

**РУКОВОДСТВО СИСТЕМНОГО ПРОГРАММИСТА ПРОГРАММНОГО
КОМПЛЕКСА «ИНТЕГРАЛ-Д»**

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ.....	3
1.1. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ.....	3
2. СТРУКТУРА ПРОГРАММЫ	14
2.1. Логическая структура программного комплекса.....	14
2.2. Структура демонстратора и управляющей платформы	16
3. НАСТРОЙКА ПРОГРАММЫ	50
3.1. Первоначальная установка демонстратора программного комплекса «Интеграл-Д»	50
3.2. Обновление версии демонстратора программного комплекса «Интеграл-Д»	51
3.3. Сборка сообщений об ошибках	52
3.4. Изменение порта веб-сервера. Настройка SSL.	54
3.5. Создание пользователя с правами админа.....	56
3.6. Одновременная работа разных версий демонстратора программного комплекса «Интеграл-Д»	57
3.7. Откат версии демонстратора программного комплекса «Интеграл-Д»	57
4. ПРОВЕРКА ПРОГРАММЫ.....	57
4.1. Проверка доступности контейнера project_apr	58
4.2. Проверка доступности функционала организаций.....	60
4.3. Проверка доступности функционала проектов.....	61
4.4. Проверка доступности функционала проверки авторизации	61
4.5. Проверка доступности функционала аутентификации	63
4.6. Проверка работоспособности	63
4.7. Проверка карт	66
4.8. Создание проекта	67
4.9. Получение логов контейнера	70
4.10. Анализ логов project_apr.....	70
5. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА.....	71

1. ОБЩИЕ СВЕДЕНИЯ

Демонстратор позволяет моделировать функционирование многоспутниковых группировок ДЗЗ, контроля околоземного космического пространства и систем связи. В ходе моделирования демонстратор проводит расчет баллистического движения всех космических объектов, моделирование съемки целевых объектов ДЗЗ, моделирование съемки целевых космических объектов, моделирование передачи данных на наземные точки приема информации, в том числе через межспутниковые каналы передачи данных, и расчет интегральных характеристик работы всей многоспутниковой КС.

Входные и выходные данные могут быть введены и продемонстрированы с помощью графического интерфейса пользователя.

Программа написана на языке разработки Python 3.7 и C++ (-std=c++17).

1.1. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

Созданная программа «Демонстратор программного комплекса для проектирования многоспутниковых группировок» обладает обширным функционалом, может использоваться для обширного спектра назначений и решает несколько классов задач.

1.1.1. Назначение программы

Для программы может быть выделено несколько основных сценариев.

- Проектирование перспективных космических систем, в том числе многоспутниковых с учетом межспутниковых каналов связи.
- Обоснование тактико-технического облика перспективных космических систем, в том числе многоспутниковых, и оценка эффективности их применения.
- Уточнение показателей эффективности целевого применения существующей космической группировки и ее отдельных сегментов (космических систем и комплексов).
- Оценка эффективности работы многоспутниковых систем в случае выхода из строя ее компонент.

– Проведение независимой научно-технической экспертизы проектных решений по созданию перспективных космических систем.

– Оценка эффекта от разработки и внедрения перспективных элементов конструкции КА.

1.1.2. Классы решаемых задач

В разделе мы приведем несколько классов решаемых задач, которые скомпонуем в несколько отдельных наборов задач.

1.1.2.1. Задачи определения возможностей заданной группировки КА

В данном разделе рассматривается набор задач по определению возможностей заданной группировки КА.

Определение возможностей решать целевую задачу ДЗЗ одиночным КА.

Входные параметры, задаваемые пользователем: баллистическое положение КА, характеристики полезной нагрузки, целевые зоны для съемки, время моделирования и настройки алгоритмов для расчета баллистического движения КА, настройки алгоритмов расчета ДЗЗ.

В ходе расчета определяется баллистическое движение КА, определяется способность КА решать целевую задачу, по полученным результатам определяются целевые показатели съемки объектов ДЗЗ.

Выходные параметры: целевые показатели съемки целевых объектов ДЗЗ с помощью космического аппарата.

Определение возможностей решать целевую задачу СКО одиночным КА.

Входные параметры, задаваемые пользователем: баллистическое положение КА, характеристики полезной нагрузки, баллистическое построение целевых объектов (может быть сгенерировано случайным образом), целевые объекты для наблюдения, время моделирования и настройки алгоритмов для расчета баллистического движения КА и целевых объектов, настройки алгоритмов расчета СКО.

В ходе расчета определяется баллистическое движение КА, определяется баллистическое движение всех целевых объектов, определяется способность КА наблюдать целевые объекты, по полученным результатам определяется целевые показатели съемки объектов ДЗЗ.

Выходные параметры: целевые показатели съемки целевых объектов с помощью космического аппарата.

Определение возможностей решать целевую задачу ДЗЗ группировкой КА.

Входные параметры, задаваемые пользователем: баллистическое положение всех КА, характеристики полезной нагрузки всех КА, целевые зоны для съемки, время моделирования и настройки алгоритмов для расчета баллистического движения КА, настройки алгоритмов расчета ДЗЗ.

В ходе расчета определяется баллистическое движение всех КА, определяется способность каждого КА решить целевую задачу, по полученным результатам определяются целевые показатели съемки объектов ДЗЗ.

Выходные параметры: целевые показатели съемки целевых объектов ДЗЗ с помощью группы аппаратов, общие и для каждого КА в отдельности.

Определение возможностей решать целевую задачу СКО группировкой КА.

Входные параметры, задаваемые пользователем: баллистическое положение всех КА, характеристики полезной нагрузки каждого КА, баллистическое построение целевых объектов (может быть сгенерировано случайным образом), целевые объекты для наблюдения, время моделирования и настройки алгоритмов для расчета баллистического движения КА и целевых объектов, настройки алгоритмов расчета СКО.

В ходе расчета определяется баллистическое движение всех КА, определяется баллистическое движение всех целевых объектов, определяется способность всех КА наблюдать целевые объекты, по полученным результатам определяются целевые показатели съемки целевых объектов.

Выходные параметры: целевые показатели съемки целевых объектов с помощью группировки КА и для каждого КА отдельно.

Определение возможностей решать целевую задачу ДЗЗ одиночным КА с учетом систем связи.

Входные параметры, задаваемые пользователем: баллистическое положение КА, характеристики полезной нагрузки, целевые зоны для съемки, время моделирования, информация о системах связи, информация о наземных пунктах приема информации и настройки алгоритмов для расчета баллистического движения КА, настройки алгоритмов расчета ДЗЗ, настройки работы систем связи.

В ходе расчета определяется баллистическое движение КА, определяется способность КА решить целевую задачу, определяется возможность КА передавать целевую информацию на наземные пункты приема, по полученным результатам определяются целевые показатели съемки объектов ДЗЗ.

