

Волга IT - Задание полуфинального этапа

Здравствуй друг! Мы тебя приветствуем на полуфинальном этапе конкурса Волга IT в котором нам предстоит сделать проект на одной из доступных технологий.

Доступные технологии - C#, Java, Python, GO.

В данном задании вы примите участие в разработке сервиса по аренде автомобилей под названием "Simbir.GO". Сервис предлагает аренду не только автомобилей, но и мотоциклов и самокатов. Также вы можете выбрать срок аренды транспортного средства, например 1 минуту или 1 день. Желаем удачи в реализации задания 😊

Разработка приложения

Необходимо реализовать данные контроллеры:

1. Контроллер аккаунтов / Админ контроллер аккаунтов
2. Контроллер оплаты
3. Контроллер транспорта / Админ контроллер транспорта
4. Контроллер аренды / Админ контроллер аренды

В вашем решении должна использоваться база данных Postgres. Для авторизации необходимо использовать JWT. Для проверки приложения необходимо сконфигурировать Swagger и добавить в него возможность авторизации по JWT.

Порядок выполнения работы

Контроллеры необходимо реализовывать в том порядке, в котором они перечислены выше в главе [Разработка приложения](#). Админ контроллеры нужно реализовывать сразу после соответствующего контроллера. Проверка также будет осуществляться в этом порядке.

Отправка задания

1. Выполните все задание или его часть.
2. Напишите Readme.md файл, в котором укажите инструкцию по запуску вашего приложения и Url адрес до Swagger (примеры файлов будут ниже).
3. Запакуйте все в .zip архив с вашими именем и фамилией (Иван_Иванов.zip)
4. Отправляйте задание на проверку в своём личном кабинете.

Ответы на вопросы

Если у вас возникли вопросы по заданию то решите его на свое усмотрение, а мы оценим ваше решение. Если ваше решение отличается от задания то обязательно напишите все изменения в файл Readme.md.

Примеры README.md

Пример инструкции python

1. pip install -r requirements.txt
2. python run.py

URL: <http://localhost:8080/ui-swagger>

Пример инструкции java

1. ./mvnw -B package -DskipTests
2. cd target
3. java -jar app.jar

URL: <http://localhost:8080/api/swagger/index.html>

AccountController

GET /api/Account/Me

описание: получение данных о текущем аккаунте

ограничения: только авторизованные пользователи

POST /api/Account/SignIn

описание: получение нового jwt токена пользователя

body:

```
{  
  "username": "string",      //имя пользователя  
  "password": "string"      //пароль  
}
```

ограничения: нет

POST /api/Account/SignUp

описание: регистрация нового аккаунта

body:

```
{  
  "username": "string",      //имя пользователя  
  "password": "string"      //пароль  
}
```

ограничения: нельзя создать пользователя с существующим username

POST /api/Account/SignOut

описание: выход из аккаунта

ограничения: только авторизованные пользователи

PUT /api/Account/Update

описание: обновление своего аккаунта

body:

```
{  
  "username": "string",      //имя пользователя  
  "password": "string"      //пароль  
}
```

ограничения: только авторизованные пользователи, нельзя использовать уже используемые в системе username

TransportController

GET /api/Transport/{id}

описание: Получение информации о транспорте по id

ограничения: нет

POST /api/Transport

описание: Добавление нового транспорта

body:

```
{
  "canBeRented": bool,           //Можно ли арендовать транспорт?
  "transportType": "string",     //Тип транспорта [Car, Bike, Scooter]
  "model": "string",            //Модель транспорта
  "color": "string",            //Цвет транспорта
  "identifier": "string",        //Номерной знак
  "description": "string",       //Описание (может быть null)
  "latitude": double,           //Географическая широта местонахождения транспорта
  "longitude": double,          //Географическая долгота местонахождения транспорта
  "minutePrice": double,        //Цена аренды за минуту (может быть null)
  "dayPrice": double            //Цена аренды за сутки (может быть null)
}
```

