



Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет имени первого Президента России
Б.Н. Ельцина (УрФУ)»
Институт радиоэлектроники и информационных технологий – РгИ

Промежуточный отчет по проекту:

Разработка и модернизация программного комплекса управления полётом
(документация)

Команда №3844:

Никифоров Роман, Алексей Рыжков

Курс: 3/2

Группа: РИ-310948, РИ-220911

Екатеринбург, 2024

Содержание

Предисловие	4
Список сокращений и терминов	5
1. Первичная установка и запуск сервиса	7
1.1. Установка PostgreSQL и создание базы данных	7
1.2. Установка FlightGear	12
1.3. Установка .NET SDK 8.0.204 и NodeJS	13
1.4. Конфигурация проекта и установка зависимостей	13
1.5. Запуск сервиса.....	16
2. Главная страница. Работа с сервисом и запуск симуляции.	18
2.1. Обзор страницы	18
2.2. Добавление/удаление контрольных точек.....	20
2.3. Запуск симуляции полёта	22
3. Настройка сервиса и симуляции	26
3.1. Установка самолёта СУ-57(Т-50).....	26
3.2. Установка местности России	28
4. Страница анализа данных и работа с ней	31
4.1. Переход на страницу анализа данных	31
4.2. Работа с графиками данных	33
5. The FlightGear User Interface	35
5.1. Подключение к The FlightGear User Interface	35
5.2. Описание основных разделов.....	35
5.2.1. Aircraft (Воздушное судно).....	36
5.2.2. Environment (Окружающая среда).....	36
5.2.3. Map (Карта).....	36
5.2.4. Tools (Инструменты).....	37
5.2.5. Simulator (Симулятор).....	37
6. Работа с API. Функции и запросы	39
6.1. План полёта	40
6.1.1. Получение информации о планах полета	40

6.1.2.	Получение информации о плане полета по ID	40
6.1.3.	Удаление плана полета по ID	41
6.1.4.	Создание нового плана полета	42
6.1.5.	Обновление плана полета	43
6.2.	Точки маршрута	44
6.2.1.	Получение точек маршрута	45
6.2.2.	Добавление точки маршрута	45
6.2.3.	Удаление точки маршрута	47
6.2.4.	Обновление точки маршрута	48
6.3.	Аэропорты/полосы	49
6.3.1.	Получение информации об аэропортах	50
6.3.2.	Получение информации об аэропортах по ID	50
6.4.	Запуск полёта	51
6.4.1.	Запуск симуляции полёта	51
6.4.2.	Получение статуса полёта/симуляции	52
6.4.3.	Получение параметров самолёта	53
6.4.4.	Завершение симуляции полёта	55
6.5.	Аналитика	55
6.5.1.	Получение информации о сессиях полётов	55
6.5.2.	Получение информации о плане полета по ID	56
6.5.3.	Удаление сохраненной сессии полёта	58
6.6.	Многократно используемые модели	58
6.6.1.	FlightPlanResponse	59
6.6.2.	RoutePointResponse	59
6.6.3.	RunwayResponse	60
6.6.4.	AirportResponse	61
6.6.5.	FlightPropertiesShotResponse	62
6.6.6.	FlightPropertyReadsResultResponse	64
Внешние ссылки и ресурсы		65

Предисловие

Настоящая документация распространяется на сервис, разработанный в рамках дисциплины в рамках 'Проектный практикум' по заказу АО "УПКБ "Деталь".

Данный продукт представляет собой программный комплекс управления полётом (сокращённо ПКУП). Он предназначен для планирования полёта на основе данных, вводимых пользователем, а также симулирования их в программе FlightGear в режиме реального времени.

Данный документ содержит информацию о:

- Первичной установке.
- Запуске сервиса, его настройке и работе с ним.
- API сервиса — о его запросах, их предназначении, входных данных и возвращаемых значениях.

Документация предназначена как для рядовых пользователей, так и для системных программистов, которые будут работать с API системы.

Список сокращений и терминов

1. ОС — операционная система.
2. ПКУП — программный комплекс управления полётом
3. КТ — контрольная точка.
4. Сессия — это результат симуляции, сохранённый в БД.
5. БД — база данных.
6. AI (Artificial intelligence, он же ИИ) — искусственный интеллект.
7. SQL (Structured Query Language) — это язык запросов, который применяют для работы с базами данных, структурированными особым образом.
8. API — программный интерфейс приложения. Представляет собой набор способов и правил, по которым различные программы обмениваются данными.
9. URL — адрес того или иного ресурса в сети.
10. Route-параметр — параметр, который задается пользователем для получения тех или иных данных по запросу.
11. GET — http-запрос на получение данных из сервера. Используется для извлечения информации из ресурса без его изменения.
12. POST — http-запрос на создание нового ресурса на сервере. Используется для отправки данных на сервер для создания нового объекта или выполнения какого-либо действия.
13. PUT — http-запрос на обновление существующего ресурса на сервере. Используется для изменения информации или состояния существующего ресурса.
14. DELETE — http-запрос на удаление существующего ресурса на сервере. Используется для удаления ресурса по его идентификатору или другому уникальному идентификатору.
15. Number — числовой тип данных (например 123).
16. String — строковый тип данных (например “ergf”).
17. Float — вещественный тип данных (например 12.1).
18. Bool — логический тип данных (true (истина)/false (ложь)).
19. Guid — уникальный глобальный идентификатор в виде строки (например 9d974548-7e90-4885-973a-dcdcb92c44f4).
20. Array — структура данных, хранящая набор значений.
21. [] — обозначение массива.
22. Object — структура данных типа ‘ключ-значение’.
23. {} — обозначение объекта.

24. Datetime — структура данных, представляющая собой дату и время (например 2012-04-23T18:25:43).

1. Первичная установка и запуск сервиса

Для первичной установки и запуска ПКУП понадобятся следующие утилиты и программы:

- 1) Редактор кода (в данной документации, в качестве примера будет использоваться [Visual Studio Code](#)) — для запуска программы и работы с ней.
- 2) [NodeJS](#) — для доступа к его пакетному менеджеру npm и установке зависимостей для клиентской части сервиса.
- 3) [PostgreSQL](#) — для создания базы данных, куда будут сохраняться результаты полета (Об его инсталляции более детально в описании процесса первичной установки).
- 4) [.NET SDK 8.0.204](#) — для запуска программы.
- 5) [Flightgear Flight Simulator](#) — для симуляции полёта.
- 6) Файл [createDB.sql](#). — скрипт для создания базы данных.

На данном этапе установите предпочитаемый редактор кода и файл createDB.sql. Процесс инсталляции остальных утилит будет описан ниже.

1.1. Установка PostgreSQL и создание базы данных

- а. Перейдите на официальный сайт [PostgreSQL](#) и нажмите кнопку ‘Download’. Далее выберите установщик для вашей ОС, как показано на *рисунке 1*.

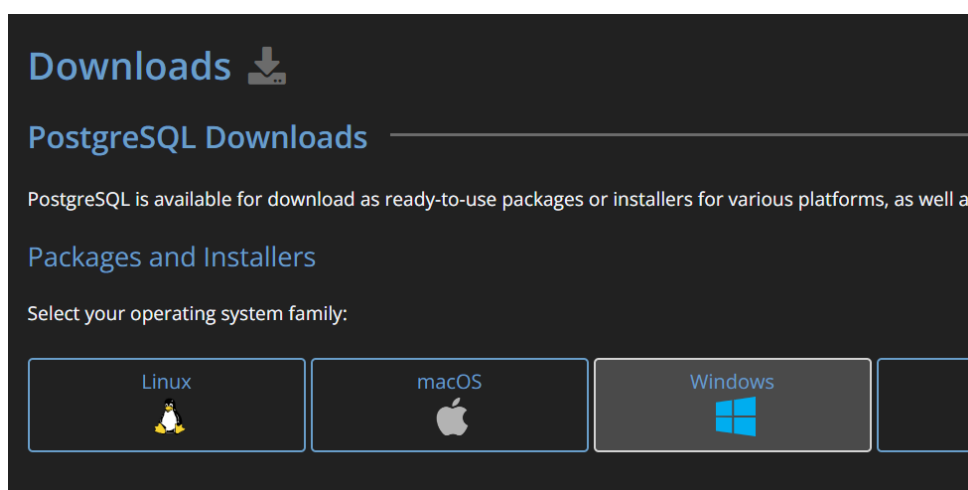


Рис.1 — Экран выбора установщика PostgreSQL для ОС

- b. На следующей странице нажмите ‘Download the installer’ и также выберите версию вашей ОС. После этого запустите установщик.
- c. Выберите путь установки (рис.2), а также выберите папку для хранения данных (рис.3).

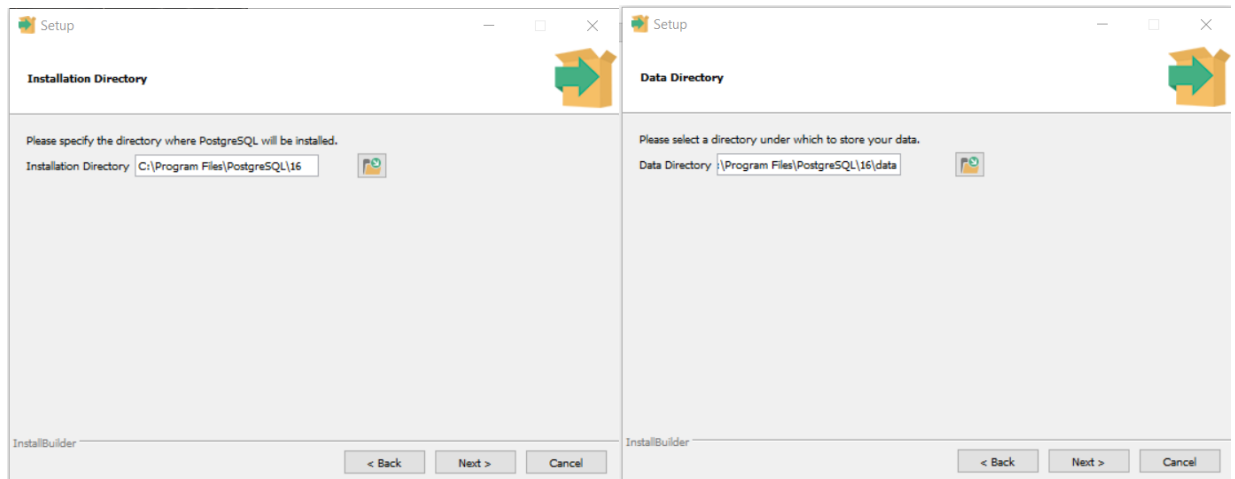


Рис.2,3 — Меню выбора пути установки и папки хранения данных

- d. Задайте пароль для пользователя postgres. Если вы пропустите этот шаг, то он будет ‘postgres’.

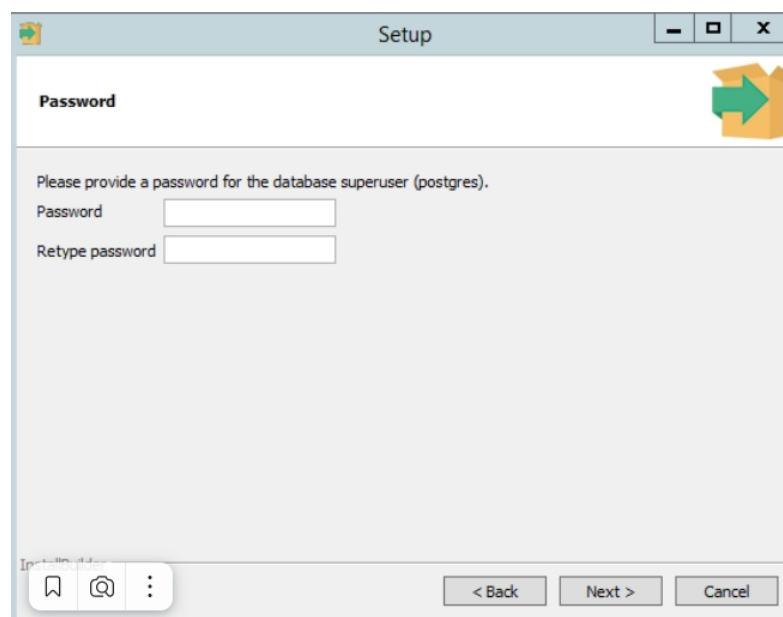


Рис.4 — Меню ввода пароля для пользователя postgres

- е. Выберите порт, на котором будет работать PostgreSQL. По умолчанию это 5432 (см. *рис. 5*), это значение понадобится для настройки конфигурации проекта.

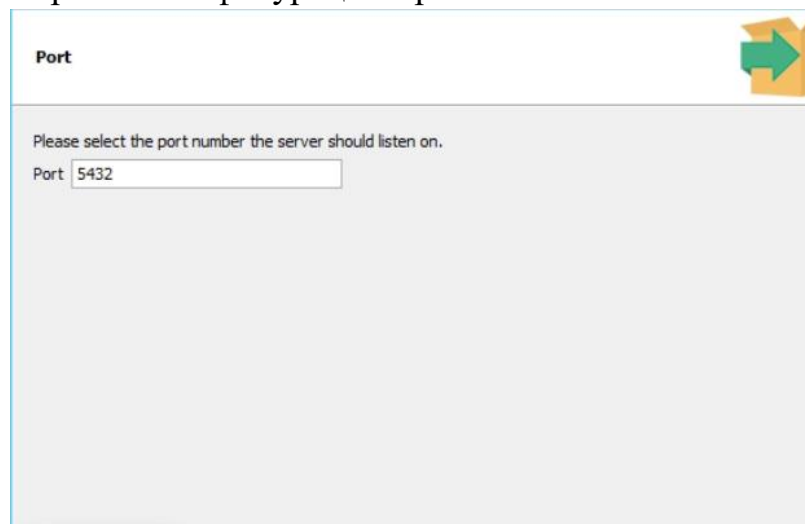


Рис.5 — Меню выбора порта для работы PostgreSQL

- ф. Нажмите далее и дождитесь завершения процесса установки.
- г. Теперь необходимо создать базу данных для хранения результатов симуляций. С PostgreSQL автоматически установилась программа pgAdmin 4 — визуальный редактор SQL. Найдите её в поиске вашей ОС и запустите его. Название и иконка представлена на *рисунке 6*.

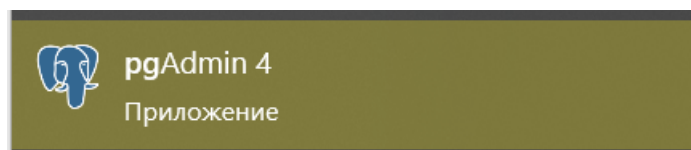


Рис.6 — PgAdmin 4

- h. В левом верхнем углу разверните меню Servers (рис.7). Введите пароль, который вы задавали при установке PostgreSQL (рис.8). По умолчанию он 'postgres'.