Выходные параметры: целевые показатели съемки целевых объектов ДЗЗ с помощью космического аппарата, дополненные показателями, связанными с передачей данных на наземные пункты приема.

Определение возможностей решать целевую задачу СКО одиночным КА с учетом систем связи.

Входные параметры, задаваемые пользователем: баллистическое положение КА, характеристики полезной нагрузки, баллистическое положение космических объектов, целевые объекты, время моделирования, информация о системах связи, информация о наземных пунктах приема информации и настройки алгоритмов для расчета баллистического движения КА, настройки алгоритмов расчета СКО, настройки работы систем связи.

В ходе расчета определяется баллистическое движение КА, определяется баллистическое движение всех целевых объектов, определяется способность КА наблюдать целевые объекты, определяется возможность КА передавать целевую информацию на наземные пункты приема, по полученным результатам определяются целевые показатели съемки целевых объектов.

Выходные параметры: целевые показатели съемки целевых объектов СКО с помощью космического аппарата, дополненные показателями, связанными с передачей данных на наземные пункты приема.

Определение возможностей решать задачу связи группой КА.

Входные параметры, задаваемые пользователем: баллистическое положение всех КА, характеристики полезной нагрузки каждого КА, положение и свойства наземных абонентов связи, характерные параметры передаваемой информации абонентами (данные по абонентам могут быть сгенерированы автоматически с задаваемыми пользователем ограничениями), время моделирования, информация о системах связи, информация о наземных пунктах приема информации и настройки алгоритмов для расчета баллистического движения КА, настройки алгоритмов расчета передачи данных, настройки работы систем связи, выбор алгоритмов маршрутизации.

В ходе расчета определяется баллистическое движение каждого КА, определяется моменты связи всех КА и всех абонентов, определяется возможность каждого КА передавать целевую информацию абонентам, по полученным результатам определяется целевые показатели группировки связи.

Выходные параметры: целевые показатели группировки в обеспечении связи абонентам.

Определение возможностей решать задачу ДЗЗ группой КА с учетом системы связи.

Входные параметры, задаваемые пользователем: баллистическое положение всех КА, характеристики полезной нагрузки каждого КА, характеристики систем связи КА, включая системы межспутниковой связи, время моделирования, информация о наземных пунктах приема информации и настройки алгоритмов для расчета баллистического движения КА, настройки алгоритмов расчета ДЗЗ, настройки работы систем связи, выбор алгоритмов маршрутизации.

В ходе расчета определяется баллистическое движение всех КА, определяется способность каждого КА решить целевую задачу, определяется возможность каждого КА передавать целевую информацию на наземные пункты

приема информации, по полученным результатам определяются целевые показатели съемки объектов ДЗЗ.

Выходные параметры: целевые показатели группировки ДЗЗ для группировки в целом и для отдельных КА, с учетом времени передачи информации на наземные пункты приема информации.

Определение возможностей решать задачу СКО группой КА с учетом системы связи.

Входные параметры, задаваемые пользователем: баллистическое положение всех КА, характеристики полезной нагрузки каждого КА, характеристики систем связи КА, включая системы межспутниковой связи, баллистическое положение всех объектов наблюдения для системы СКО, время моделирования, информация о наземных пунктах приема информации и настройки алгоритмов для расчета баллистического движения КА, настройки алгоритмов расчета СКО, настройки работы систем связи, выбор алгоритмов маршрутизации.

В ходе расчета определяется баллистическое движение каждого КА, определяется баллистическое движение всех целевых объектов, определяется способность всех КА наблюдать целевые объекты, определяется возможность каждого КА передавать целевую информацию на наземные пункты приема, по полученным результатам определяются целевые показатели съемки объектов СКО.

Выходные параметры: целевые показатели группировки СКО для группировки в целом и для отдельных КА, с учетом времени передачи информации на наземные пункты приема информации.

1.1.2.2. Задачи по проектированию группировки КА для целевых задач

Как следует из названия раздела, эта задача посвящена созданию проекта группировки КА, нескольких проектов группировок или отдельных КА для выполнения заданной целевой задачи.

Ключевой особенностью этой категории задач является то, что они стоят неоднозначно, так как каждая поставленная целевая задача может иметь

множество группировок или аппаратов, удовлетворяющих поставленным целям. Кроме того, построение группировки под задачу, как правило, ставит перед собой задачу оптимизации и поиска наилучшего или рационального решения поставленной задачи без подробного описания требуемых от группировки характеристик. Расчет происходит в 2 шага. Первый – подбирается группировка, второй – проверяется подходит ли она под задачу; проверка подходящих решений проводится на основании прямых задач, описанных в разделе 1.1.2.1.

Подбор одиночного КА для решения целевой задачи ДЗЗ одиночным КА.

Входные параметры, задаваемые пользователем: целевые области ДЗЗ, диапазон и шаг для многовариантных расчетов (многовариантный расчет может вестись при использовании в одном расчете множества аппаратов).

В ходе расчета определяется баллистическое движение КА для каждого варианта, определяется способность КА решить целевую задачу в каждом варианте, по полученным результатам определяются максимальные достижимые целевые показатели. Пользователь должен выбрать наиболее подходящие варианты баллистического положения КА и провести проверку полученных характеристик путем решения задачи из раздела 1.1.2.1.

Выходные параметры: оптимальное баллистическое положение КА, достижимые целевые показатели съемки целевых объектов ДЗЗ с помощью космического аппарата.

Подбор одиночного КА для решения целевой задачи СКО.

Входные параметры, задаваемые пользователем: характеристики полезной нагрузки, баллистическое построение целевых объектов (может быть сгенерировано случайным образом), целевые объекты для наблюдения, время моделирования и настройки алгоритмов для расчета баллистического движения КА и целевых объектов, настройки алгоритмов расчета СКО, диапазон значений баллистического построения или расположения полезной нагрузки на космическом аппарате.

В ходе расчета определяется баллистическое движение КА для каждого варианта, определяется способность КА решить целевую задачу в каждом

варианте, по полученным результатам определяются максимальные достижимые целевые показатели. Пользователь должен выбрать наиболее подходящие варианты баллистического положения КА и/или расположения полезной нагрузки на КА и провести проверку полученных характеристик путем решения задачи из раздела 1.1.2.1.

Выходные параметры: оптимальное баллистическое положение КА, достижимые целевые показатели съемки целевых объектов СКО с помощью космического аппарата.

Подбор группировки КА для решения задачи ДЗЗ.

Входные параметры, задаваемые пользователем: характеристики полезной нагрузки всех КА, целевые зоны для съемки, время моделирования и настройки алгоритмов для расчета баллистического движения КА, настройки алгоритмов расчета ДЗЗ и диапазон значений баллистического построения КА в группировке.

