

Общее описание

Сервер на C# ASP.NET отвечает за приём данных от мобильного приложения, преобразование этих данных в формат миссий для QGroundControl, а также за взаимодействие с базой данных, которая хранит состояние дронов. Сервер выполняет ключевую роль в управлении данными о дронах и координации полетов.

Основные функции сервера:

1. **Приём координат от мобильного приложения:** Сервер получает координаты, отправленные с мобильного устройства, и преобразует их в файл формата `.plane`, который используется для создания миссий в QGroundControl.
2. **Интеграция с QGroundControl:** После обработки координат сервер сохраняет сгенерированный файл миссии в папку `missions` для последующей загрузки в QGroundControl.
3. **Взаимодействие с базой данных:** Сервер хранит и обрабатывает информацию о состоянии дронов по их ID, отправляет данные в мобильное приложение и обрабатывает изменения, поступающие с него.

Основные компоненты

1. Приём и обработка координат

Сервер принимает координаты через HTTP-запросы от мобильного приложения и преобразует их в нужный формат для QGroundControl.

Этапы обработки:

1. Сервер получает координаты в виде широты и долготы через POST-запрос от мобильного приложения.
2. Эти данные преобразуются в файл формата `.plane`. В этом файле содержится информация о миссии для QGroundControl, включая координаты точки для полёта.
3. Сгенерированный файл сохраняется в специальную папку `missions` на сервере, откуда его можно загрузить в QGroundControl для планирования миссии.

2. Интеграция с QGroundControl

QGroundControl использует файлы формата `.plane` для создания и управления миссиями дронов. Сервер автоматически генерирует такие файлы на основе входных данных, предоставляемых мобильным приложением. Пользователи могут вручную загрузить эти миссии через интерфейс QGroundControl для выполнения полетов.

3. Работа с базой данных

Сервер взаимодействует с базой данных, которая хранит состояние дронов (например, "в полёте", "зарядка" и т.д.). Каждый дрон имеет уникальный идентификатор (ID), привязанный к его состоянию.

Основные операции с базой данных:

1. Чтение данных:

- Сервер отправляет информацию о всех дронах в мобильное приложение, включая их ID и текущее состояние.
- Эти данные предоставляются через GET-запрос, который выполняется при каждом запросе со стороны приложения.

2. Обновление данных:

- Мобильное приложение может отправить обновленные данные о состоянии конкретного дрона.
- Сервер обрабатывает эти изменения и вносит их в базу данных с помощью SQL-запросов, обновляя состояние дронов в соответствии с новыми данными.

Технические аспекты

1. Технологии

- **ASP.NET Core:** используется для разработки серверной части, поддерживающей работу с API и взаимодействие с базой данных.
- **Entity Framework:** применяется для работы с базой данных.
- **SQL Server или другой реляционный DBMS:** хранение информации о дронах и их состояниях.
- **HTTP API:** взаимодействие с мобильным приложением осуществляется через HTTP-запросы (POST и GET).
- **QGroundControl:** программное обеспечение для управления миссиями дронов, использующее файлы `.plane`.

2. Взаимодействие с мобильным приложением

- **POST-запросы:** используются для отправки координат и обновлений состояния дронов.
- **GET-запросы:** используются для получения данных о текущем состоянии дронов.
- Все запросы обрабатываются сервером с помощью контроллеров API, что обеспечивает гибкость и масштабируемость системы.

3. Пример структуры данных

- **Дроны:**
 - ID : уникальный идентификатор дрона.
 - Статус : текущее состояние дрона (в полёте, на зарядке и т.д.).
 - Другие параметры, такие как текущие координаты или последние задания.

4. Преобразование в формат .plane

Пример процесса преобразования:

- Полученные координаты обрабатываются в формате JSON/XML.
- Создаётся файл `.plane`, содержащий информацию о точке полёта, которая будет интерпретирована QGroundControl как цель миссии.
- Файл сохраняется в папку `missions`, откуда его можно загрузить в QGroundControl.

Возможные улучшения

- **Автоматическая загрузка миссий:** Настроить автоматическую загрузку сгенерированных файлов миссий в QGroundControl без необходимости ручной загрузки.
- **Добавление валидации данных:** Проверка корректности введенных координат и статусов дронов для предотвращения ошибок.
- **Оптимизация работы с базой данных:** Внедрение механизмов кеширования для ускорения операций чтения данных о дронах.

Заключение

Сервер на C# ASP.NET выполняет ключевые задачи по приёму координат, их преобразованию для QGroundControl, а также по управлению данными о дронах и их

состояниях в базе данных. Система обеспечивает связь между мобильным приложением, QGroundControl и базой данных, что делает процесс управления дронами удобным и эффективным.