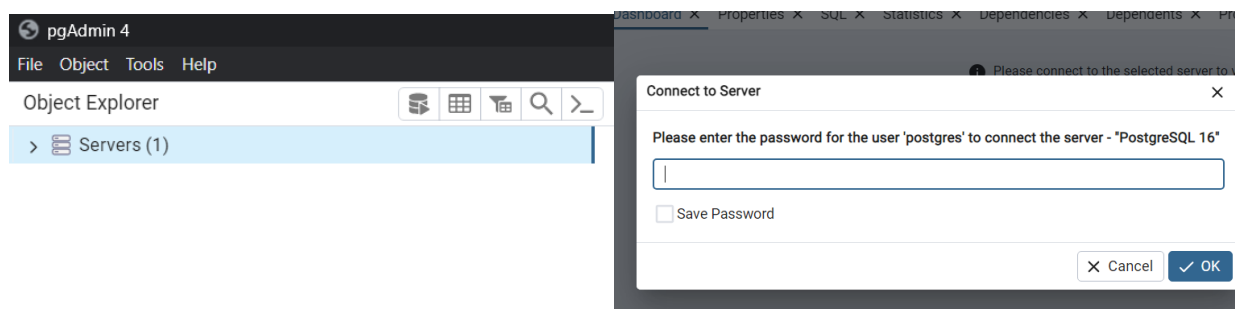


Рис.7,8 — Выбор серверов и получение доступа к ним

- i. После авторизации в том же углу нажмите правой кнопкой мыши по меню 'Databases', выберите поле 'Create' и 'Database'.

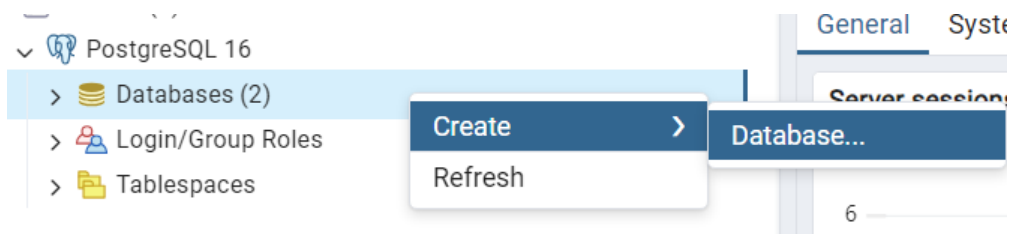


Рис.9 — Переход к базам данных (Database)

- j. Введите название вашей базы данных (Рекомендуем назвать её `flightgearapi`, чтобы при дальнейшей конфигурации проекта нужно было настраивать меньше параметров). Затем нажмите кнопку 'Save'. После этого создается пустая БД.

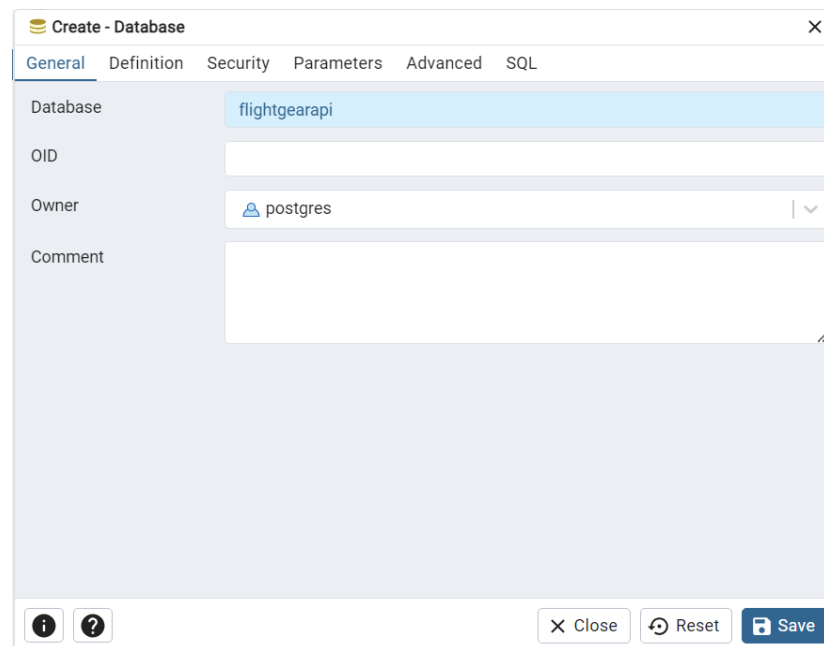


Рис.10 — Меню создания БД

- к. В том же боковом меню Servers разверните вашу БД, а затем нажмите правой кнопкой мыши по блоку 'Public' внутри меню 'Schemas' и выберите 'Query Tool' (рис.11).

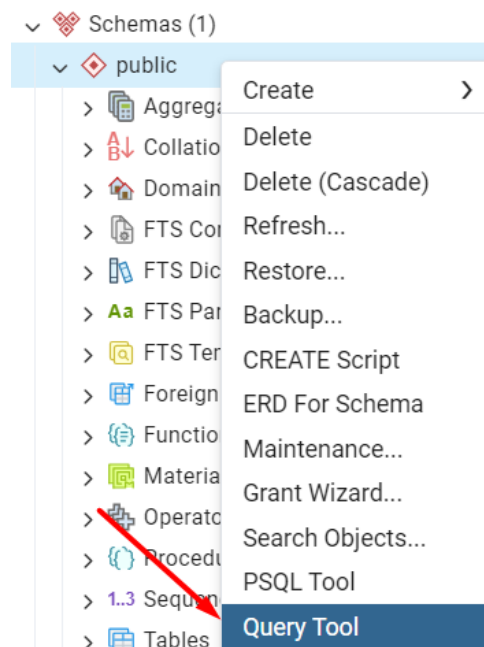


Рис.11 — Переход к Query Tool

1. Откройте файл [createDB.sql](#), который вы установили. В нём находится скрипт, необходимый для создания базы данных. Скопируйте его оттуда и вставьте в меню Query, как показано на *рисунке 12*. Затем нажмите кнопку ‘Execute script’ на панели сверху (см. *рисунок 13*). После этого ваша БД заполнится, и создастся пользователь.

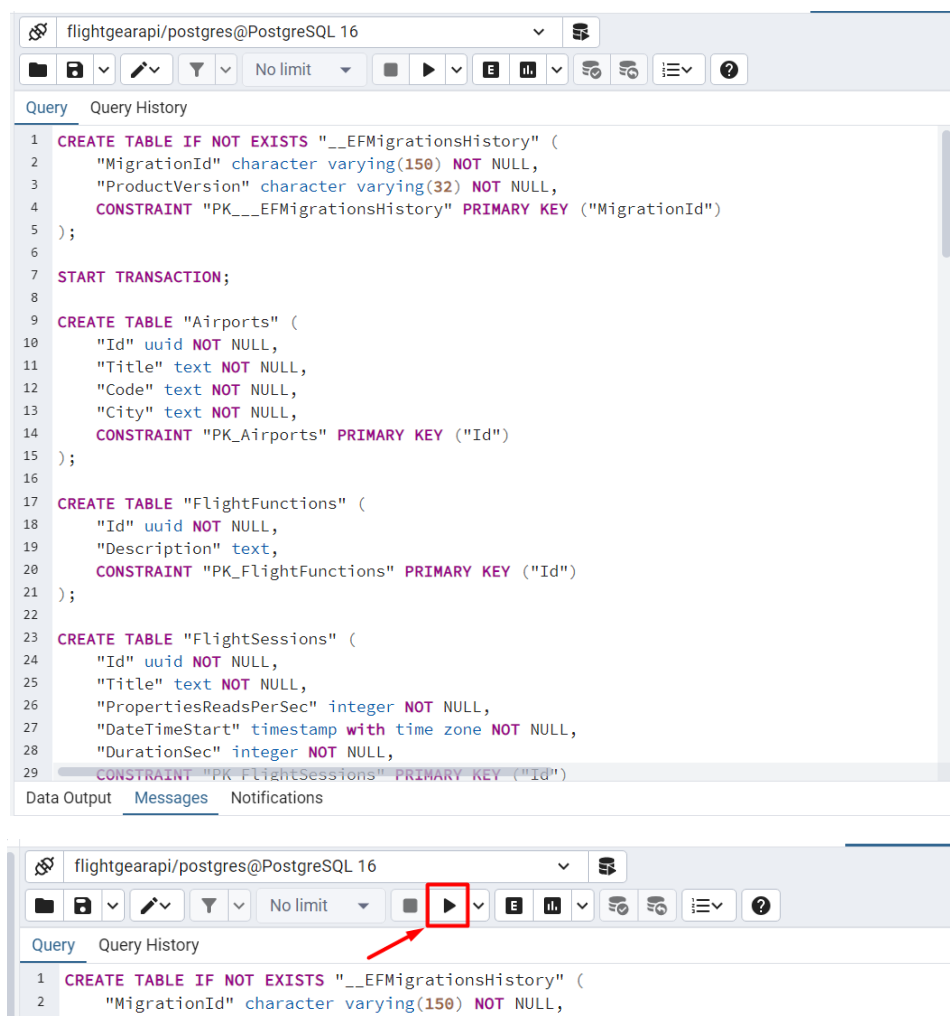


Рис.12,13 — Скрипт для создания БД и кнопка ‘Execute script’

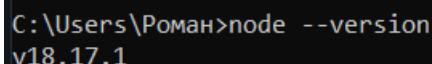
1.2. Установка FlightGear

- а. Перейдите на официальный сайт [официальный сайт FlightGear](#). Там нажмите на кнопку ‘Download Now and Fly Free!’. После перехода на другую страницу выберите вашу ОС и дождитесь загрузки установщика на ваш компьютер.

ВАЖНО: Установите FlightGear HE в папку с путём, содержащим Program Files, поскольку для работы с ней необходимы права администратора. Из-за этого могут возникнуть неполадки при запуске сервиса.

1.3. Установка .NET SDK 8.0.204 и NodeJS

- a. Перейдите на [официальный сайт .NET](#) и установите SDK 8.0.204 для вашей ОС.
- b. Перейдите на [официальный сайт NodeJS](#) и его последнюю версию. Вместе с ним также появится и пакетный менеджер npm — Node Package Manager. Чтобы проверить правильность установки, откройте командную строку и введите команду `node --version`. Если всё корректно, то выводится текущая версия, как на *рисунке 15*



```
C:\Users\Роман>node --version
v18.17.1
```

Рис.15 — Проверка установки NodeJS

1.4. Конфигурация проекта и установка зависимостей

- a. Откройте [файлы ПКУП \(FlightGearApi\)](#) при помощи вашего редактора кода. Находясь внутри, выберите файл `appsettings.json` (*рис. 16*), который находится в папке WebApi.

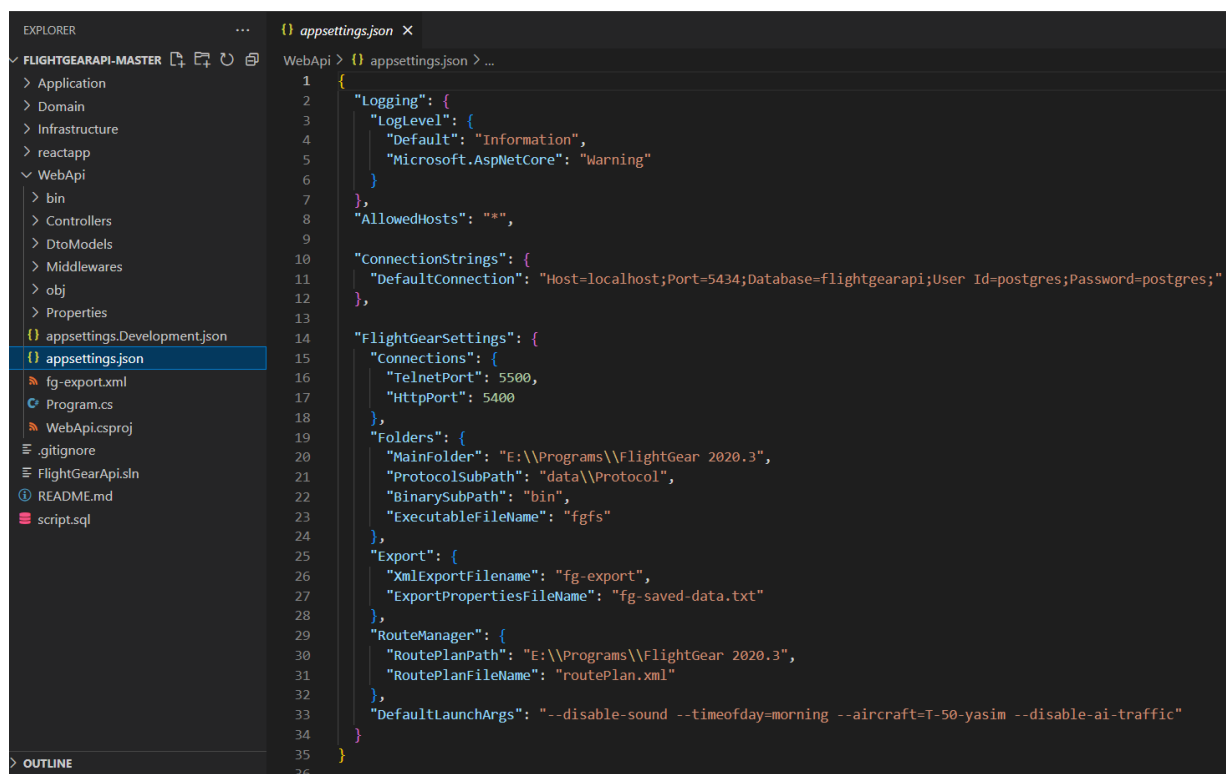


Рис.16 — Файл appsettings.json

- b. Внутри него, в разделе “Folders”, настройте параметр “MainFolder” — это путь к папке с файлами авиасимулятора FlightGear.

ВАЖНО: Из-за .json формата используйте \\ для обозначения пути к папке/файлу.