В ходе расчета проверяется способность решать целевую задачу с помощью группировки КА, в каждом варианте расчета определяется баллистическое движение всех КА, определяется способность каждого КА решить целевую задачу, по полученным результатам определяются достигаемые целевые показатели съемки объектов ДЗЗ. Пользователь должен выбрать наиболее подходящие варианты баллистического положения КА и провести проверку полученных характеристик путем решения задачи из раздела 1.1.2.1.

Выходные параметры: варианты баллистического построения группировки ДЗЗ и целевые показатели съемки целевых объектов ДЗЗ, общие и для каждого КА в отдельности.

Подбор группировки КА для решения задачи СКО.

Входные параметры, задаваемые пользователем: характеристики полезной нагрузки всех КА, целевые объекты для наблюдения в задаче СКО, время моделирования и настройки алгоритмов для расчета баллистического движения КА, настройки алгоритмов расчета СКО и диапазон значений баллистического построения КА в группировке и/или расположения полезной нагрузки на отдельных КА.

В ходе расчета проверяется способность решать целевую задачу с помощью группировки КА, в каждом варианте расчета определяется баллистическое движение всех КА, определяется способность каждого КА решить целевую задачу, по полученным результатам определяются достигаемые целевые показатели съемки объектов СКО. Пользователь должен выбрать наиболее подходящие варианты баллистического положения КА и/или расположения полезной нагрузки на КА и провести проверку полученных характеристик путем решения задачи из раздела 1.1.2.1.

Выходные параметры: варианты баллистического построения группировки СКО и целевые показатели съемки целевых объектов СКО, общие и для каждого КА в отдельности.

Подбор одиночного КА для решения задачи ДЗЗ с учетом систем связи.

Входные параметры, задаваемые пользователем: характеристики полезной нагрузки всех КА, целевые зоны для съемки, время моделирования и настройки алгоритмов для расчета баллистического движения КА, настройки алгоритмов расчета ДЗЗ, алгоритмы расчета передачи данных, расположение наземных пунктов приема информации, свойства систем связи на космическом аппарате и диапазон значений баллистического построения КА в группировке.

В ходе расчета проверяется способность решать целевую задачу с помощью КА, в каждом варианте расчета. Определяется баллистическое движение КА, определяется способность КА решить целевую задачу, определяется маршрут передачи данных на наземные пункты приема информации, по полученным результатам определяются достигаемые целевые показатели съемки объектов ДЗЗ. Пользователь должен выбрать наиболее подходящие варианты баллистического положения КА и провести проверку полученных характеристик путем решения задачи из раздела 1.1.2.1.

Выходные параметры: варианты баллистического положения КА ДЗЗ и целевые показатели съемки целевых объектов ДЗЗ.

Подбор одиночного КА для решения задачи СКО с учетом систем связи.

Входные параметры, задаваемые пользователем: характеристики полезной нагрузки всех КА, ориентация элементов полезной нагрузки на КА, баллистическое положение целевых объектов, время моделирования и настройки алгоритмов для расчета баллистического движения КА, настройки алгоритмов расчета СКО, алгоритмы расчета передачи данных, расположение наземных пунктов приема информации, свойства систем связи на космическом аппарате и диапазон значений баллистического построения КА в группировке.

В ходе расчета проверяется способность решать целевую задачу с помощью КА, в каждом варианте расчета. Определяется баллистическое движение КА, определяется способность КА решить целевую задачу, определяется маршрут передачи данных на наземные пункты приема информации, по полученным результатам определяются достигаемые целевые показатели съемки целевых объектов СКО. Пользователь должен выбрать наиболее подходящие варианты баллистического положения КА и провести проверку полученных характеристик путем решения задачи из раздела 1.1.2.1.

Выходные параметры: варианты баллистического положения КА СКО, целевые показатели съемки целевых объектов СКО и время передачи их на Землю.

Определение возможностей решать задачу связи группой КА.

Входные параметры, задаваемые пользователем: диапазон баллистического положение всех КА, характеристики полезной нагрузки каждого КА, положение и свойства наземных абонентов связи, характерные параметры передаваемой информации абонентами, (данные по абонентам могут быть сгенерированы автоматически с задаваемыми пользователем ограничениями), время моделирования, информация о системах связи, информация о наземных пунктах приема информации и настройки алгоритмов для расчета баллистического движения КА, настройки алгоритмов расчета ДЗЗ, настройки работы систем связи, выбор алгоритмов маршрутизации.

В ходе расчета проверяется способность решать целевую задачу с помощью группы КА, в каждом варианте расчета. Определяется баллистическое движение каждого КА, определяется способность КА решить целевую задачу, определяется

маршрут передачи данных на наземные пункты приема информации, по полученным результатам определяются достигаемые целевые показатели связи для абонентов. Пользователь должен выбрать наиболее подходящие варианты баллистического положения КА и провести проверку полученных характеристик путем решения задачи из раздела 1.1.2.1.

Выходные параметры: целевые показатели группировки в обеспечении связи абонентам для всех вариантов группировки.

Определение возможностей решать задачу ДЗЗ группой КА с учетом системы связи.

Входные параметры, задаваемые пользователем: диапазон баллистического положение всех КА, характеристики полезной нагрузки каждого КА, целевые объекты ДЗЗ, время моделирования, информация о системах связи, информация о наземных пунктах приема информации и настройки алгоритмов для расчета баллистического движения КА, настройки алгоритмов расчета ДЗЗ, настройки работы систем связи, выбор алгоритмов маршрутизации.

В ходе расчета проверяется способность решать целевую задачу с помощью группы КА, в каждом варианте расчета. Определяется баллистическое движение каждого КА, определяется способность КА решить целевую задачу, определяется маршрут передачи данных на наземные пункты приема информации, по полученным результатам определяются достигаемые целевые показатели задачи ДЗЗ. Пользователь должен выбрать наиболее подходящие варианты баллистического положения КА и провести проверку полученных характеристик путем решения задачи из раздела 1.1.2.1.

Выходные параметры: целевые показатели группировки в решении задачи ДЗЗ для всех вариантов группировки.

Определение возможностей решать задачу СКО группой КА с учетом системы связи.

Входные параметры, задаваемые пользователем: диапазон баллистического положение всех КА, характеристики полезной нагрузки каждого КА, баллистическое положение целевых объектов для задачи СКО, время

моделирования, информация о системах связи, информация о наземных пунктах приема информации и настройки алгоритмов для расчета баллистического движения КА, настройки алгоритмов расчета ДЗЗ, настройки работы систем связи, выбор алгоритмов маршрутизации.