ограничения: Только авторизованные пользователи

PUT /api/Transport/{id}

описание: Изменение транспорта по id

body:

```
{
  "canBeRented": bool,           //Можно ли арендовать транспорт?
  "model": "string",            //Модель транспорта
  "color": "string",            //Цвет транспорта
  "identifier": "string",        //Номерной знак
  "description": "string",       //Описание (может быть null)
  "latitude": double,           //Географическая широта местонахождения транспорта
  "longitude": double,          //Географическая долгота местонахождения транспорта
  "minutePrice": double,        //Цена аренды за минуту (может быть null)
  "dayPrice": double            //Цена аренды за сутки (может быть null)
}
```

ограничения: Только владелец этого транспорта

DELETE /api/Transport/{id}

описание: Удаление транспорта по id

ограничения: Только владелец этого транспорта

RentController

GET /api/Rent/Transport

описание: Получение транспорта доступного для аренды по параметрам

параметры:

lat: double //Географическая широта центра круга поиска транспорта
long: double //Географическая долгота центра круга поиска транспорта
radius: double //Радиус круга поиска транспорта
type: "string" //Тип транспорта [Car, Bike, Scooter, All]

ограничения: нет

GET /api/Rent/{rentId}

описание: Получение информации об аренде по id

ограничения: Только арендатор и владелец транспорта

GET /api/Rent/MyHistory

описание: Получение истории аренд текущего аккаунта

ограничения: Только авторизованные пользователи

GET /api/Rent/TransportHistory/{transportId}

описание: Получение истории аренд транспорта

ограничения: Только владелец этого транспорта

POST /api/Rent/New/{transportId}

описание: Аренда транспорта в личное пользование

параметры:

rentType: "string" //Тип аренды [Minutes, Days]

ограничения: Только авторизованные пользователи, нельзя брать в аренду собственный транспорт

POST /api/Rent/End/{rentId}

описание: Завершение аренды транспорта по id аренды

параметры:

lat: double //Географическая широта местонахождения транспорта

long: double //Географическая долгота местонахождения транспорта

ограничения: Только человек который создавал эту аренду

PaymentController

POST /api/Payment/Hesoyam/{accountId}

описание: Добавляет на баланс аккаунта с id={accountId} 250 000 денежных единиц.

ограничения: Администратор может добавить баланс всем, обычный пользователь только себе

AdminController

GET /api/Admin/Account

описание: Получение списка всех аккаунтов

параметры:

start: int //Начало выборки

count: int //Размер выборки

ограничения: Только администраторы

GET /api/Admin/Account/{id}

описание: Получение информации об аккаунте по id

ограничения: Только администраторы

POST /api/Admin/Account

описание: Создание администратором нового аккаунта

body:

```
{  
  "username": "string",      //имя пользователя  
  "password": "string",     //пароль  
  "isAdmin": bool,          //является ли пользователь администратором  
  "balance": double         //баланс пользователя  
}
```

ограничения: Только администраторы, нельзя создать аккаунт с уже существующим в системе username

PUT /api/Admin/Account/{id}

описание: Изменение администратором аккаунта по id

body:

```
{  
  "username": "string",      //имя пользователя  
  "password": "string",     //пароль  
  "isAdmin": bool,          //является ли пользователь администратором  
  "balance": double         //баланс пользователя  
}
```

ограничения: Только администраторы, нельзя изменять username на уже существующий в системе

DELETE /api/Admin/Account/{id}

описание: Удаление аккаунта по id

ограничения: Только администраторы

AdminTransportController

GET /api/Admin/Transport

описание: Получение списка всех транспортных средств

параметры:

start: int //Начало выборки
count: int //Размер выборки
transportType: "string" //Тип транспорта [Car, Bike, Scooter, All]