- c. В разделе “RouteManager” измените параметр “RoutePlanPath” — это путь, где будет создаваться план полёта (файл routePlan.xml). В данном примере будет указано в папку с файлами FlightGear.

```

"RouteManager": {
  "RoutePlanPath": "C:\\Users\\Роман\\Downloads\\FlightGear 2020.3",
  "RoutePlanFileName": "routePlan.xml"
},

```

Рис.17 — Раздел RouteManager

- d. В разделе “DefaultLaunchArgs” (рис.18) при желании вы можете отредактировать некоторые параметры запуска:

1. `--disable-sound` — отключает звук в симуляции FlightGear.
2. `--timeofday` — изменяет время суток в симуляции (например можно изменить на `night` для наступления ночи).
3. `--aircraft` — здесь указан летательный аппарат, использующийся в симуляции. **ВАЖНО:** НЕ изменяйте данный параметр для корректной работы ПКУП.
4. `--disable-ai-traffic` — отключает [AI трафик](#).

```
},
"DefaultLaunchArgs": "--disable-sound --timeofday=morning --aircraft=T-50-yasim --disable-ai-traffic"
```

Рис. 18 — Раздел “DefaultLaunchArgs”

Все доступные команды для изменения параметров запуска доступны [здесь](#). Учтите, что из-за некоторых команд сервис может перестать работать. Используйте команды, которые меняют только визуальную/звуковую составляющую. (Текстуры, настройки времени суток, погоды и т. д.).

- е. В разделе “ConnectionStrings” в поле “DefaultConnection” (рис.19) измените следующие параметры:

1. Port — это порт, который вы указывали при установке PostgreSQL.
2. Database — название базы данных, которую вы создали при работе с PgAdmin4.
3. User Id — имя пользователя базы данных. По умолчанию это ‘postgres’.
4. Password — пароль, который вы задавали при установке PostgreSQL. По умолчанию это ‘postgres’.

```
"ConnectionStrings": {
  "DefaultConnection": "Host=localhost;Port=5432;Database=flightgearapi;User Id=postgres;Password=password;"
},
```

Рис.19 — Настройка соединения

- ф. Перейдите в папку `reactapp` и в терминале редактора кода напишите команду `'npm i'`, как показано на [риунке. 20](#). После

ввода установятся необходимые для клиентской части сервиса зависимости.

```
PS C:\Users\Роман\OneDrive\Рабочий стол\doc\FlightGearApi-master> cd ./reactapp
PS C:\Users\Роман\OneDrive\Рабочий стол\doc\FlightGearApi-master\reactapp> npm i
```

Рис.20 — Установка зависимостей

1.5. Запуск сервиса

- а. Находясь в терминале напишите команду `npm start` — она запустит клиентскую (frontend) часть ПКУП. Если всё прошло успешно, появятся две ссылки, как показано на *рисунке 21*. Перейдите под той, которая указана в пункте ‘Local’.

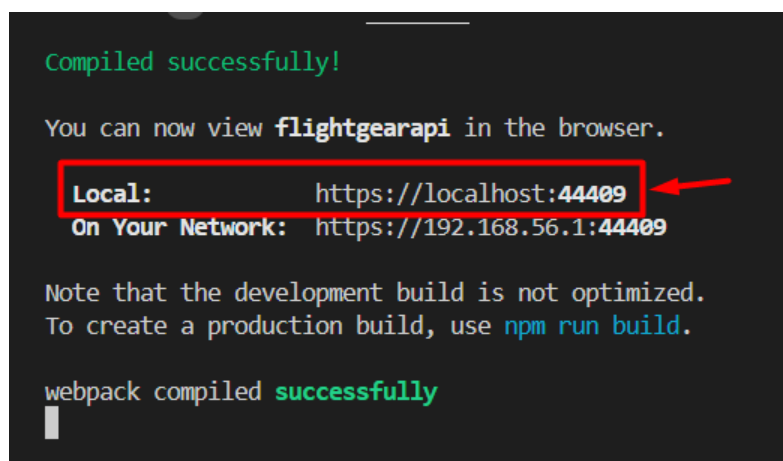


Рис. 21 — Ссылка для запуска клиентской части

- б. Для корректной работы ПКУП нужно также запустить серверную (backend) часть. В терминале перейдите обратно в главную директорию, при помощи команды `cd ../` и в верхней панели нажмите кнопку ‘Run without debugging’.

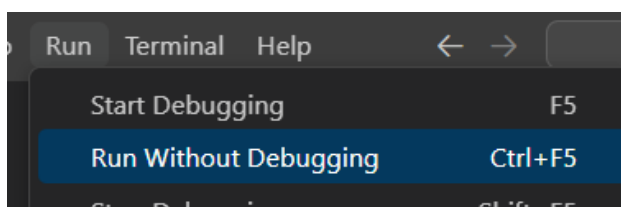


Рис. 22 — Команда запуска серверной части через редактор кода

- с. После нажатия кнопки вам будет предложен выбор компилятора. Выберите 'C#', как показано на *рисунке 23*.

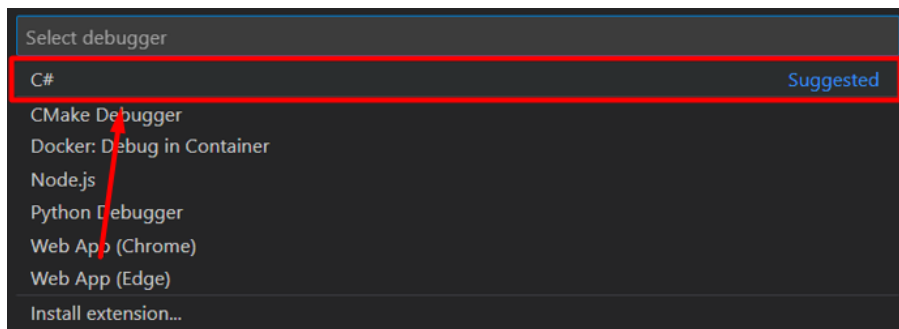


Рис.23 — Выбор компилятора

- д. После выбора компилятора выберите способ запуска. Выберите https.

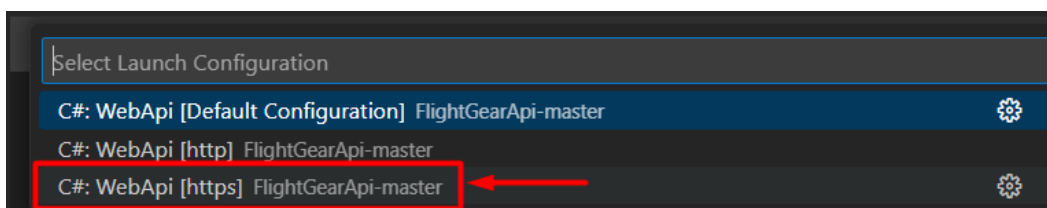


Рис.24 — Выбор способа запуска

- е. После загрузки, в терминале вашего редактора кода появятся две ссылки, как показано на *рисунке 25*. Выберите первую и тогда запустится сервер, а в браузере откроется пустая страница. Оставьте вкладку с ней. Она понадобится для запуска симуляции.

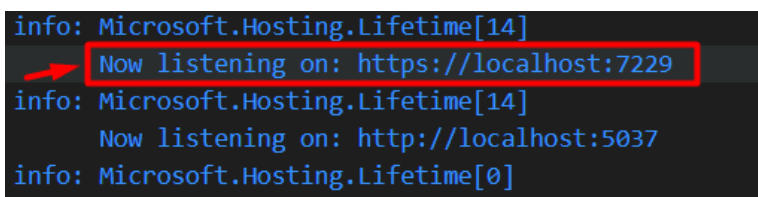


Рис.25 — Запуск серверной части

ВАЖНО: При последующих запусках, не будет необходимости проделывать все шаги с нуля. Достаточно будет выполнить пункт 1.5.

2. Главная страница. Работа с сервисом и запуск симуляции.

Данный раздел посвящен главной странице ПКУП. На ней вы можете создавать/удалять/изменять планы полётов, по которым будет осуществляться симуляция.

2.1. Обзор страницы

На верхней панели главной страницы присутствуют следующие кнопки:

- Current Flight (рис. 26) — это меню выбора сессии, с которой вы будете работать. Вы можете как создать новую при помощи кнопки ‘Create New’, так и выбрать из уже имеющихся, используя кнопку ‘Choose current flight’.

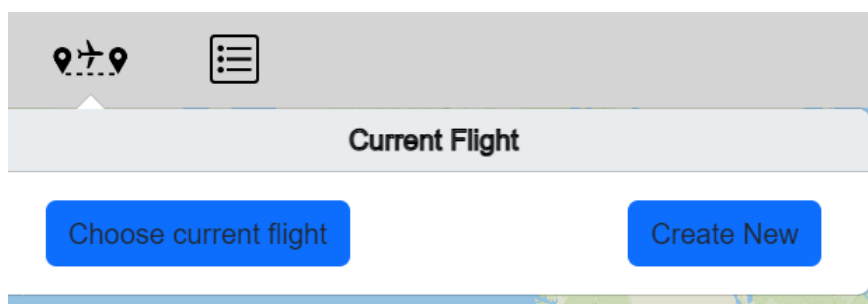


Рис.26 — Меню Current Flight

- All Flights (рис. 27) — это меню, в котором находятся все имеющиеся сессии полётов. В них описаны контрольные точки, по которым самолёт будет лететь в симуляции.

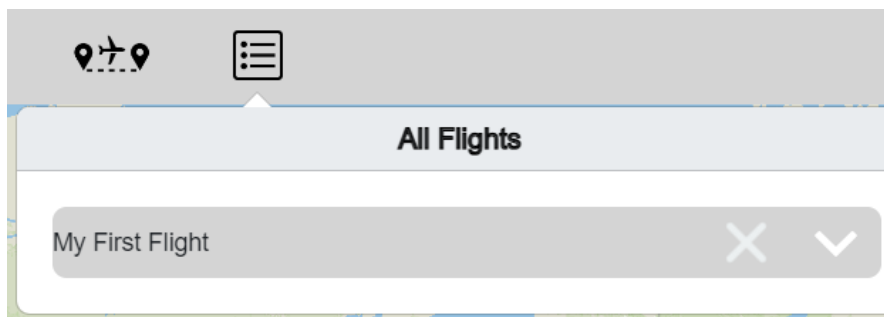


Рис.27 — Меню All Flights

- Planner (рис. 28) — кнопка перехода на главную страницу.

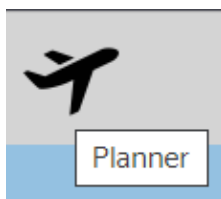


Рис.27 — Кнопка перехода на главную

- Docs (рис. 28) — ссылка на документацию ПКУП.

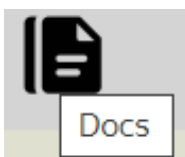


Рис.28 — ссылка на документацию

- Analytics (рис. 29) — кнопка перехода на [страницу анализа данных](#).

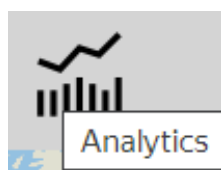


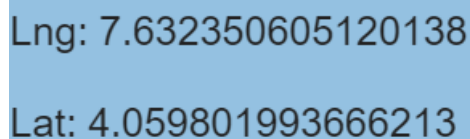
Рис.29 — Кнопка перехода на страницу анализа данных

В центре страницы находится карта мира. На неё вы можете добавлять контрольные точки, которые будут использованы в симуляции полёта. О том, как их создать будет рассказано в [пункте ниже](#).

С картой можно взаимодействовать следующими способами:

- Удерживая левую кнопку мыши, можно перемещаться по карте.
- Удерживая правую кнопку мыши, можно поворачивать карту.
- При нажатии правой кнопки мыши по карте, создается КТ.
- Используя колёсико мыши (или кнопки в правом нижнем углу), можно увеличить/уменьшить масштаб карты.

В левом нижнем углу страницы отмечены координаты (Lng — долгота, Lat — широта), значение которых определяется положением вашего курсора мыши на карте.



Lng: 7.632350605120138
Lat: 4.059801993666213

Рис.30 — Координаты на главной странице

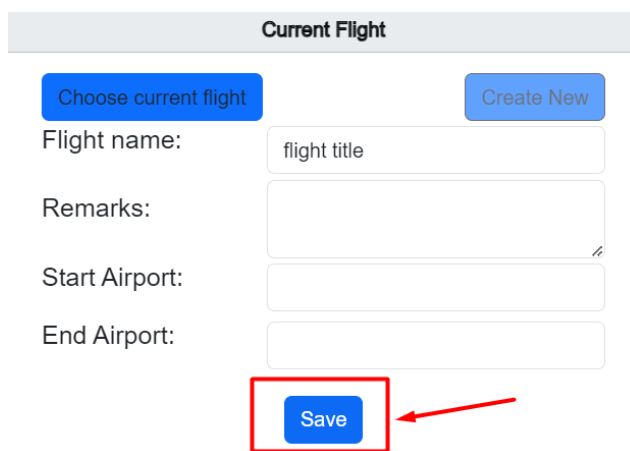
2.2. Добавление/удаление контрольных точек

а. Для запуска симуляции необходимо создать план полёта, по которому будет лететь самолёт. Чтобы это сделать, откройте меню Current Flight, и нажмите на кнопку ‘Create New’.

б. В открывшейся форме заполните следующие поля:

- Flight name — название плана полёта.
- Remarks — комментарии для плана полёта. Опциональное поле.
- Start Airport — аэропорт, с которого будет стартовать самолёт. Опциональное поле.
- End Airport — аэропорт, куда будет приземляться самолёт. Опциональное поле.

После заполнения формы, нажмите на кнопку ‘Save’ (см. рисунок 31).



The image shows a web form titled "Current Flight". At the top, there are two buttons: "Choose current flight" and "Create New". Below these are four input fields: "Flight name:" with a placeholder "flight title", "Remarks:" (a larger text area), "Start Airport:", and "End Airport:". At the bottom of the form, there is a blue "Save" button, which is highlighted by a red rectangular box. A red arrow points from the right towards the "Save" button.

Рис.31 — меню Current Flight

- с. Откройте меню All Flights и выберите план полёта, который только что создали с помощью кнопки √. Вы можете также удалить её с помощью кнопки ×.
- д. Создайте контрольные точки (минимум 2), по которым будет осуществляться симуляция. Для этого выберите место на карте и нажмите правую кнопку мыши. Данные о КТ появятся в меню (1), а на месте клика отобразится маркер (2) (см. рисунок 32).