В ходе расчета проверяется способность решать целевую задачу с помощью группы КА, в каждом варианте расчета. Определяется баллистическое движение каждого КА, определяется баллистическое движение целевых объектов СКО, определяется способность КА решить целевую задачу, определяется маршрут передачи данных на наземные пункты приема информации, по полученным результатам определяются достигаемые целевые показатели решения задачи СКО. Пользователь должен выбрать наиболее подходящие варианты баллистического положения КА и провести проверку полученных характеристик путем решения задачи из раздела 1.1.2.1.

Выходные параметры: целевые показатели группировки в решении задачи СКО для всех вариантов группировки.

2. СТРУКТУРА ПРОГРАММЫ

2.1. Логическая структура программного комплекса

При работе программы «Интеграл-Д» решаются множество отдельных задач. Общая схема устройства комплекса продемонстрирована на рисунке 2-1.

Комплекс состоит из следующих блоков.

Блок баллистики осуществляет баллистический расчет движения космических объектов и аппаратов.

Блок формирования сети связи формирует информацию по потенциальным системам связи между наземными станциями и КА.

Блок оптики для расчета наблюдения за наземными целевыми объектами и космическими целевыми объектами СКО.

Блок ЧТЗ используется для формирования заданий для задания для каждого космического корабля из группировки.

Блок интерфейса используется для ввода и вывода данных.

Блок визуализации является подблоком интерфейса. Он осуществляет отображение движения КА и выполнение ими целевой задачи и передачи данных.

Блок базы данных содержит все данные, используемые в процессе расчета.

Блок имитационного моделирования проводит расчет работы отдельных элементов космического аппарата.

Блок анализа результатов проводит подсчет достигаемых целевых показателей по результатам моделирования.

Блок управляющей платформы осуществляет взаимодействие всех блоков друг с другом.

Каждый из блоков имеет свой API для взаимодействия с другими блоками. Использование блоков позволяет решать все требуемые от демонстратора задачи.

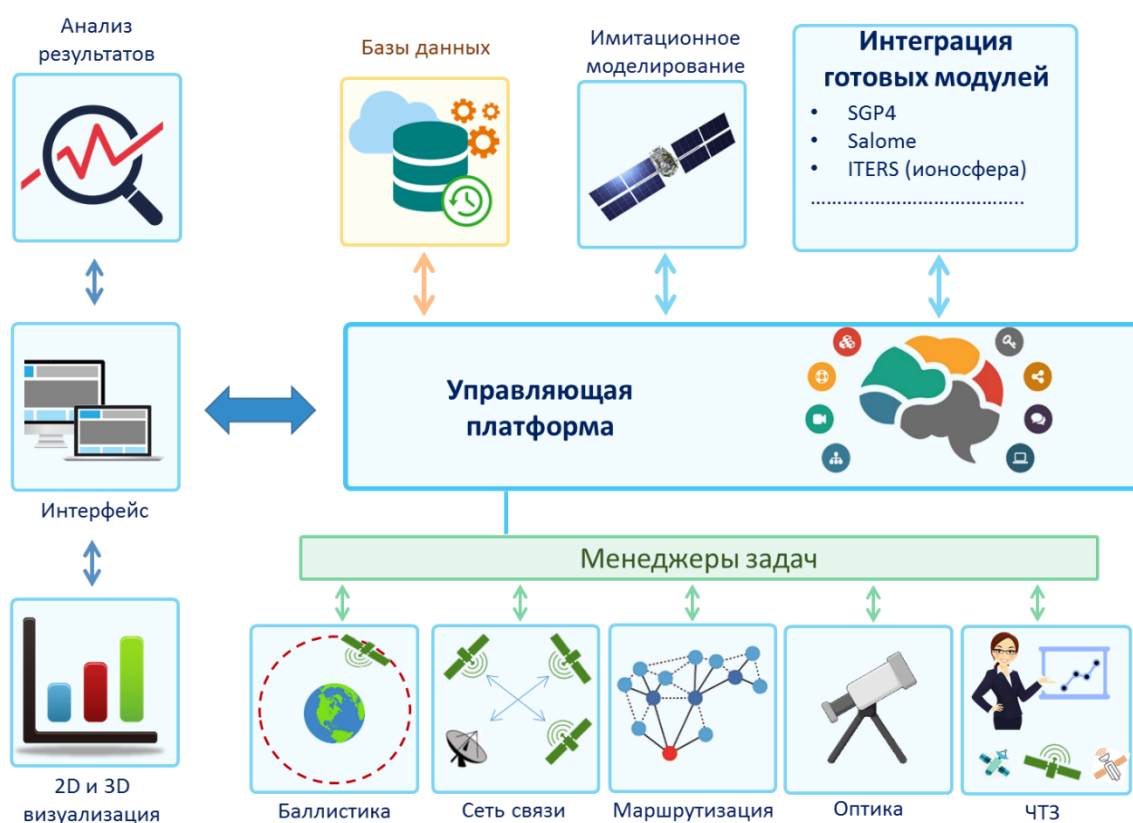


Рисунок 2-1 – Общий комплект блоков демонстратора программного комплекса «Интеграл-Д»

При работе программного комплекса в общем случае реализуется следующий алгоритм действий:

- Получение входных данных через интерфейс.
- Расчет баллистического движения всех космических объектов, с помощью выбранного пользователем набора алгоритмов.
- Расчет зон прямой видимости аппаратов ДЗЗ и СКО целевых объектов ДЗЗ и СКО соответственно.
- Расчет прямой видимости между аппаратами для систем связи.
- Формирование заданий для отдельных КА по решению целевых задач.
- Расчет маршрутизации данных, получаемых при решении задачи ДЗЗ, СКО или связи.
- Формирование расписания работы отдельных элементов КА.
- Проведение имитационного эксперимента работы отдельных элементов КА.
- Анализ полученных результатов.

2.2. Структура демонстратора и управляющей платформы

Общая структура программы предполагает использование клиент-серверной архитектуры, на клиентской части находится frontend, на серверной -- backend.

Логической основой программного комплекса Интеграл-Д является модуль «Центральная управляющая платформа». Однако его реальная структура несколько отличается от логической. Общее представление о структуре отображено на рисунке 2-2.

запускаются строго последовательно и в текущей версии последовательность реализации блоков не может быть изменена. В дальнейшем, может быть реализована возможность перестановки последовательности выполнения основных элементов программы.

Отдельными элементами являются карты, для реализации которых используются готовые свободно распространяемые библиотеки. Для работы с картами используются отдельные методы. Карты внутри интерфейса используются на соответствующих вкладках frontend, тепловые карты используются для демонстрации итоговых результатов по результатам расчетов, расчеты по визуализации тепловых карт и карт интерфейса делается непосредственно на frontend, без передачи вычислений на Project API.