ограничения: Только администраторы

GET /api/Admin/Transport/{id}

описание: Получение информации о транспортном средстве по id

ограничения: Только администраторы

POST /api/Admin/Transport

описание: Создание нового транспортного средства

body:

```
{  
  "ownerId": long, //id аккаунта владельца  
  "canBeRented": bool, //Можно ли арендовать транспорт?  
  "transportType": "string", //Тип транспорта [Car, Bike, Scooter]  
  "model": "string", //Модель транспорта  
  "color": "string", //Цвет транспорта  
  "identifier": "string", //Номерной знак  
  "description": "string", //Описание (может быть null)  
  "latitude": double, //Географическая широта местонахождения транспорта  
  "longitude": double, //Географическая долгота местонахождения транспорта  
  "minutePrice": double, //Цена аренды за минуту (может быть null)  
  "dayPrice": double //Цена аренды за сутки (может быть null)  
}
```

ограничения: Только администраторы

PUT /api/Admin/Transport/{id}

описание: Изменение транспортного средства по id

body:

```
{  
  "ownerId": long, //id аккаунта владельца  
  "canBeRented": bool, //Можно ли арендовать транспорт?  
  "transportType": "string", //Тип транспорта [Car, Bike, Scooter]  
  "model": "string", //Модель транспорта
```

```
"color": "string",           //Цвет транспорта
"identifier": "string",      //Номерной знак
"description": "string",     //Описание (может быть null)
"latitude": double,         //Географическая широта местонахождения транспорта
"longitude": double,        //Географическая долгота местонахождения транспорта
"minutePrice": double,      //Цена аренды за минуту (может быть null)
"dayPrice": double          //Цена аренды за сутки (может быть null)
}
```

ограничения: Только администраторы

DELETE /api/Admin/Transport/{id}

описание: Удаление транспорта по id

ограничения: Только администраторы

AdminRentController

GET /api/Admin/Rent/{rentId}

описание: Получение информации по аренде по id

ограничения: Только администраторы

GET /api/Admin/UserHistory/{userId}

описание: Получение истории аренд пользователя с id={userId}

ограничения: Только администраторы

GET /api/Admin/TransportHistory/{transportId}

описание: Получение истории аренд транспорта с id={transportId}

ограничения: Только администраторы

POST /api/Admin/Rent

описание: Создание новой аренды

body:

```
{
  "transportId": long,           //id транспортного средства, которое взяли в аренду
  "userId": long,              //id аккаунта который будет владеть транспортом на время
аренды
  "timeStart": "string",       //дата и время начала аренды в ISO 8601
  "timeEnd": "string",         //дата и время окончания аренды в ISO 8601 (может быть
null)
  "priceOfUnit": double,       //цена единицы времени аренды (цена за минуту или за
день)
  "priceType": "string",       //тип оплаты [Minutes, Days]
  "finalPrice": 0               //финальная стоимость аренды (может быть null)
}
```

ограничения: Только администраторы

POST /api/Admin/Rent/End/{rentId}

описание: Завершение аренды транспорта по id аренды

параметры:

lat: double //Географическая широта местонахождения транспорта

long: double //Географическая долгота местонахождения транспорта

ограничения: Только администраторы

PUT /api/Admin/Rent/{id}

описание: Изменение записи об аренде по id

body:

```
{
  "transportId": long,           //id транспортного средства, которое взяли в аренду
  "userId": long,               //id аккаунта который будет владеть транспортом на время
аренды
  "timeStart": "string",        //дата и время начала аренды в ISO 8601
  "timeEnd": "string",          //дата и время окончания аренды в ISO 8601 (может быть
null)
  "priceOfUnit": double,        //цена единицы времени аренды (цена за минуту или за
день)
  "priceType": "string",        //тип оплаты [Minutes, Days]
  "finalPrice": 0                //финальная стоимость аренды (может быть null)
}
```

ограничения: Только администраторы

DELETE /api/Admin/Rent/{rentId}

описание: Удаление информации об аренде по id

ограничения: Только администраторы