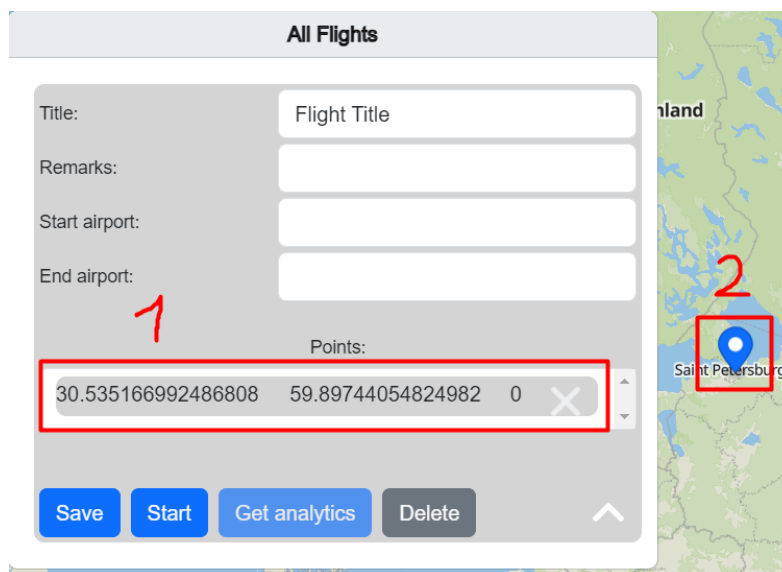
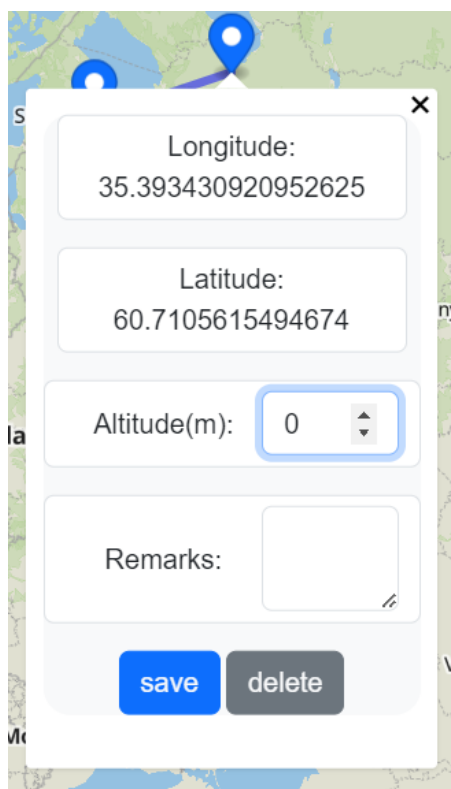


Рис.32 — данные о созданной КТ и её маркер

- е. Чтобы изменить высоту над уровнем моря, на которой должен будет находиться самолёт по достижению КТ, нажмите на её маркер. После клика откроется меню, показанное на рисунке 33.



Longitude:
35.393430920952625

Latitude:
60.7105615494674

Altitude(m): 0

Remarks:

save delete

Рис.33 — меню контрольной точки

- f. Внутри меню находятся значения долготы (Longitude) и широты (Latitude). Вы их можете изменить отсюда же, передвинув маркер КТ в желаемое место на карте. Также поменяйте значение высоты над уровнем моря (Altitude). При желании добавьте комментарий к КТ.
- g. После внесения изменений, нажмите на кнопку Save, или 'Delete', если желаете удалить КТ.
- h. После того, как вы добавили все необходимые вам КТ, вернитесь в меню вашей сессии в разделе All Flights и нажмите кнопку 'Save' для сохранения сессии.

2.3. Запуск симуляции полёта

- a. Для того чтобы запустить симуляцию, необходимо дублировать вкладку с сервисом. На второй вкладке в адресной строке, допишите к числу 7229 '/swagger, как на *рисунке 34*. Вас перенесет на страницу с возможными API запросами (см. *Рисунок 35*).

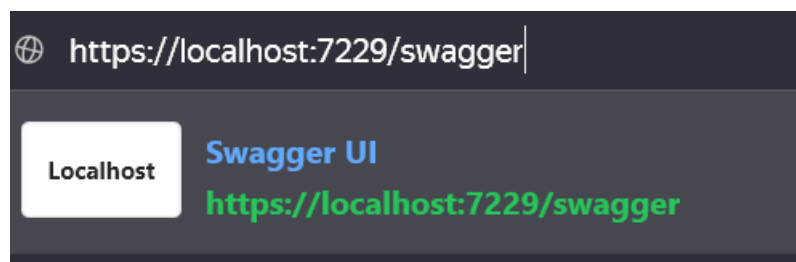


Рис.34- Ссылка переход на страницу с запросами

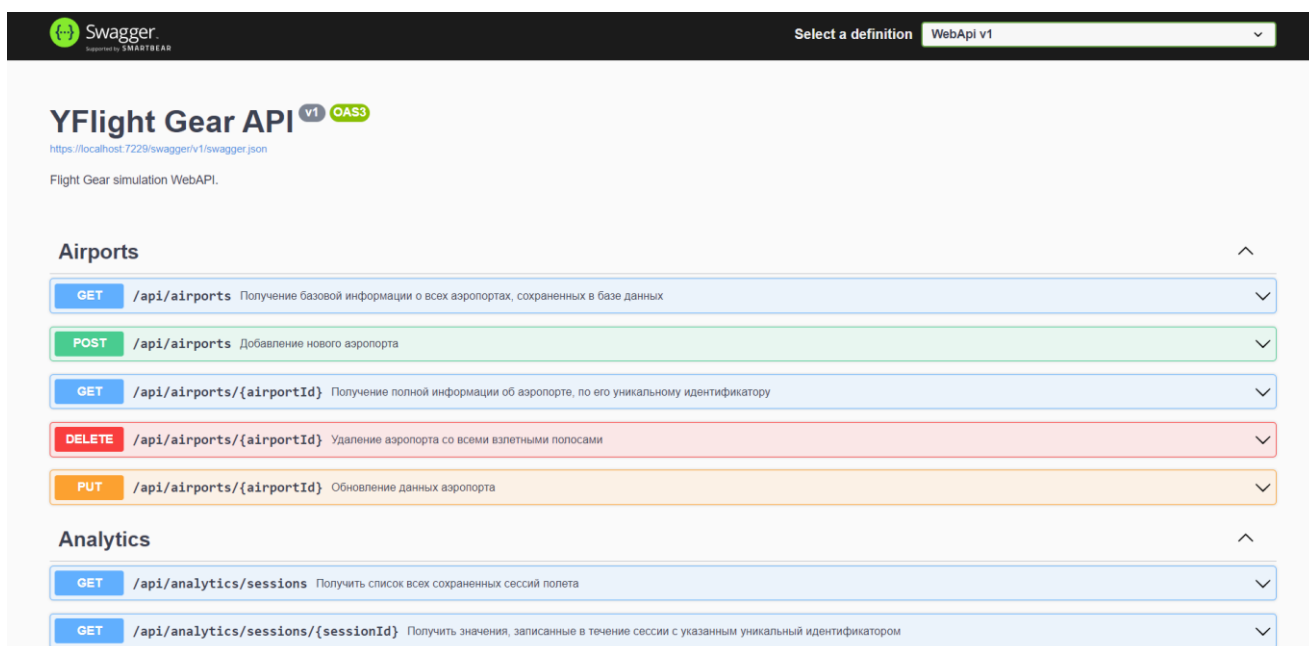
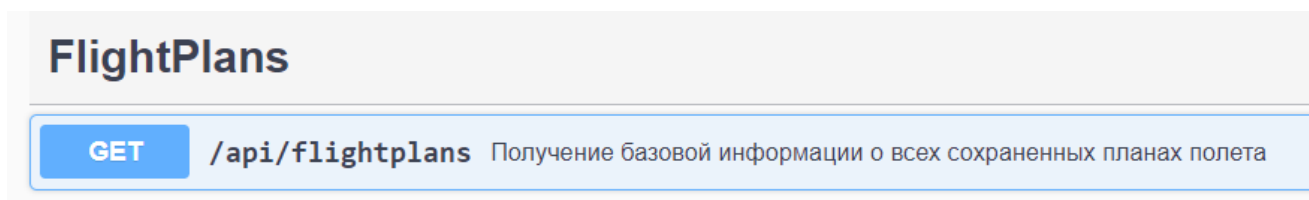


Рис.35 — На странице с запросами

- б. Чтобы запустить симуляцию по нужному плану полёта, нужно получить ID последнего. Для этого в разделе 'Flightplans' выберите запрос 'Получение базовой информации о всех сохраненных планах полета', нажав на него. Сам он представлен на рисунке ниже.



- с. Внутри запроса нажмите кнопку 'Try it out' в правом верхнем углу, а затем кнопку 'Execute' чуть ниже.

- d. После нажатия выведется список всех имеющихся планов полёта. Выберите тот, который хотите симулировать и скопируйте его ID из соответствующего поля (см. *рисунок 36*).

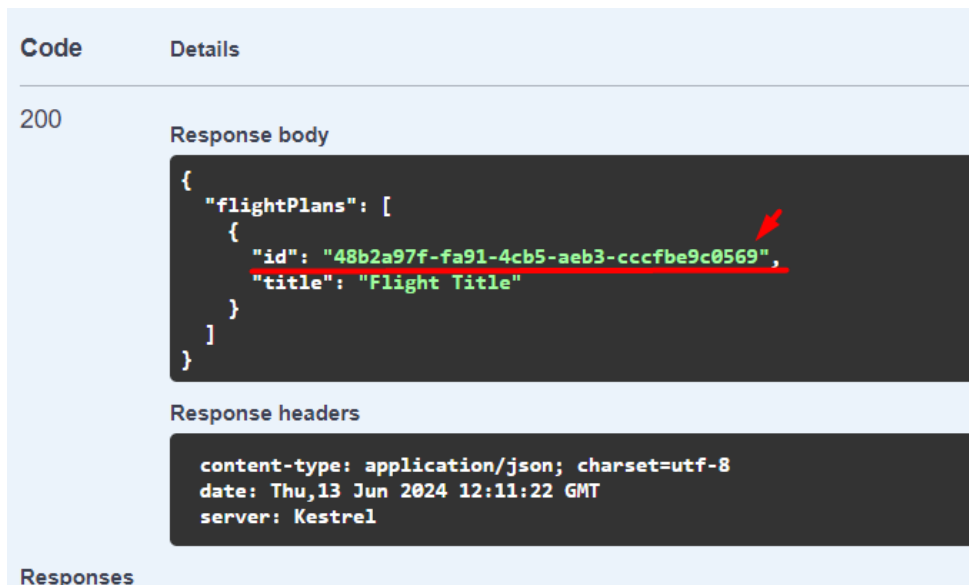
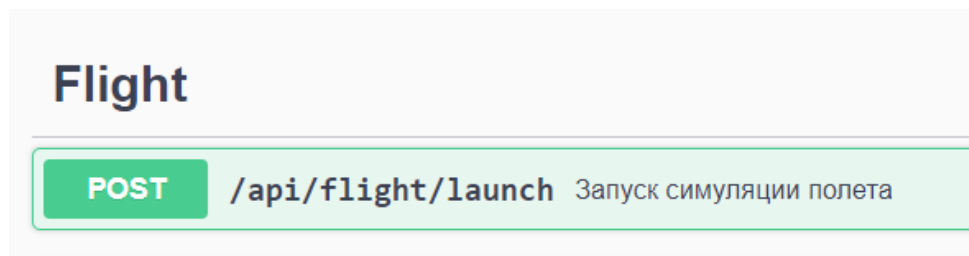


Рис.36 — Выбор плана полёта по его идентификатору

- e. В разделе ‘Flight’ выберите запрос ‘Запуск симуляции полёта’



- f. Внутри запроса нажмите кнопку Try it out в верхнем правом углу

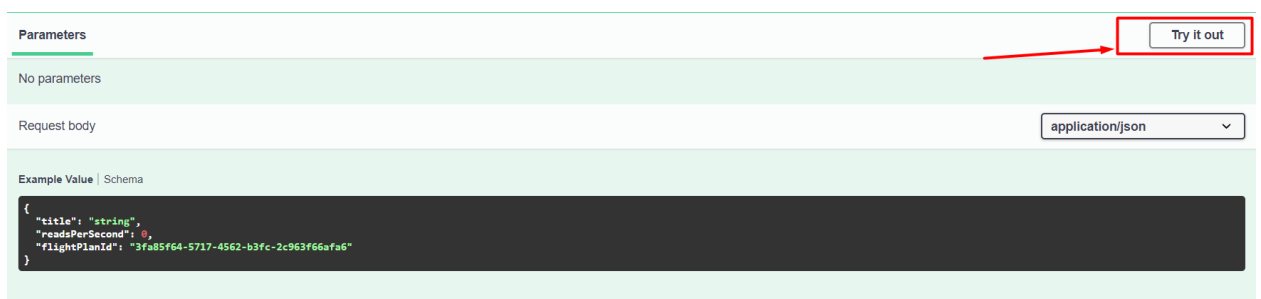


Рис.36 — Создание запроса

g. Внутри запроса, введите необходимые параметры:

- title — название симуляции.
- readsPerSecond — частота считывания параметров (раз в *ваше значение* секунд). **Должно быть больше 0.**
- flightPlanId — идентификатор плана полёта. В это поле вставьте тот ID, который вы скопировали в шаге d.

```
{  
  "title": "string",  
  "readsPerSecond": 0,  
  "flightPlanId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"  
}
```

Рис.37 — Параметры запроса

h. После задавания параметров нажмите кнопку 'Execute'.

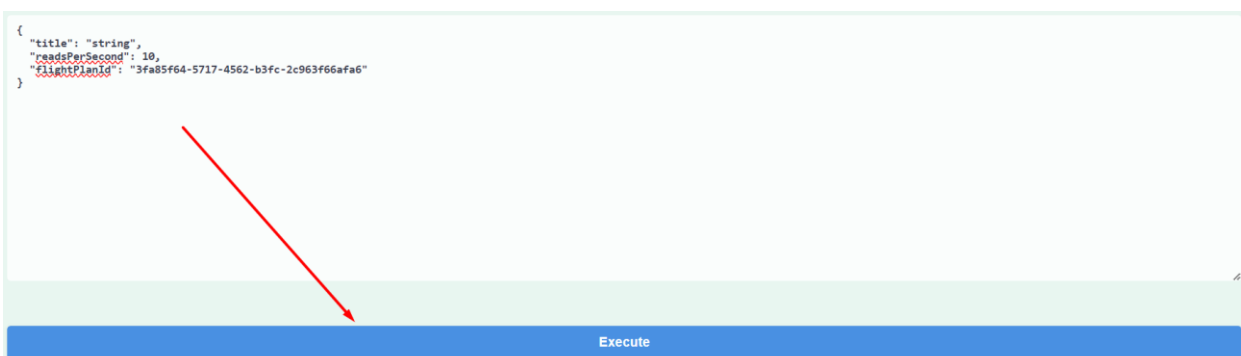


Рис.38 — Выполнение запроса

i. После запуска откроется FlightGear, в котором, в режиме реального времени, осуществится полет в режиме 'Автопилот' по заданным вами КТ.



Рис.39 — Экран FlightGear. Запуск симуляції

- і. После завершения симуляції, её результаты будут проанализированы и сохраняться в базе данных, к которой вы можете легко получить доступ перейдя на страницу 'К анализу'.

3. Настройка сервиса и симуляції

В данном разделе описываются дополнительные настройки для симуляций и сервиса. В частности установка нового самолёта и иной территории.