2.2.1. Описание клиента

Структура frontend-приложения представляет собой совокупность, как правило, трех компонентов:

1. данных, введенных пользователем или пришедших со стороны сервера;
2. визуальных элементов для отображения этих данных;
3. прослойки, которая посредством манипуляции данными позволяет осуществлять взаимодействие двух предыдущих компонентов.

Такое разделение концепций позволяет достичь наибольшей независимости элементов внутри приложения, повышая их заменяемость и переиспользуемость. Визуализация данной концепции представлена на рисунке 2-3. В англоязычной литературе, такая концепция известна как MVC.

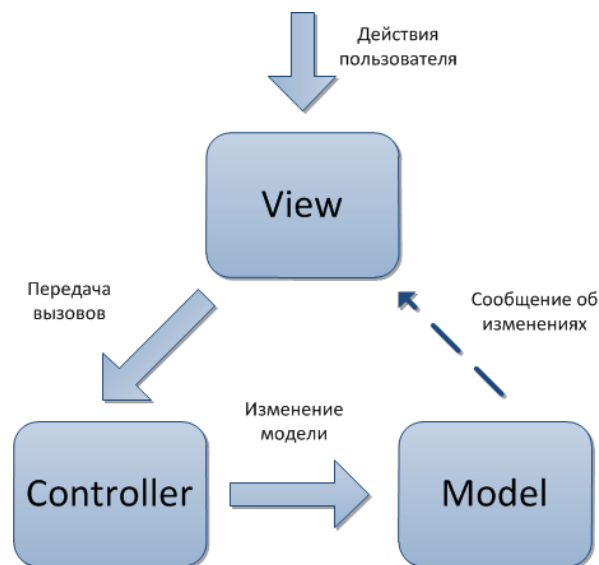


Рисунок 2-3 - Схема концепта MVC.

Для того, чтобы спроектировать программу с применением подобной парадигмы и успешно ее реализовать, необходимо определить правила, согласно которым каждому компоненту назначается область ответственности и условия поведения на границах соприкосновения этих областей.

Клиент «Интеграла» написан с использованием библиотеки React, представляющей собой набор утилит, позволяющих создавать реактивные приложения, т.е. такие приложения, поведение которых - это реакция на определенные действия, совершенные пользователем или на изменение данных. Абстрагирование же самих данных осуществляется при помощи вспомогательной библиотеки Redux, определяющей модель, в соответствии с которой данные можно получать от сервера и хранить независимо от визуальной части.

Управление данными и адаптирование их к условиям контекста выполняется путем создания переходных структур, преобразующих исходные данные в понятный для пользователя формат и, наоборот, конвертируя данные от пользователя в адекватный вид для отправки на хранение. Например, ключевой такой структурой на страницах ввода данных является компонент динамической формы (DynamicForm), который представляет из себя набор разрозненных элементов, которые по необходимости применяются для построения формы ввода данных на основе текстового файла (json), приходящего от сервера. Одной из особенностей данной структуры является то, что элементы, из которых она

состоит, ничего "не знают" друг о друге и выполняют роль автономных исполнителей или модулей, что позволяет легко изменять состав и содержание этих элементов, не вовлекая в этот процесс свое окружение.

Динамическая форма - это один из примеров необходимости деления целого на независимые составляющие при проектировании комплексного приложения. Поэтому, развивая идею модульности, элементы, созданные и стилизованные специально для проекта "Интеграл-Д", также собраны и выделены в отдельную библиотеку с целью минимизации зависимости основного приложения от меняющихся условий, в которых развивается проект.

2.2.2. Описание серверной части «Интеграла»

Серверная часть демонстратора является приложением, написанным на языке Python, взаимодействие с которым осуществляется через REST API (Representational State Transfer / «передача репрезентативного состояния»). Пример клиент-серверной архитектуры представлен на рисунке 2-4.

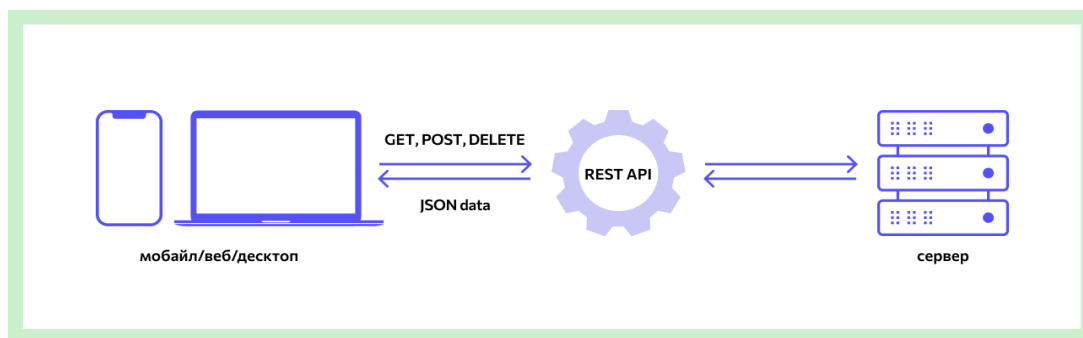


Рисунок 2-4: Схема взаимодействия клиента с сервером.

Серверная часть состоит из нескольких отдельных сервисов:

Project-API — основной сервис, работающий с такими сущностями как проекты, расчеты, варианты расчета, а также проводящий все расчеты;

MapTileServer — сервис для получения фрагментов геополитических карт;

MapService — сервис для создания тепловых карт.

Каждый из этих сервисов работает в своем Docker-контейнере и взаимодействуют с клиентом и другими сервисами, используя REST API. Часть методов API изображена на рисунке 2-5.

Для реализации API используется фреймворк FastAPI в связке с pydantic для удобной валидации и преобразования заданных пользователем данных. Этот фреймворк позволяет легко работать с запросами и автоматически валидировать и конвертировать данные из тела запроса.

