ресурсов `/stock`, `/auto/<string:mark>` требуют наличия JWT метки.
- **GET** `/auto/<string:mark>` - возвращает информацию про автомобиль с именем mark и код **200**. В случае, если автомобиля нет в БД в текущий момент возвращаем `{"Error" : "Auto with that mark not found"}` и код **404**.
- **POST** `/auto/<string:mark>` - добавляет автомобиль с именем mark в БД. В случае успеха - **201** и сообщение `{"Message" : "Auto created"}`. В случае, если автомобиль с таким именем уже существует - **400** и `{"Error" : "Auto with that mark exists"}`. Структура автомобиля в теле запроса выглядит следующим образом:

```
{
  "max_speed" : 280,
  "distance" : 400,
  "handler" : "Auto Motors",
  "stock" : "Germany"
}
```

- **PUT** `/auto/<string:mark>` - обновляет информацию про автомобиль с именем mark в БД. В случае успеха - **202** и сообщение `{"Message" : "Auto updated"}`. В случае, если автомобиля нет в БД в текущий момент возвращаем `{"Error" : "Auto with that mark not found"}` и код **404**.
- **DELETE** `/auto/<string:mark>` - удаляет информацию про автомобиль с именем mark из БД. В случае успеха - **202** и сообщение `{"Message" : "Auto deleted"}`. В случае, если автомобиля нет в БД в текущий момент возвращаем `{"Error" : "Auto with that mark not found"}` и код **404**.

- `GET /stock` - возвращает информацию про все имеющиеся на данный момент в БД автомобили и код `200` в случае, если имеется хотя бы один автомобиль в наличии. В противном случае - `400` и сообщение `{"Error" : "No one autos found in DataBase"}`.

Решение должно содержать :

- необходимые модули `Go` кода для работы приложения,
- коллекцию `postman` ,
- конфиги подключения к СУБД (любой)
- миграционный репозиторий . up/down миграции
- скомпилированный исполняемый файл

Решение необходимо упаковать в `github.com` репозиторий и прислать преподавателю ссылку на репозиторий в чат.