3.1. Установка самолёта СУ-57(Т-50)

- а. Скачайте архив [Т-50](#).
- б. Скопируйте из архива папку "Т-50" в папку по пути "FlightGear 2020.3\data\Aircraft".
- в. Запустите FlightGear, во вкладке "ЛА". В ней появился новый самолет Sukhoi T-50 PAK FA (рис.40).

ВАЖНО: у Sukhoi T-50 PAK FA есть две версии: JSBSim и YASim, — мы будем работать с версией YASim. Выберите её во FlightGear, нажав на название самолёта, в выпадающем списке (рис.40).

- d. Для проверки, что все установлено верно, отправьте ваш самолёт на аэродром (*рис.41*).

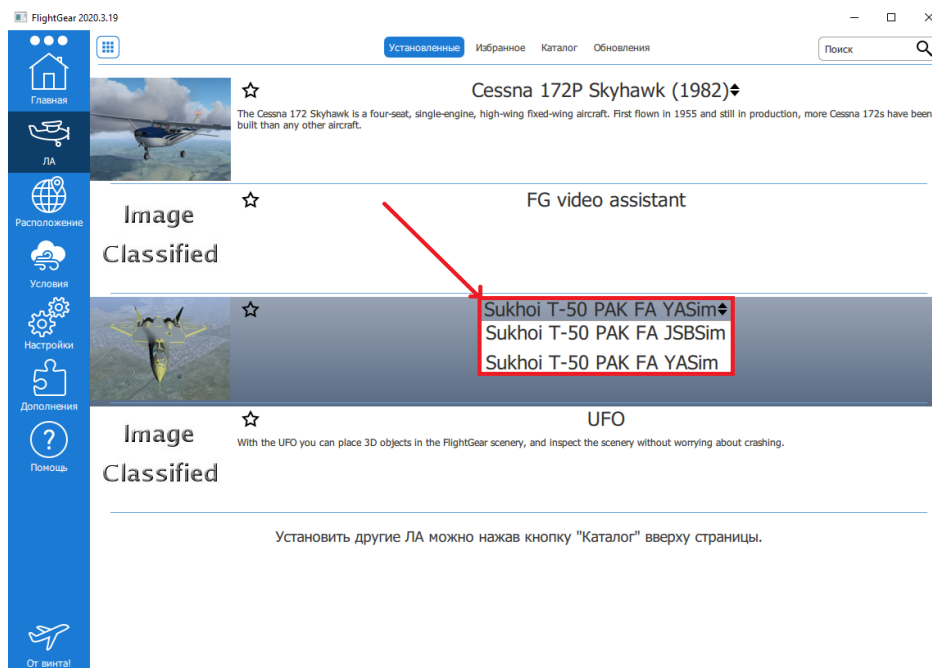


Рис.40 — Sukhoi T-50 PAK FA во вкладке "ЛА" в FlightGear



Рис.41 — Sukhoi T-50 PAK FA YASim на аэродроме

3.2. Установка местности России

На данном этапе к скачиванию доступны только некоторые города: Гудаута, Абхазия, Санкт-Петербург, Геленджик, Ростов-на-Дону, Сочи, Адлер.

- а. Чтобы скачать архив, перейдите по ссылке [Project-Russia-master](#), далее нажмите на зеленую кнопку "Code", в открывшемся окошке нажмите на "Download ZIP" (рис. 42) и дождитесь скачивания архива.

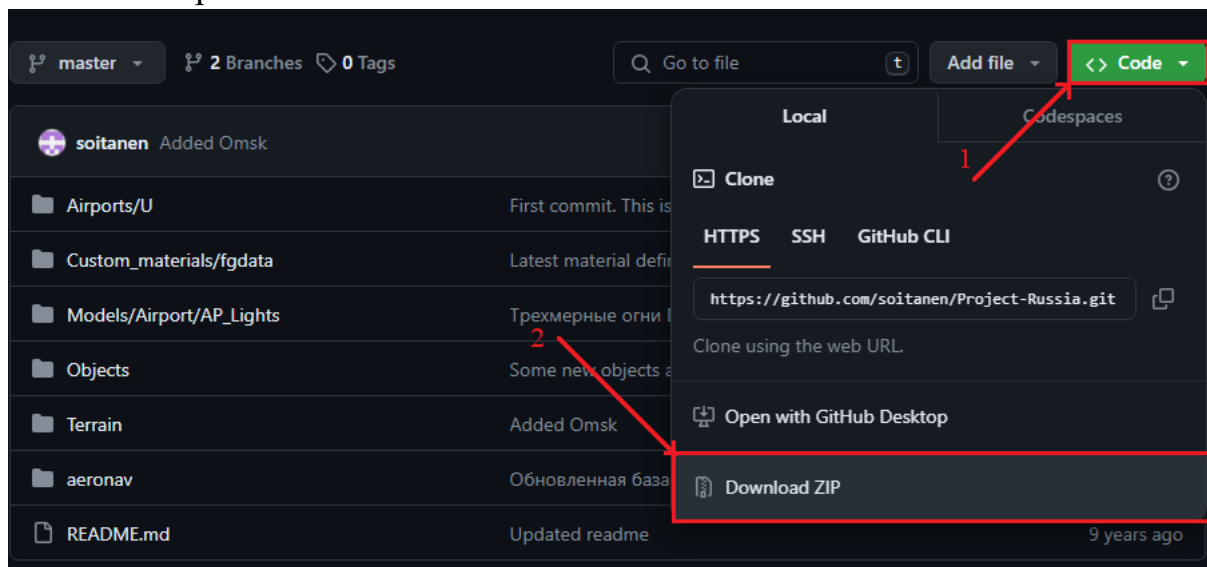


Рис.42 — Скачивание архива из GitHub

- б. Откройте скачанный файл, перейдите в папку "fgdata", которая находится по пути "Project-Russia-master/Custom_materials/fgdata". Из "fgdata" скопируйте две папки "Materials" и "Textures" в папку "data" по пути установки FlightGear.
- в. Для лучшей навигации перейдите по пути "Project-Russia-master/aeronav/fgdata" и оттуда также скопируйте две папки "Airports" и "Nav aids" в папку "data" по пути установки FlightGear. При одинаковых файлах выберите "заменить".
- г. Скопируйте папки "Airports", "Models", "Objects" и "Terrain" из папки "Project-Russia-master" в папку FlightGear по пути "...\\data\\Scenery". При одинаковых файлах выберите "заменить".
- д. Для проверки, что все установлено верно, рассмотрим город Сочи. Запускаем FlightGear, во вкладке "Расположение" в поиске пишем Sochi, выбираем URSS – Sochi (рис. 43) и запускаем.

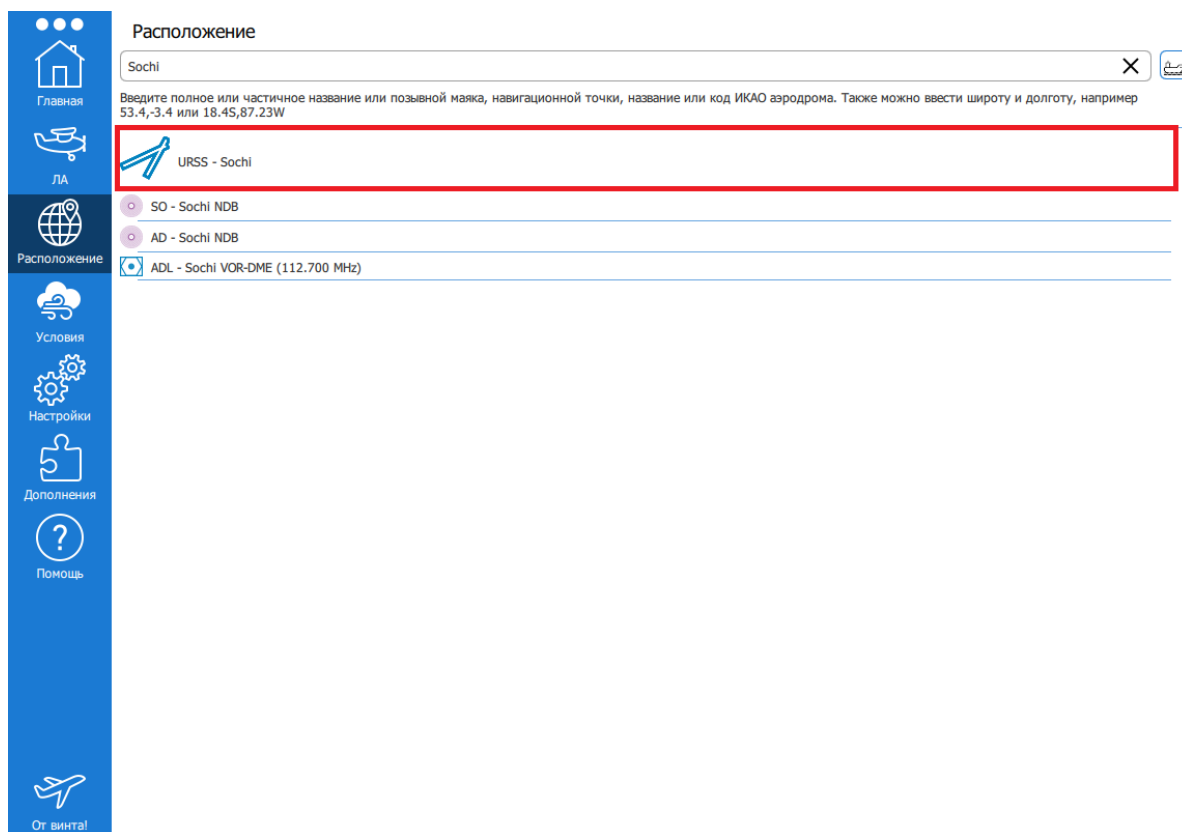


Рис.43 — аэродром Сочи во вкладке "Расположение"



Рис.44 — Самолёт Sukhoi T-50 PAK FA YASim над Сочи

4. Страница анализа данных и работа с ней

На данной странице вы можете получить доступ к результатам симуляции и визуализировать полученные характеристики и показатели на графике.

4.1. Переход на страницу анализа данных

Чтобы перейти на страницу анализа данных, вернитесь на главную страницу сервиса и в правом нижнем углу нажмите кнопку ‘К анализу’.

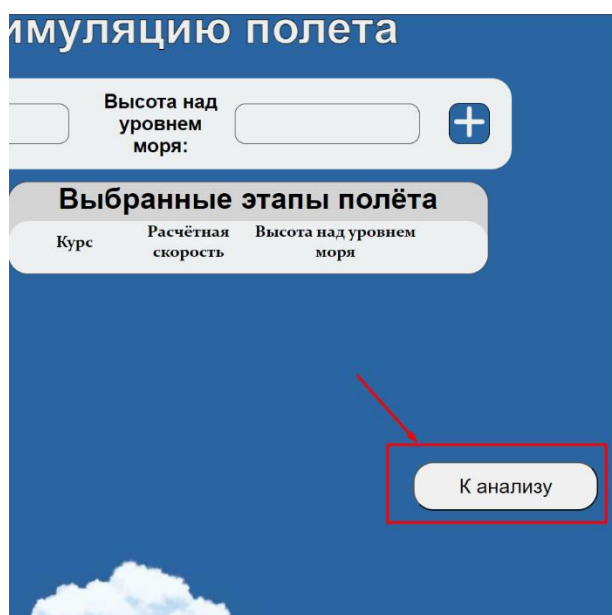


Рис.45 — Кнопка для перехода на страницу анализа

- а. Выберите из списка сессий (см. рисунок 46) ту, результаты который вы хотите изучить.

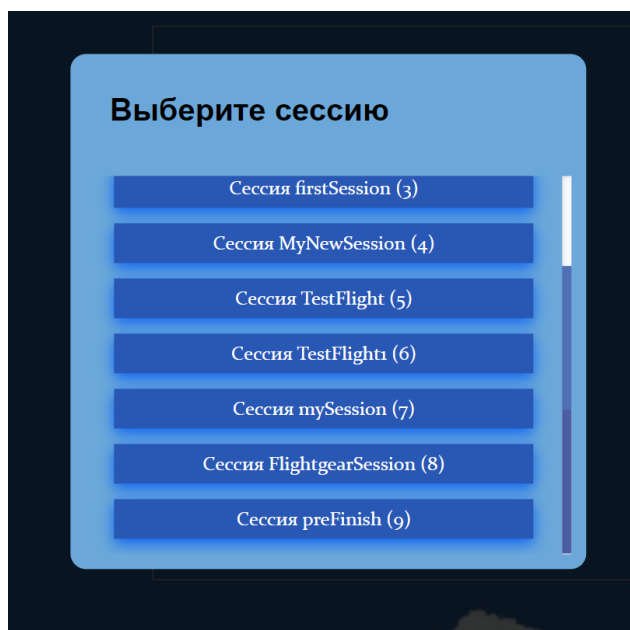


Рис.46 — Меню выбора сессии

- б. После выбора желаемой сессии вы попадаете на страницу с анализом данных. Чтобы вернуться обратно, нажмите на кнопку ‘К планированию’ в левом нижнем углу страницы.

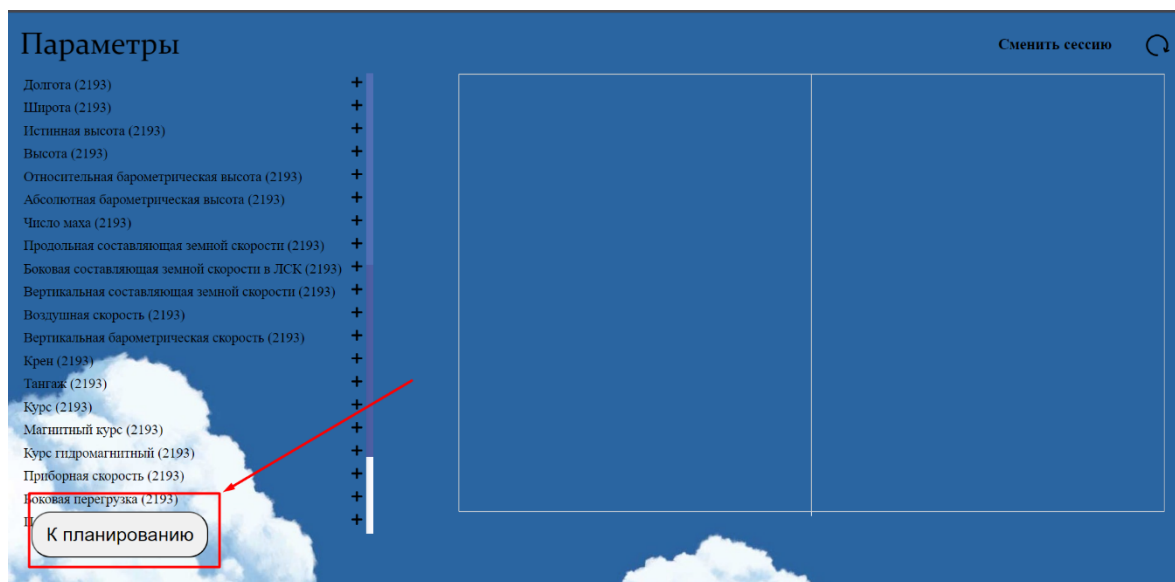


Рис.47 — Страница с анализом и с кнопкой перехода на главную

- с. Чтобы сменить сессию на другую в правом верхнем углу страницы нажмите кнопку ‘Сменить сессию’ (см. Рис. 43).