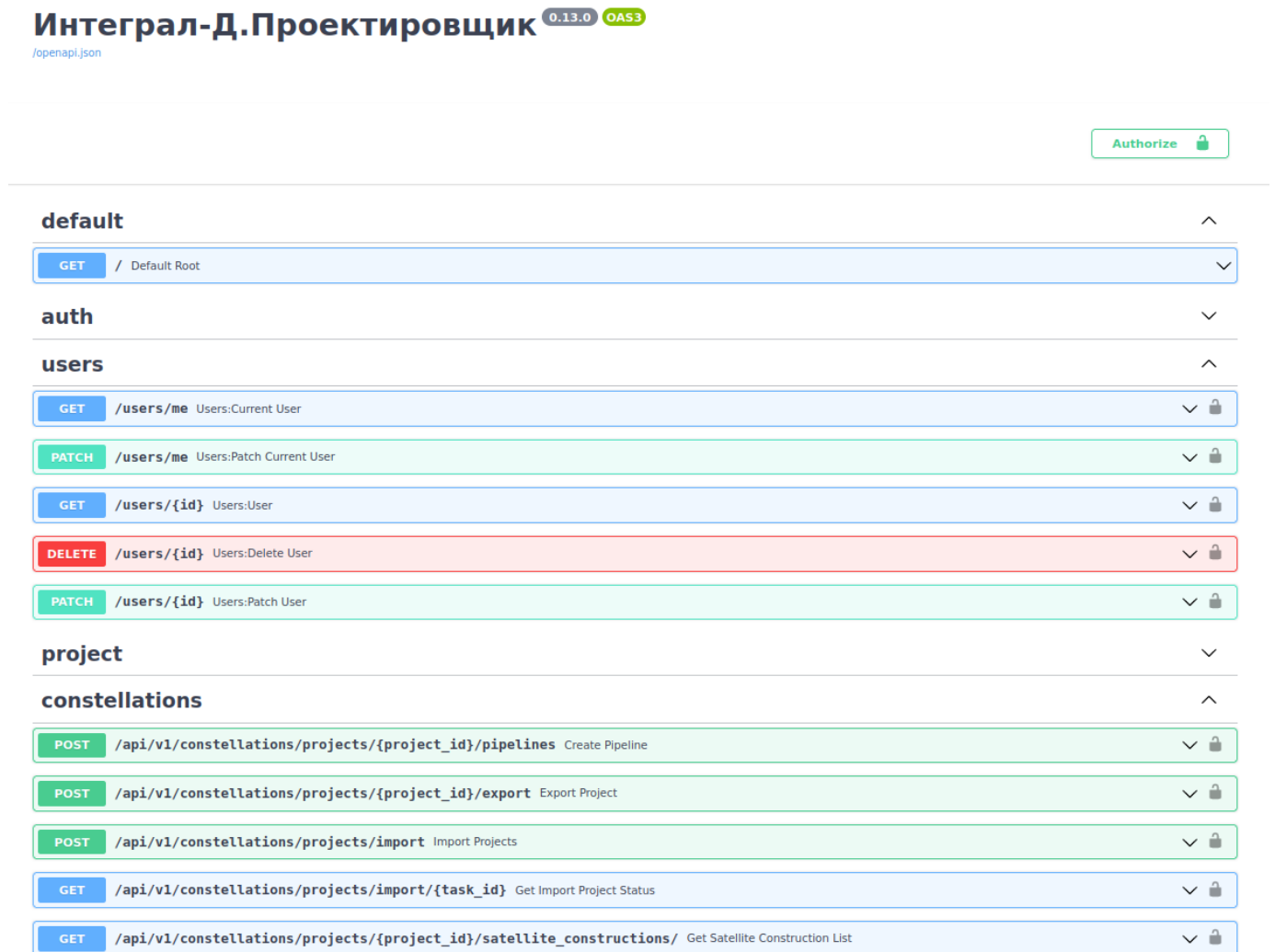


Рисунок 2-5: Пример маршрутов API, используемых в Project-API

2.2.3. Сервис Project-API

Этот сервис является ядром серверной части программы — центральным звеном, к которому происходит подключение всех остальных модулей и плагинов, а также является хранилищем и менеджером всех моделей данных, используемых в расчетах.

Сервис представляет из себя набор плагинов, каждый из которых является небольшим приложением со своим списком API. Среди них есть главный, который занимается поиском всех других подключенных плагинов, инициализацией и построением иерархии контейнеров. На самом нижнем уровне находится Basic-plugin, в котором происходит инициализация баз данных. К нему происходит подключение плагинов пользователей, проектов, контейнер задач и контейнер нодового редактора.

Плагины пользователей и проектов довольно просты. Они отвечают только за создание соответствующих моделей в базе и за дальнейшую работу с этими сущностями.

Модель пользователя содержит его логин, хешированный пароль, а также данные ФИО и внутренние идентификаторы принадлежности к группам пользователей.

Модель проекта состоит полей имени и описания, а также на эту модель ссылаются все другие ресурсы, которые принадлежат этому проекту (Компоненты КА, Конструкции КА, Цели ДЗЗ и другие).

Контейнер задач содержит в себе все те модели, что формируют задачи и всю логику проведения расчетов. Здесь хранятся сущности, которые формируют задачу: Цели ДЗЗ, Наземные станции, Абоненты, Задачи отправки сообщений, Целевые объекты СКО, а также сущности, описывающие группировки: Компоненты КА, Конструкции КА, Группировки КА. Все данные вводимые в интерфейсе сохраняются в базе данных под управлением MongoDB.

В данный момент реализовано 3 основных задачи: Задача ДЗЗ, Задача Связи и Задача СКО.

При запуске расчета происходит сборка графа зависимостей, который состоит из всех ресурсов проекта. Далее происходит анализ имеющихся частей проекта, по результатам которого расчет может сразу завершиться со статусом «пропущен» в случае, если были заданы не все необходимые данные. При запуске расчета граф зависимостей, собранный по проекту, пересобирается заново, но привязывается уже к конкретной версии расчета. Так, на рисунке 2-6 представлен

упрощенный граф проекта (слева) и пересобранный граф расчета. Цветом компонент подчеркивается то, что все узлы графа были пересозданы заново (имеют другой идентификатор и хранятся уже отдельно от своих прообразов) и принадлежат уже не проекту, а варианту расчета.

За счет этого можно перезапускать расчет с разными параметрами, и каждый раз все данные будут сохранены, к варианту расчета можно будет вернуться.

Также, за счет этого механизма реализованы многовариантные расчеты. В этом случае при запуске расчета собираются сразу несколько новых графов – вариантов расчета, каждый из которых имеет свои параметры.

