



# Проект: Vehicle Service Platform



## Описание:

Сервис позволяет регистрировать пользователей и их транспортные средства, а также отправлять уведомления (например, о ТО или штрафах) между двумя микросервисами через RabbitMQ.

---



## Стек:

- **Frontend:** React + TypeScript
  - **Backend:**
    - **User Service:** Node.js + Express + PostgreSQL
    - **Vehicle Service:** Node.js + Express + PostgreSQL
    - Взаимодействие между сервисами: **RabbitMQ**
  - Docker (опционально)
  - Postman (для тестирования API)
- 



## Архитектура:

- **Микросервис 1 — User Service**
    - CRUD для пользователей
    - Отправляет сообщение в RabbitMQ при создании нового пользователя
  - **Микросервис 2 — Vehicle Service**
    - CRUD для транспортных средств
    - Получает сообщения из RabbitMQ и автоматически создает "пустую запись" для привязки ТС к новому пользователю
    - Может ассоциировать ТС с пользователем (user\_id)
-

## Функциональные требования:

### User Service:

- `POST /users` — создать пользователя
- `GET /users` — получить список пользователей
- `GET /users/:id` — получить пользователя по ID
- `PUT /users/:id` — обновить пользователя
- `DELETE /users/:id` — удалить пользователя

 При создании пользователя отправлять событие:

```
{
  "type": "USER_CREATED",
  "data": {
    "id": 1,
    "email": "john@example.com"
  }
}
```

---

### Vehicle Service:

- `POST /vehicles` — создать транспортное средство (с user\_id)
- `GET /vehicles` — список всех ТС
- `GET /vehicles/:id` — одно ТС
- `PUT /vehicles/:id` — обновить ТС
- `DELETE /vehicles/:id` — удалить ТС

 При получении события `USER_CREATED`, создавать пустое ТС:

```
{  
  "make": "Unknown",  
  "model": "Unknown",  
  "year": null,  
  "user_id": 1  
}
```

---



## UI (React):

- Страница со списком пользователей
  - Страница со списком транспортных средств
  - Форма создания пользователя
  - Форма создания ТС
  - Вывод связанных данных (ТС пользователя)
- 



## Бонусные задачи (по желанию):

- Валидация форм (React Hook Form или Zod)
- Докеризация микросервисов + docker-compose с PostgreSQL и RabbitMQ
- Авторизация (JWT)
- UI библиотека (например, MUI или Tailwind)

## Структура проекта (пример):

```
vehicle-platform/  
├── user-service/  
│   ├── controllers/  
│   ├── models/  
│   ├── routes/  
│   └── rabbit/  
├── vehicle-service/  
│   ├── controllers/  
│   ├── models/  
│   ├── routes/  
│   └── rabbit/  
├── frontend/  
│   └── src/  
│       ├── pages/  
│       ├── components/  
│       └── api/  
└── docker-compose.yml
```