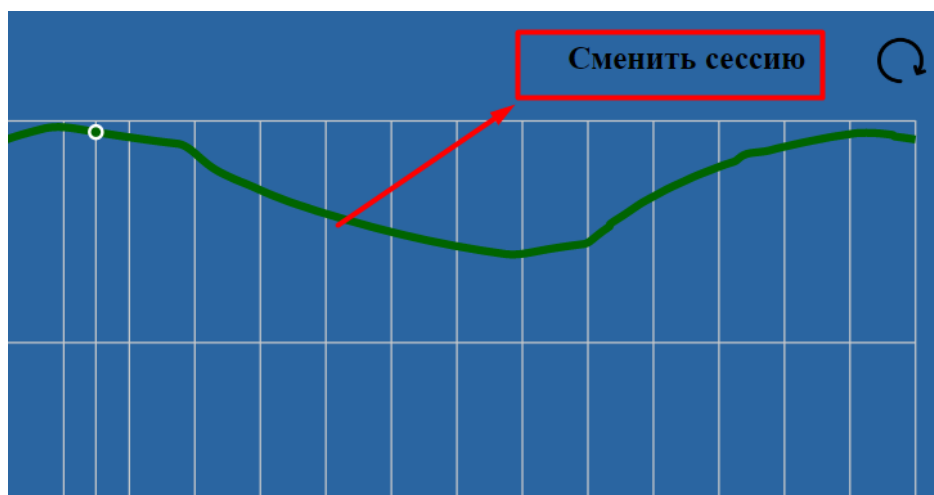


Рис.48— Кнопка 'Сменить сессию'

4.2. Работа с графиками данных

- а. На панели справа представлены показатели, зафиксированные в процессе полёта.
- б. Выберите желаемый показатель, чтобы отобразить его на графике слева.

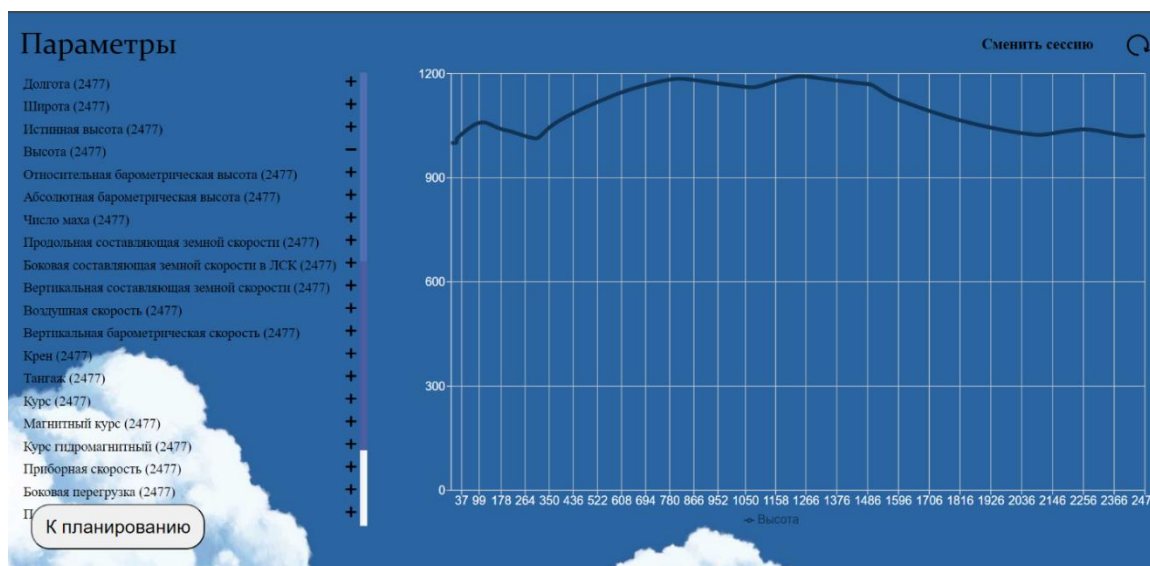


Рис.49 — График с показателем 'Высота'

- с. На графике вы можете отобразить сколько угодно показателей одновременно. Для этого вы также можете выбрать их на панели параметров.

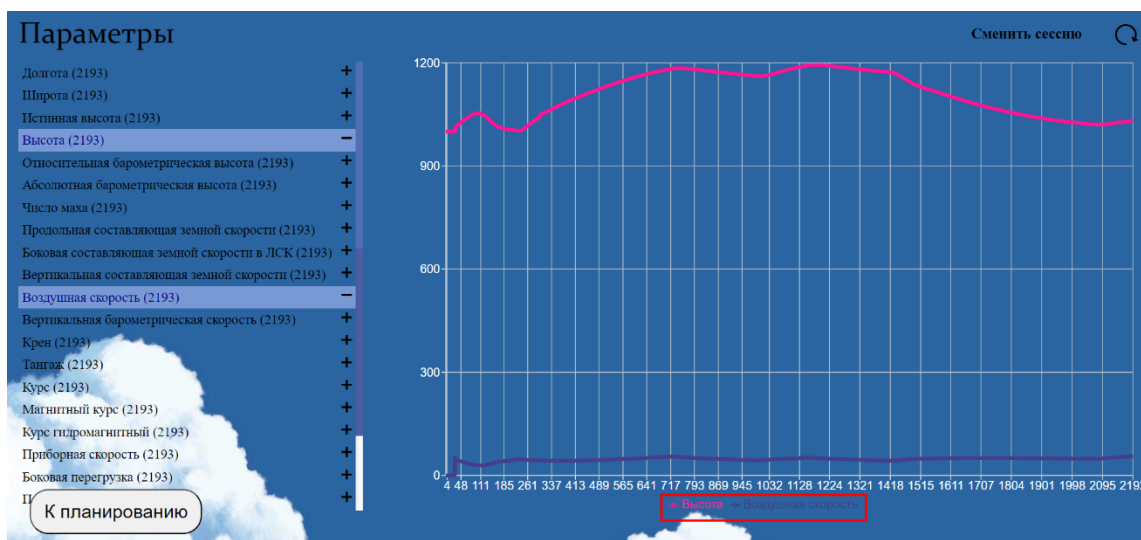



Рис. 50 — График с 2 показателями (Высота и воздушная скорость)

- d. Если цвета линий не устраивают, вы можете сменить их, нажав на кнопку  в правом верхнем углу экрана.

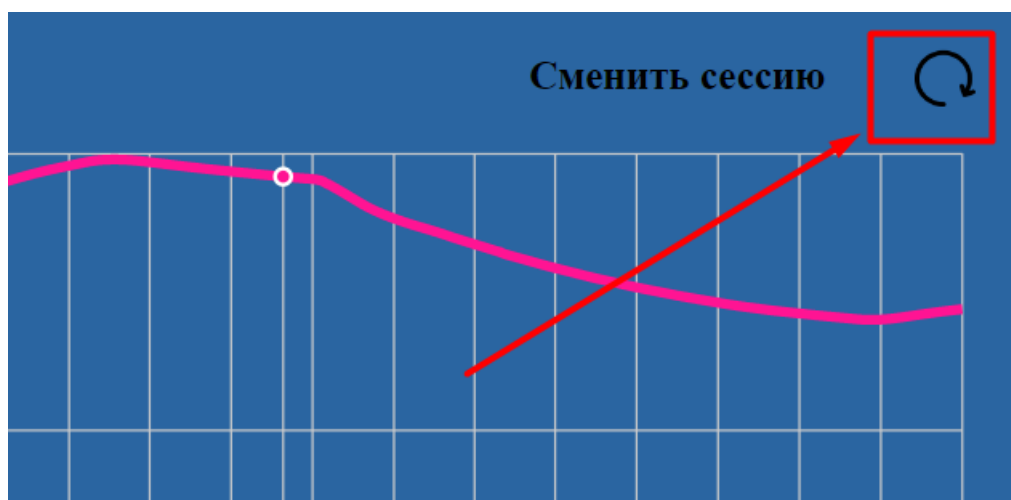


Рис.51 — Кнопка смены цвета линий на графике

5. The FlightGear User Interface

Ещё одним из способов получения информации о текущей симуляции о симуляции является The FlightGear User Interface (Phi). Это мультиплатформенный пользовательский интерфейс на основе HTML5/CSS3/JS. Помимо получения информации о воздушном судне, Phi позволяет изменять симуляцию в режиме реального времени, например переместить самолёт в совершенно другое место или сменить текущее время.

5.1. Подключение к The FlightGear User Interface

Чтобы подключиться к Phi, перейдите на страницу <http://localhost:5400> во время активной работы симуляции полёта. После перехода на неё перед вами откроется интерфейс, показанный на *рисунке 47*.

ВАЖНО: В рамках работы с данным сервисом он будет активен лишь во время симуляции полёта. Как только она завершится, связь с Phi прервется.

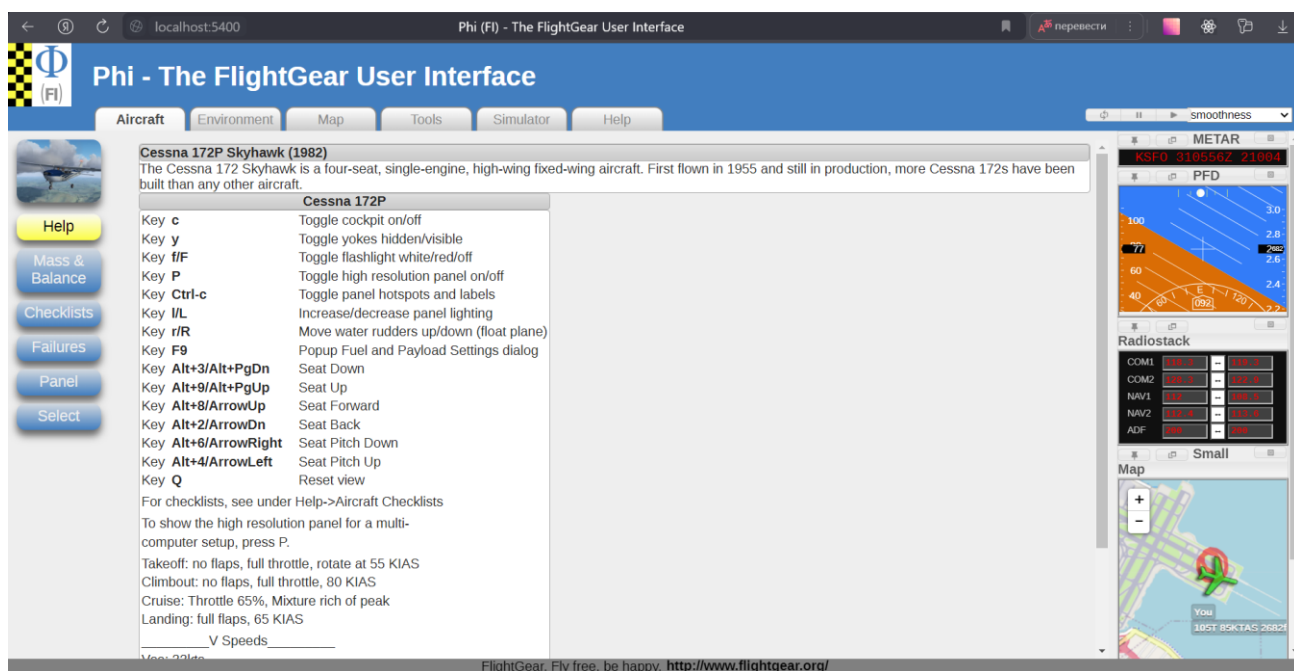


Рис.52 — Интерфейс Phi

5.2. Описание основных разделов

Phi имеет двустороннюю привязку данных к FlightGear через внутренний веб-сервер HTTP. Это позволяет ему стать полнофункциональным пользовательским интерфейсом, который может напрямую управлять FlightGear.

Далее представлен список представляет собой набор имеющихся и рабочих функций и разделов:

5.2.1. Aircraft (Воздушное судно)

Здесь находится вся информация, связанная с текущим летательным аппаратом:

- Help — описание текущего самолёта и советы по управлению.
- Mass & Balance — отображает параметры топлива и загрузки самолета.
- Checklists — отображает текущие показатели различных приборов внутри самолёта.

5.2.2. Environment (Окружающая среда)

Здесь находятся настройки для параметров окружающей среды:

- Date & Time — позволяет определять время суток в симуляции.
- Weather — отображает текущую погоду.
- Position — интерфейс, позволяющий выполнять поиск аэропортов по [кодам ICAO](#), а затем выбирать взлётно-посадочную полосу / место парковки для загрузки в FlightGear.

5.2.3. Map (Карта)

Отображает на карте положение летательного аппарата в режиме реального времени и позволяет изменять его, путем перетаскивания значка самолета в произвольное место.