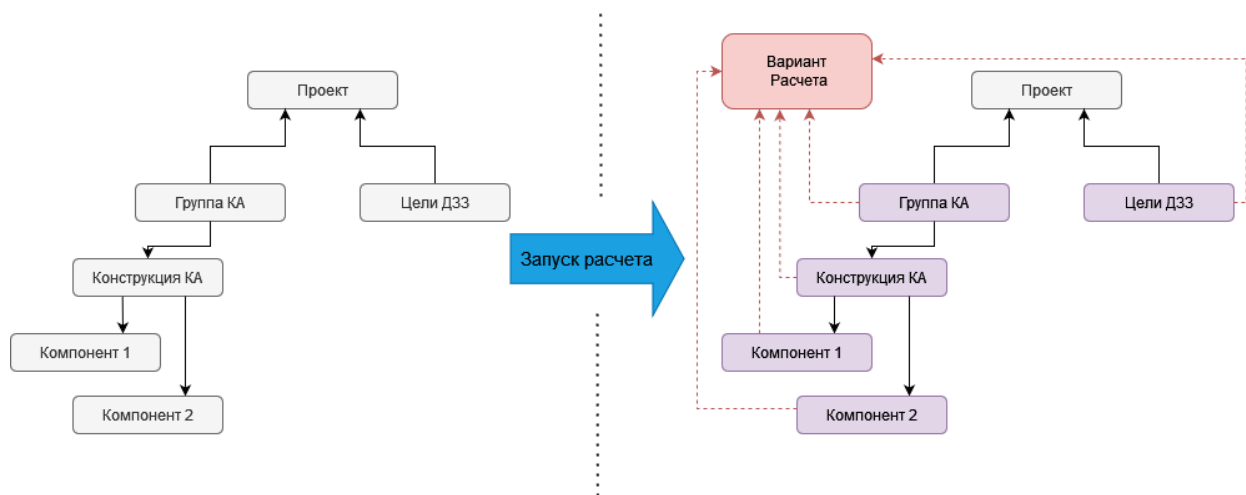


Рисунок 2-6: Пересборка графа зависимостей при запуске расчета.

Также при запуске расчета создается сам объект расчета, который является последовательной цепочкой задач (Расчет баллистики, Расчет ДЗЗ, Расчет СКО, Расчет связи). Каждая такая цепочка после создания получает себе граф входных данных и попадает в распределенную очередь задач Celery, где дожидается, пока под нее будет выделен процесс. После чего уже она перейдет к исполнению.

Все выполняемые задачи состоят из 3 этапов:

1. Этап загрузки данных.

На нем происходит загрузка из базы необходимых для конкретной задачи входных данных. Для этого выполняется обход графа ресурсов этого варианта расчета. В случае, если данные неполны, следующие этапы пропускаются, и производится запуск следующей задачи.

2. Этап вычислений.

На нем вызываются расчётные модули, свои для каждой задачи. Во время расчета задача будет находиться в статусе «в процессе». После завершения собираются выходные данные.

3. Этап сохранения результатов.

Происходит сохранение результатов расчёта в соответствующие хранилища. Если выходных данных много, то этот этап тоже может занимать продолжительное время.

Описание баз данных.

В проекте используются базы данных под управлением различных СУБД:

1. MongoDB. Документоориентированная система управления базами данных.

В этой базе хранятся все вводимые в интерфейс данные, а также данные предназначенные для вывода обратно в интерфейс. Позволяет работать со своими объектами как со словарями, а также фильтровать элементы по полям объектов.

2. PostgreSQL. Реляционная база данных, которая хранит таблицы данных.

В ней возможно удобно хранить массивы простых данных неизвестной длины.

Отлично подходит для хранения данных для графиков, а также сырых данных, полученных в расчете. Также в ней хранятся все баллистические траектории объектов.

3. Redis. База данных работающая по принципу «ключ – значение». Она используется для формирования очередей задач;

4. MINIO. Объектное хранилище. Представляет собой древовидную файловую систему, которая используется для хранения изображений (тепловые карты), экспортируемых проектов, а также других файловых данных.

2.2.4. Сервис тайлов геополитических карт

Для отображения в интерфейсе регионов зондирования, местоположений наземных станций и абонентов, а также удобной навигации по географическим координатам, требовался встроенный в демонстратор сервис карт, охватывающий всю планету. Причем, карта Земли должна быть достаточно подробной и с

2.2.5. Сервис тепловых карт

Часто оказывается необходимо вывести пользователю информацию по какому-то региону в формате тепловой карты (см. рисунок 2-8).

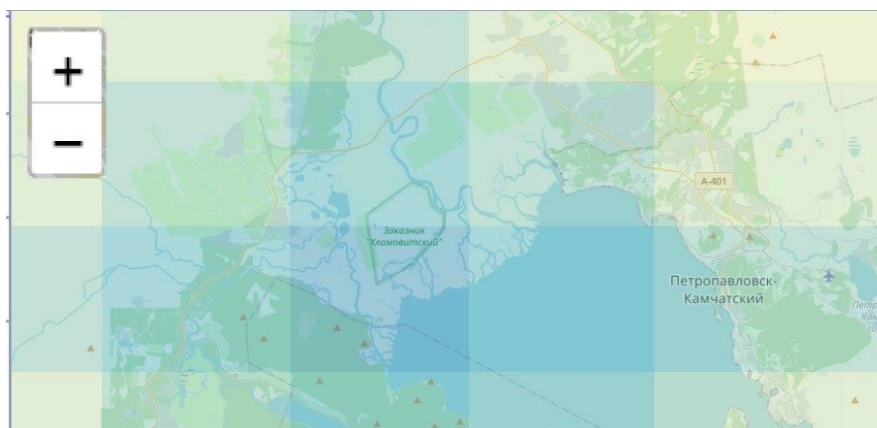


Рисунок 2-8: Пример тепловой карты.

Работа такого сервиса заключается в наложении поверх географической карты каких-либо численных данных. Сервис должен уметь по географическим координатам получать фрагмент карты, после чего поверх него отображать интерполированные данные из двумерного массива.

Для такой задачи уже не обязательно уметь на лету отрисовывать карты с различным масштабированием, т. к. вызвать этот сервис уже должен не клиент, а сервер при подготовке данных. Таким образом, сервер работает с одним большим изображением (21600 на 10800 пикселей), вырезает из него требуемый фрагмент, после чего накладывает на него маску.

2.2.6. Блок расчета движения в околоземном пространстве

Основное назначение модуля - проводить расчет движения тел в околоземном пространстве, при этом модуль может использовать различные алгоритмы расчета и различные физические приближения.

2.2.6.1. Состав модуля

Модуль движения в околоземном пространстве состоит из следующих библиотек:

- библиотека для работы с геометрическими объектами (GeometryCpp);
- библиотека для работы с системами координат (CoordinateSystemCpp);

- библиотека для работы с космическим объектом (BaseBallisticsUtils);
- библиотека для работы с солнечным давлением (SolarPressure);
- библиотека для работы с атмосферой (AtmosphereModel);
- библиотека для работы с атмосферным давлением (DragPressure);
- библиотека для работы с гравитационным притяжением (CelestialBodyGravity);
- библиотека для работы с релятивистскими коррекциями (RelativisticsCorrections);
- библиотека для работы с интерполяцией траекторий (TrajectoryInterpolator);
- библиотека численных методов (Integrator);
- библиотека для работы с управляющей силой (VectoredForce);
- библиотека для работы с баллистическими системами координат (BallisticsCoordinateConverter);
- библиотека для создания динамической библиотеки (PythonBinding);
- библиотека для Python-интерфейса (BallisticsPythonInterface).
-

Библиотека GeometryCpp представляет собой набор функций и методов для работы с геометрическими примитивами, такими как вектор, матрица, кватернион, сфера и эллипсоид.