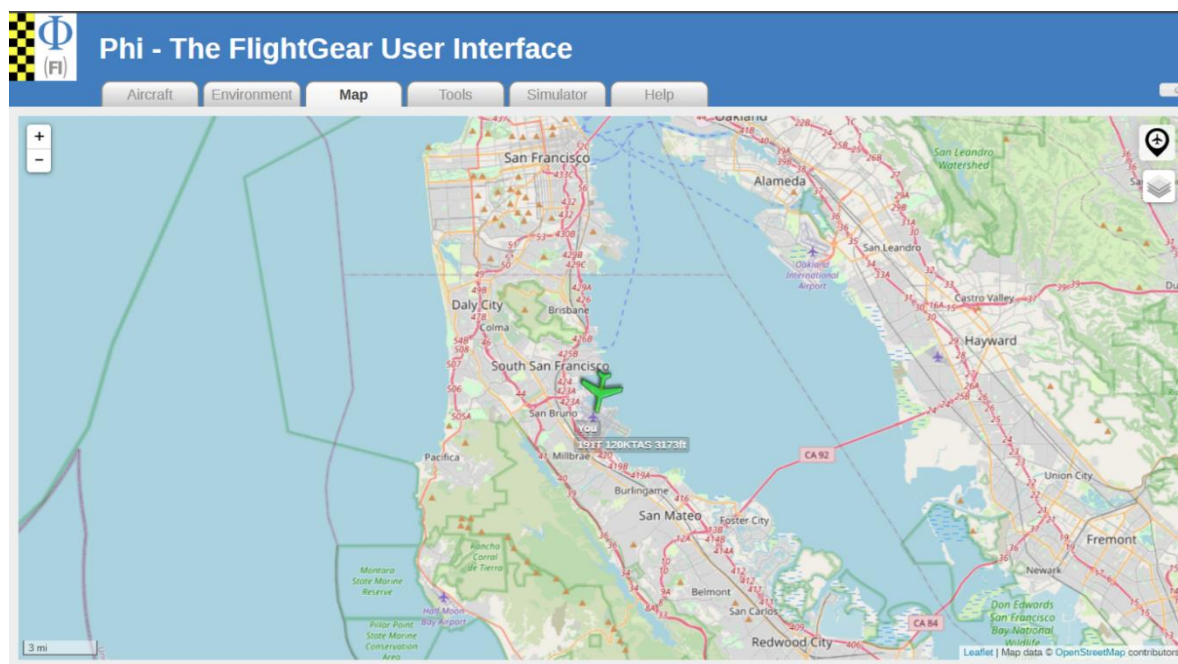


Рис.53 — Карта в Phi

5.2.4. Tools (Инструменты)

Здесь находятся инструменты для подсчёта времени и различных показателей:

- Holding pattern — инструмент, помогающий понять, как ввести зону ожидания. Введите маршрут подхода (Inbound track) и курс (Heading). После этого схема наглядно покажет зону ожидания.
- Stopwatch — позволяет использовать секундомер. При желании можно добавить несколько.

5.2.5. Simulator (Симулятор)

Здесь находятся дополнительные способы для взаимодействия с симуляцией:

- Screenshot — делает снимок экрана FlightGear раз в n секунд (n- установленное пользователем время).
- Property tree — это интерфейс для низкоуровневых переменных состояния во время выполнения через древовидную иерархию. Она позволяет контролировать поведение симулятора и управлять им во время выполнения. Однако, не все свойства, которые являются можно изменить.

Например, нельзя установить положения элеронов и руля высоты.

- Config — позволяет включить ИИ и воспользоваться многопользовательской игрой. Эти функции не относятся к данному сервису.
- Reset — перезапуск симуляции.
- Exit — досрочное завершение симуляции.

6. Работа с API. Функции и запросы

Данный раздел посвящён работе с API сервиса. В нём описываются доступные функции системы (API запросы), их предназначение, входные данные и возвращаемые значения.

Чтобы перейти к списку доступных функций, необходимо дублировать вкладку с серверной частью. На ней в адресной строке, допишите к числу 7229 '/swagger', как на *рисунке 54*. Вас перенесет на страницу с возможными API запросами (см. *Рисунок 55*).

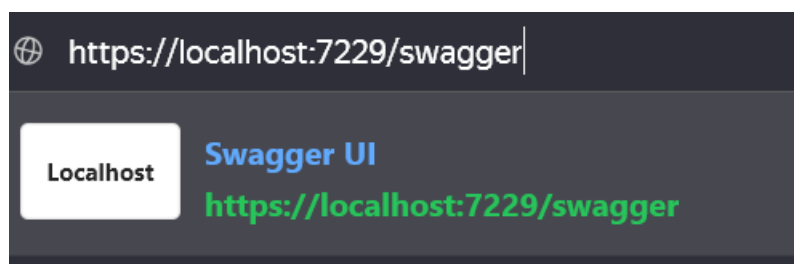


Рис.54- Ссылка переход на страницу с запросами

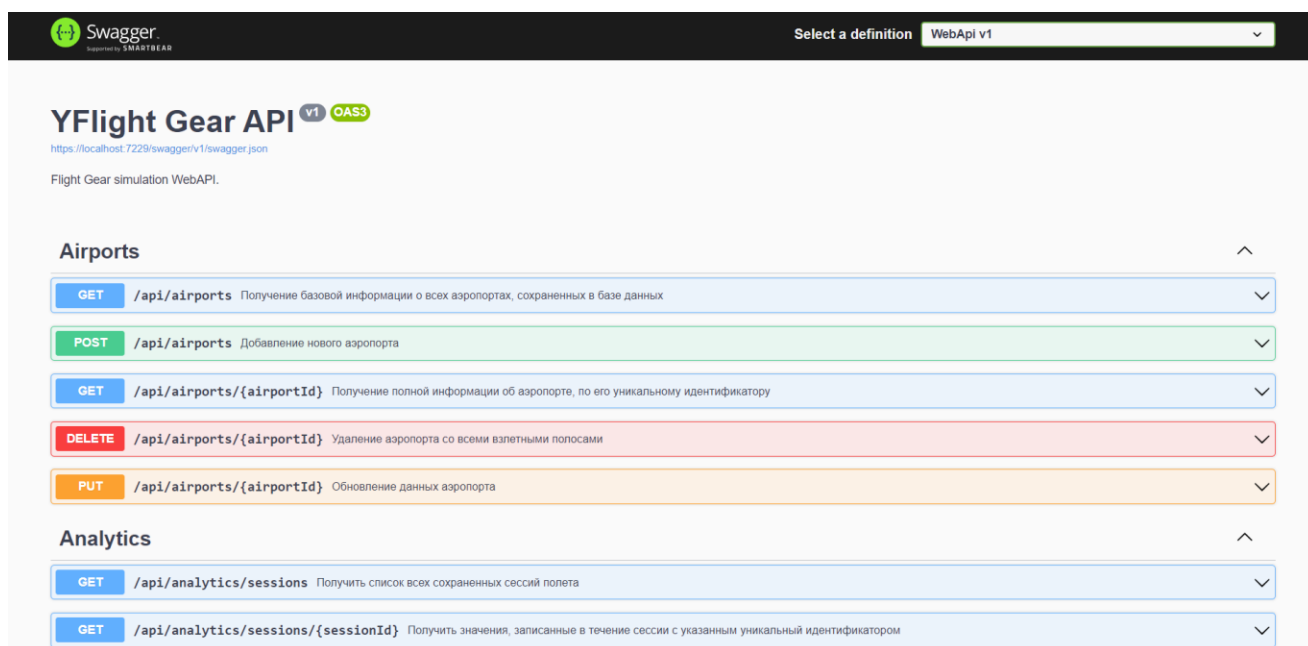


Рис.55- На странице с API

6.1. План полёта

В данной категории находятся запросы, позволяющие работать с планами полёта, а также получать информацию об уже имеющихся. Категория расположена в разделе FlightPlans.

6.1.1. Получение информации о планах полета

- a. Метод запроса: GET
- b. Действие: Выводит базовую информацию о планах полета, сохранённых в базе данных
- c. URL: /api/flightplans
- d. Входные данные: Отсутствуют
- e. Ответ:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1.1	FlightPlans	Array of objects	Да	Массив со всеми планами полёта, которые сохранены в базе данных	[]
1.2	FlightPlans.Id	guid	Да	Идентификатор плана полёта	9d974548-7e90-4885-973a-dcdcb92c44f4
1.3	FlightPlans.Title	string	Да	Название плана полёта	test-plan1

6.1.2. Получение информации о плане полета по ID

- a. Метод запроса: GET
- b. Действие: Выводит детальную информацию о плане полета по указанному идентификатору
- c. URL: /api/flightplans/{id:guid}
- d. Route-параметры:

- id — уникальный идентификатор плана полета
- e. Входные данные: отсутствуют
- f. Ответ: object [[FlightPlanResponse](#)]

6.1.3. Удаление плана полета по ID

- a. Метод запроса: DELETE
- b. Действие: Удаляет план полета по указанному идентификатору
- c. URL: /api/flightplans/{id:guid}
- d. Route-параметры:
 - id — идентификатор плана полета, который необходимо удалить. Тип данных — guid.
- e. Входные данные: отсутствуют
- f. Ответ:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Status	string	Да	Статус выполнения запроса	1) Success — план полёта был создан и сохранен 2) Failed — во время выполнения произошла ошибка. Подробности в поле Comment.
2	Comment	string	Нет	Комментарий от сервера, полученный во время выполнения запроса	Указанный план полета не существует.

6.1.4. Создание нового плана полета

- a. Метод запроса: POST
- b. Действие: Создаёт новый план полёта на основе параметров, указанных пользователем
- c. URL: /api/flightplans
- d. Входные данные:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Title	string	Да	Название плана полета	test-plan1
2	Remarks	string	Нет	Пользовательские пометки для плана полета	Прямой перелет из Питера в Москву
3	DepartureRunwayId	guid	Нет	Идентификатор взлётной полосы аэропорта	fc46746c-513e-4124-8e58-30fde2647ad7
4	ArrivalRunwayId	guid	Нет	Идентификатор посадочной полосы аэропорта	5a4e92f1-9af1-40dd-919f-815b729531cc

e. Ответ:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Status	string	Да	Статус выполнения запроса	1) Success — план полета был создан и сохранен 2) Failed — во время выполнения произошла ошибка. Подробност

					и в поле Comment.
2	Comment	string	Нет	Комментарий от сервера, полученный во время выполнения запроса.	Указанная взлётная полоса не существует.
3	FlightPlan	object [FlightPlanResponse]	Нет	Обновленный план полета.	{ }

6.1.5. Обновление плана полета

- a. Тип запроса: PUT
- b. Действие: Обновляет выбранный пользователем план полета по идентификатору
- c. URL: /api/flightplans/{id:guid}
- d. Route-параметры:
 - id — идентификатор плана полета, который необходимо обновить, тип данных — guid.
- e. Входные данные:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Title	string	Да	Название плана полета	test-plan1
2	Remarks	string	Нет	Пользовательские пометки для плана полета	Прямой полёт из Питера в Москву
3	Departure RunwayId	guid	Нет	Идентификатор взлётной полосы аэропорта. Может быть null	fc46746c-513e-4124-8e58-30fde2647ad7

4	ArrivalRunwayId	guid	Нет	Идентификатор посадочной полосы аэропорта. Может быть null	5a4e92f1-9af1-40dd-919f-815b729531cc
---	-----------------	------	-----	--	--------------------------------------

f. Ответ:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Status	string	Да	Статус выполнения запроса	1) Success — план полета был создан и сохранен 2) Failed — во время выполнения произошла ошибка. Подробности в поле Comment.
2	Comment	string	Нет	Комментарий от сервера, полученный во время выполнения запроса.	Указанная взлётная полоса не существует.
3	FlightPlan	object [FlightPlanResponse]	Нет	Обновленный план полета.	{ }

6.2. Точки маршрута

В данной категории находятся запросы, позволяющие работать с точками маршрута у определённого плана полета, а также получать

інформацію об уже існуючих. Категорія розташована в розділі FlightPlans.

6.2.1. Получение точек маршрута

- a. Тип запроса: GET
- b. Действие: Выводит список точек маршрута у конкретного плана полета.
- c. URL: /api/flightplans/{flightPlanId:guid}/points
- d. Route-параметры:
 - flightPlanId — идентификатор плана полета, точки маршрута которого нужно получить.
- e. Входные данные: Отсутствуют.
- f. Ответ:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	RoutePoints	Array of objects [RoutePoint Response]	Да	Массив точек маршрута плана полёта	[]

6.2.2. Добавление точки маршрута

- a. Метод запроса: POST
- b. Действие: Добавляет новую точку маршрута к существующему плану полета
- c. URL: /api/flightplans/{flightPlanId:guid}/points
- d. Route-параметры:
 - flightPlanId — идентификатор плана полета, к которому нужно добавить точку.
- e. Входные данные:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
---	----------	------------	----------------	----------	-------------------

1	Order	number	Да	Порядковый номер, на который нужно вставить новую точку, отсчет с нуля	0
2	Longitude	float	Да	Географ. долгота точки. Отрицательные значения — южная долгота	59.885082
3	Latitude	float	Да	Географ. широта точки. Отрицательные значения - западная широта	30.302281
4	Altitude	float	Да	Высота над уровнем моря в метрах	900
5	Remarks	string	Нет	Дополнительные пользовательские пометки для точки	На этой точке чуть-чуть опускаемся

f. Ответ:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Status	string	Да	Статус выполнения запроса	1) Success — новая точка была добавлена 2) Failed — во время выполнения произошла ошибка. Подробности в поле Comment.
2	Comment	string	Нет	Комментарий от сервера, полученный	Указанный план полета не существует

				во время выполнения запроса	
3	FlightPlan	object [FlightPlanResponse]	Нет	Обновленный план полёта	{ }

6.2.3. Удаление точки маршрута

- a. Метод запроса: DELETE
- b. Действие: Удаление точки маршрута у существующего плана полета
- c. URL:
/api/flightplans/{flightPlanId:guid}/points/{pointOrder:number}
- d. Route-параметры:
 - flightPlanId — идентификатор плана полета, у которого нужно удалить точку.
 - pointOrder — порядковый номер точки маршрута, которую нужно удалить. Отсчет идет с нуля.
- e. Входные параметры: Отсутствуют
- f. Ответ:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Status	string	Да	Статус выполнения запроса	1) Success — точка была удалена 2) Failed — во время выполнения произошла ошибка. Подробности в поле Comment.
2	Comment	string	Нет	Комментарий от сервера, полученный	Указанный план полета не существует

				во время выполнения запроса.	
3	FlightPlan	object [FlightPlanResponse]	Нет	Обновленный план полета.	{ }

6.2.4. Обновление точки маршрута

- a. Метод запроса: PUT
- b. Действие: Обновление точки маршрута у существующего плана полета
- c. URL:
/api/flightplans/{flightPlanId:guid}/points/{pointOrder:number}
- d. Route-параметры:
 - flightPlanId — идентификатор плана полета, у которого нужно обновить точку маршрута.
 - pointOrder — порядковый номер точки маршрута, которую нужно обновить. Отсчет идет с нуля.
- e. Входные данные:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Longitude	float	Да	Географ. долгота точки. Отрицательные значения — южная долгота.	59.885082
2	Latitude	float	Да	Географ. широта точки. Отрицательные значения — западная широта.	30.302281
3	Altitude	float	Да	Высота над уровнем моря в метрах	900

4	Remarks	string	Нет	Дополнительные пользовательские пометки для точки	На этой точке чуть-чуть опускаемся
---	---------	--------	-----	---	------------------------------------

f. Ответ:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Status	string	Да	Статус выполнения запроса	1) Success — точка была обновлена 2) Failed — во время выполнения произошла ошибка. Подробности в поле Comment.
2	Comment	string	Нет	Комментарий от сервера, полученный во время выполнения запроса	Точки с указанным порядком номером не существует в данном плане полета.
3	FlightPlan	object [FlightPlanResponse]	Нет	Обновленный план полета	{ }

6.3. Аэропорты/полосы

В данной категории находятся запросы, позволяющие получать информацию об аэропортах/взлетно-посадочных полосах. Категория находится в разделе Airports.

6.3.1. Получение информации об аэропортах

- a. Метод запроса: GET
- b. Действие: Выводит всю базовую информацию о всех аэропортах, сохраненных в базе данных.
- c. URL: /api/airports
- d. Входные данные: Отсутствуют
- e. Ответ:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Airports	Array of objects	Да	Массив сохраненных аэропортов	[]
1.1	Id	guid	Да	Уникальный идентификатор аэропорта	5a4e92f1-9af1-40dd-919f-815b729531cc
1.2	Title	string	Да	Название аэропорта	Пулково
1.3	Code	string	Да	Код аэропорта	ULLI
1.4	City	string	Да	Название города	Санкт-Петербург

6.3.2. Получение информации об аэропортах по ID

- a. Метод запроса: GET
- b. Действие: Выводит полную информацию об аэропорте, по его уникальному идентификатору.
- c. URL: /api/airports/{id:guid}
- d. Route-параметры:
 - id — уникальный идентификатор аэропорта
- e. Входные данные: Отсутствуют
- f. Ответ: object [[AirportResponse](#)]

6.4. Запуск полёта

В данной категории находятся запросы, позволяющие взаимодействовать непосредственно с симуляцией. Категория находится в разделе Flight

6.4.1. Запуск симуляции полёта

- a. Метод запроса: POST
- b. Действие: [Запускает симуляцию полета](#)
- c. URL: /api/flight/launch
- d. Входные данные:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Title	string	Да	Название сессии, которое будет использоваться на странице с аналитикой для отображения	testFlight1
2	ReadsPerSecond	number	Да	Кол-во считываний параметров полета в секунду (частота). Должна быть не меньше 1	10
3	FlightPlanId	guid	Да	Уникальный идентификатор плана полёта, который будет использован	5a4e92f1-9af1-40dd-919f-815b729531cс

е. Ответ:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Status	string	Да	Статус выполнения запроса	1) Success — симуляция

					была запущена 2) Failed — во время выполнения произошла ошибка. Подробности в поле Comment
2	Comment	string	Нет	Комментарий от сервера, полученный во время выполнения запроса	Плана полета с указанным идентификатором не существует

6.4.2. Получение статуса полёта/симуляции

- a. Метод запроса: GET
- b. Действие: Выводит текущий статус симуляции и полёта.
- c. URL: /api/flight/status
- d. Входные данные: Отсутствуют
- e. Ответ:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Status	string	Да	Статус полета	1) NotRunning — симуляция не запущена 2) Running — симуляция запущена, полёт идёт 3) Launching — симуляция в процессе запуска 4) Finished — симуляция

					завершена, полёт окончен 5) Exited — был выполнен предварительны й выход из симуляции
2	IsRunning	bool	Да	Проверка на то, выполняется ли в данный момент симуляция полёта	true
3	LastReached PointOrder	number	Да	Порядковый номер последней точки, которая была достигнута самолётом. Если статус NotRunning, то = -1	2
4	PercentComp leted	number	Да	Процент выполнения полета	0

6.4.3. Получение параметров самолёта

- Метод запроса: GET
- Действие: Выводит текущую геолокацию (местоположение в координатах) самолета и некоторые его параметры.
- URL: /api/flight/properties
- Входные данные: Отсутствуют.
- Ответ:

№	Параметр	Тип данных	Обязател ьность	Описание	Варианты значений
---	----------	---------------	--------------------	----------	----------------------

1	Status	string	Да	Статус полета	1) NotRunning — симуляція не запущена 2) Running — симуляція запущена, політ йде 3) Launching — симуляція в процесі запуску 4) Finished — симуляція завершена, політ закінчено 5) Exited — було виконано попередній вихід з симуляції
2	Longitude	float	Да	Довгота	59.885082
3	Latitude	float	Да	Широта	30.302281
4	AltitudeAgl	float	Да	Істинна висота (над рівнем землі)	859.123
5	Altitude	float	Да	Висота над рівнем моря (в метрах)	912.123
6	Roll	float	Да	Крен	10.12
7	Pitch	float	Да	Тангаж	5.52
8	Heading	float	Да	Курс	182.16
9	Indicated Speed	float	Да	Приборная швидкість	182.16
10	Airspeed	float	Да	Воздушная швидкість	182.16

6.4.4. Завершение симуляции полёта

- a. Метод запроса: GET
- b. Действие: Досрочно завершает симуляцию полета.
- c. URL: /api/flight/exit
- d. Входные данные: Отсутствуют.
- e. Ответ:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Status	string	Да	Статус выполнения запроса	1) Success — симуляция была запущена 2) Failed — во время выполнения произошла ошибка. Подробности в поле Comment
2	Comment	string	Нет	Комментарий от сервера, полученный во время выполнения запроса	Плана полета с указанным идентификатором не существует

6.5. Аналитика

В данной категории находятся запросы, позволяющие взаимодействовать с сессиями полётов, хранящимися в базе данных. Категория расположена в разделе Analytics.

6.5.1. Получение информации о сессиях полётов

- a. Метод запроса: GET
- b. Действие: Выводит информацию обо всех сохраненных сессий полётов.
- c. URL: /api/analytics/sessions
- d. Входные данные: Отсутствуют.

е. Ответ:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Sessions	Array of objects	Да	Массив сессий полёта	[]
1.1	Id	guid	Да	Уникальный идентификатор сессии	54561b7c-d4df-4add-a59a-3b66c8a46dcb
1.2	Title	string	Да	Название сессии	Тестовый полёт 1
1.3	PropertiesReadsPerSecond	number	Да	Кол-во считываний параметров в секунду	10
1.4	DateTimeStart	datetime	Да	Дата и время запуска симуляции	2012-04-23T18:25:43
1.5	DurationSec	number	Да	Длительность полёта в секундах	63
1.6	DateTimeEnd	datetime	Да	Дата и время окончания симуляции	2012-04-23T18:26:46

6.5.2. Получение информации о плане полета по ID

- a. Метод запроса: GET
- b. Действие: Выводит полную информацию о сохраненной сессии полёта по указанному идентификатору.
- c. URL: /api/analytics/sessions/{id:guid}
- d. Route-параметры:
 - id — уникальный идентификатор сессии полёта.
- e. Входные данные: Отсутствуют.
- f. Ответ:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Id	guid	Да	Уникальный идентификатор сессии	54561b7c-d4df-4add-a59a-3b66c8a46dcb
2	Title	string	Да	Название сессии	Test-flight1
3	PropertiesReadsPerSecond	number	Да	Кол-во считываний параметров в секунду	10
4	DateTimeStart	datetime		Дата и время запуска симуляции	2012-04-23T18:25:43
5	DurationSec	number	Да	Длительность полета в секундах	63
6	DateTimeEnd	datetime	Да	Дата и время окончания симуляции	2012-04-23T18:26:46
7	Properties	Array of objects [FlightPropertyShotResponse]	Да	Массив считываний параметров полёта	[]

6.5.3. Удаление сохраненной сессии полёта

- a. Метод запроса: DELETE
- b. Действие: Удаление сохраненной сессии полёта со всеми параметрами по указанному идентификатору.
- c. URL: /api/analytics/sessions/{id:guid}
- d. Route-параметры:
 - id — уникальный идентификатор сессии, которую необходимо удалить
- e. Входные данные: Отсутствуют.
- f. Ответ:

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Status	string	Да	Статус выполнения запроса	1) Success — сессия была удалена 2) Failed — во время выполнения произошла ошибка. Подробности в поле Comment.
2	Comment	string	Нет	Комментарий от сервера, полученный во время выполнения запроса	1) Сессия с указанным идентификатором не существует 2) Сессия успешно удалена

6.6. Многократно используемые модели

В этом подразделе подробнее описываются те модели, которые были представлены в виде типов данных у различных параметров запросов.

6.6.1. FlightPlanResponse

Информация о плане полета, получаемая от сервера. Если взлётная и/или посадочная полосы указаны, то в массиве RoutePoints (№6) они конвертируются в точки маршрута.

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Id	guid	Да	Уникальный идентификатор плана полета	9d974548-7e90-4885-973a-dcdcb92c44f4
2	Title	string	Да	Название плана полета	Test-plan1
3	Remarks	string	Нет	Пользовательские пометки для плана полета	Тестовый маршрут по Санкт-Петербургу
4	Departure Runway	object [RunwayResponse]	Нет	Информация о взлётной полосе в начале маршрута	{}
5	ArrivalRunway	object [RunwayResponse]	Нет	Информация о посадочной полосе в конце маршрута	{}
6	RoutePoints	Array of objects [RoutePointResponse]	Да	Массив точек маршрута	[]

6.6.2. RoutePointResponse

Информация о точке маршрута в плане полета.

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
---	----------	------------	----------------	----------	-------------------

1	Id	guid	Да	Уникальный идентификатор точки маршрута	9d974548-7e90-4885-973a-dcdcb92c44f4
2	Order	number	Да	Порядковый номер точки, отсчет с нуля	1
3	IsEditable	bool	Да	Можно ли редактировать точку. (Если она создана из посадки/взлета, то нельзя)	true
4	Longitude	float	Да	Географ. долгота точки. Отрицательные значения - южная долгота	59.885082
5	Latitude	float	Да	Географ. широта точки. Отрицательные значения — западная широта	30.302281
6	Altitude	float	Да	Высота над уровнем моря в метрах	900
7	Remarks	string	Нет	Дополнительные пользовательские пометки для точки	На этой точке опускаемся

6.6.3. RunwayResponse

Информация о взлётной полосе в аэропорту. Может быть указана как в качестве посадки в конце маршрута, так и в качестве взлета в начале маршрута.

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Id	guid	Да	Уникальный идентификатор взлетной/посадочной полосы	9d974548-7e90-4885-973a-dcdcb92c44f4

2	Title	string	Да	Кодовое название полосы	28L
3	CanBeDeparture	bool	Да	Проверка, может ли полоса быть использована в качестве посадочной полосы в конце маршрута	false
4	CanBeArrival	bool	Да	Проверка, может ли полоса быть использована в качестве взлетной полосы в начале маршрута	true
5	Airport	object	Да	Базовая информация о аэропорте	{}
5.1	Id	guid	Да	Уникальный идентификатор аэропорта	5a4e92f1-9af1-40dd-919f-815b729531cc
5.2	Title	string	Да	Название аэропорта	Пулково
5.3	Code	string	Да	Код аэропорта	ULLI
5.4	City	string	Да	Название города	Санкт-Петербург

6.6.4. AirportResponse

Информация об аэропорте, который добавлен в программу.

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Id	guid	Да	Уникальный идентификатор аэропорта	5a4e92f1-9af1-40dd-919f-815b729531cc

2	Title	string	Да	Название аэропорта	Пулково
3	Code	string	Да	Код аэропорта	ULLI
4	City	string	Да	Название города	Санкт-Петербург
5	Runways	array of objects	Да	Массив из базовой информации о взлетных/посадочных полосах в аэропорте	[]
5.1	Id	guid	Да	Уникальный идентификатор взлетной/посадочной полосы	9d974548-7e90-4885-973a-dcdcb92c44f4
5.2	Title	string	Да	Кодовое название полосы	28L
5.3	CanBeDeparture	bool	Да	Флаг, может ли полоса быть использована в качестве посадочной полосы в конце маршрута	false
5.4	CanBeArrival	bool	Да	Флаг, может ли полоса быть использована в качестве взлетной полосы в начале маршрута	true

6.6.5. FlightPropertiesShotResponse

Снимок значений параметров полёта.

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	DateTime	datetime	Да	Дата и время считывания данных	2012-04-23T18:25:43

2	Longitude	float	Да	Долгота	59.885082
3	Latitude	float	Да	Широта	30.302281
4	AltitudeAgl	float	Да	Истинная высота (над уровнем земли)	859.123
5	Altitude	float	Да	Высота над уровнем моря в метрах	912.123
6	AltitudeIndica tedBaro	float	Да	Относительна я барометричес кая высота	856.123
7	Roll	float	Да	Крен	10.12
8	Pitch	float	Да	Тангаж	5.52
9	Heading	float	Да	Курс	182.16
10	HeadingMagn etic	float	Да	Курс магнитный	180.41
11	HeadingMagn eticIndicated	float	Да	Курс гидромагнитн ый	180.41
12	IndicatedSpee d	float	Да	Приборная скорость	
13	Airspeed	float	Да	Воздушная скорость	
14	VerticalBaroS peed	float	Да	Вертикальная барометричес кая скорость	
15	Mach	float	Да	Число маха	
16	UBodyMps	float	Да	Продольная составляющая земной скорости	
17	VBodyMps	float	Да	Боковая составляющая земной скорости в ЛСК	

18	WBodyMps	float	Да	Вертикальная составляющая земной скорости	
19	PilotOverload	float	Да	Перегрузка пилота	
20	Acceleration Y	float	Да	Боковое ускорение	
21	Acceleration X	float	Да	Продольное ускорение	
22	Acceleration Normal	float	Да	Нормальное ускорение	26.12
23	Temperature	float	Да	Температура воздуха снаружи С	26.12

6.6.6. FlightPropertyReadsResultResponse

Объект представляет все значения считываний определенного параметра внутри одной сессии.

№	Параметр	Тип данных	Обязательность	Описание	Варианты значений
1	Name	string	Да	Название параметра	Долгота
2	Data	Array of objects	Да	Массив всех считываний этого параметра внутри сессии	[]
2.1	Data.Id	number	Да	Порядковый номер считывания значения	0
2.2	Data.Value	float	Да	Считанное значение параметра	30.302281

Внешние ссылки и ресурсы

1. Архив с ПКУП- <https://disk.yandex.ru/d/bugCqNzR7Rk7sw>
2. Редактор кода Visual Studio Code- <https://code.visualstudio.com/>
3. Установка NodeJS- <https://nodejs.org/en/download/package-manager>
4. Утилита для создания баз данных PostgreSQL-
<https://www.postgresql.org/>
5. Установщик .NET SDK- <https://dotnet.microsoft.com/en-us/download/dotnet/8.0>
6. Авиасимулятор FlightGear- <https://www.flightgear.org/download/>
7. Файл для создания базы данных и архив с самолётом Т-50 (СУ-57)-
<https://disk.yandex.ru/d/bugCqNzR7Rk7sw>
8. Доступные команды изменения симуляции-
https://wiki.flightgear.org/Command_line_options
9. GitHub репозиторий для установки местности России во FlightGear-
<https://github.com/soitanen/Project-Russia>
10. О кодах ICAO-
https://ru.wikipedia.org/wiki/Международная_организация_гражданск_ой_авиации
11. О зоне ожидания- https://www.avsim.ru/wiki/Зона_ожидания