Библиотека CoordinateSystemCpp обеспечивает функционал для перехода между систем координат ITRF, GCRS и TEME и функционал для перехода между системами времен TT, TAI, UTC, UT1, TCG, TCB, TDB и GPS.

Библиотека BaseBallisticsUtils представляет собой набор структур для хранения данных о космическом объекте — массы космического объекта, его эффективной площади для солнечного и атмосферного давлений и двигательной установки.

Библиотека SolarPressure обеспечивает функционал для определения функции тени и прямого солнечного давления, оказываемого на космический объект.

Библиотека `AtmosphereModel` предназначен для определения плотности атмосферы по моделям ГОСТ и NRLMSISE00.

Библиотека `CelestialBodyGravity` обеспечивает функционал для вычисления гравитационного тяготения Земли и других небесных тел солнечной системы.

Библиотека `RelativisticsCorrections` представляет собой набор функций для вычисления релятивистских поправок к уравнению движения космического аппарата.

Библиотека `TrajectoryInterpolator` предназначен для создания интерполянтов траекторий космического объекта.

Библиотека `Integrator` содержит в себе библиотеку численных методов для решения задач Коши для систем обыкновенных дифференциальных уравнений.

Библиотека `VectoredForce` обеспечивает функционал для задания работы двигательной установки космического объекта.

Библиотека `BallisticsCoordinateConverter` обеспечивает конвертацию между различными системами координат, используемых в небесной механике.

Библиотека `PythonBinding` предназначена для формирования динамической библиотеки, содержащей функционал всех вышеперечисленных библиотек.

Библиотека `BallisticsPythonInterface` представляет собой интерфейс для работы с динамической библиотекой.

2.2.6.2. Связи между библиотеками

Библиотека `GeometryCpp` использует стороннюю библиотеку `Eigen`.

Библиотека `CoordinateSystemCpp` использует библиотеку `GeometryCpp`.

Библиотека `BaseBallisticsUtils` использует библиотеки `GeometryCpp` и `CoordinateSystemCpp`.

Библиотека `SolarPressure` использует библиотеки `BaseBallisticsUtils`, `GeometryCpp`, `CoordinateSystemCpp`.

Библиотека `AtmosphereModel` использует библиотеки `BaseBallisticsUtils`, `GeometryCpp`, `CoordinateSystemCpp`.

Библиотека `DragPressure` использует библиотеки `AtmosphereModel`, `BaseBallisticsUtils`, `GeometryCpp`, `CoordinateSystemCpp`.

Библиотека `CelestialBodyGravity` использует библиотеки `BaseBallisticsUtils`, `GeometryCpp`, `CoordinateSystemCpp`.

Библиотека `RelativisticsCorrections` использует библиотеки `BaseBallisticsUtils`, `GeometryCpp`, `CoordinateSystemCpp`.

Библиотека `TrajectoryInterpolator` использует библиотеки `BaseBallisticsUtils`, `GeometryCpp`.

Библиотека `Integrator` использует библиотеки `BaseBallisticsUtils`, `GeometryCpp`, `CoordinateSystemCpp`.

Библиотека `VectoredForce` использует библиотеки `BallisticsCoordinateConverter`, `BaseBallisticsUtils`, `GeometryCpp`, `CoordinateSystemCpp`.

Библиотека `BallisticsCoordinateConverter` использует `GeometryCpp`, `CoordinateSystemCpp`.

Библиотека `PythonBinding` использует все вышеперечисленные библиотеки.

Библиотека `BallisticsPythonInterface` использует все вышеперечисленные библиотеки.

2.2.6.3. Входные данные

Входными данными для модуля движения в околоземном пространстве являются:

– начальное положение космического объекта — положение и скорость в системе GCRS;

– начальное время в шкале TT, в которое заданы начальное положение и скорость

– сухую массу космического объекта;

– массу топлива космического объекта;

– эффективную площадь для аэродинамического сопротивления космического объекта;

– эффективную площадь для солнечного давления космического объекта;

– удельный импульс и расход двигательной установки (при наличии);

– времена в шкале TT, в которые необходимо предоставить результат интегрирования;

- минимальный шаг численного интегрирования;
- максимальный шаг численного интегрирования;
- начальный шаг численного интегрирования;
- максимальная относительная ошибка численного интегрирования;
- режим работы двигательной установки (при наличии).

2.2.6.4. Выходные данные

Выходными данными модуля движения в околоземном пространстве являются:

- положения космического объекта в заданные моменты времени;
- скорости космического объекта в заданные моменты времени;
- масса топлива космического объекта в эти же моменты времени.

2.2.6.5. Возможные исключения при работе

Для обработки ошибок в течение работы модуля движения в околоземном пространстве используются исключения.

– Исключение `GeometryException` используется в случае, когда невозможна интерполяция значения, когда невозможно считывание или запись в файл матрицы или вектора, когда невозможно произвести процедуры с геометрическими примитивами: сферой, эллипсоидом, прямой и плоскостью;

– Исключение `CoordinateSystemException` используется в случае, когда невозможно считать файл с эфемеридами, когда невозможно считать файл с параметрами вращения Земли, когда производятся арифметические операции с временами в разных шкалах, когда невозможно сконвертировать одну шкалу времени к другой или когда невозможно подсчитать матрицу перехода между системами координат GCRS, ITRS или TEME (запрашиваемое время значительно превосходит временной интервал информации, указанной в файле с данными о вращении Земли);

- Исключение `SolarPressureException` используется в случае, если координаты космического объекта поданы таким образом, что находятся под поверхностью какого-либо из небесных тел;
- Исключение `AtmosphereModelException` используется в случае, если не удалось считать параметры атмосферы из файла, если параметры атмосферы запрашиваются в момент времени, не входящий в временной интервал данных в файле атмосферы;
- Исключение `BallisticsCoordinateConverterException` используется в случае, когда при конвертации радиус-вектора и скорости в Кеплеровы параметры метод итерационный метод решения уравнения Кеплера не сходится к решению;
- Исключение `CelestialBodyGravityException` используется в случае, когда координаты точки, в которой необходимо посчитать притяжение Земли, находится под Земной корой;
- Исключение `BallisticsPythonInterfaceException` используется в случае, когда указаны некорректные параметры физических моделей при создании баллистического пропагатора.

2.2.7. Модуль дистанционного зондирования земли

Основное назначение модуля – рассчитывать выполнение целевой задачи КА дистанционного зондирования Земли.

2.2.7.1. Состав модуля дистанционного зондирования земли

Модуль дистанционного зондирования земли состоит из следующих библиотек:

- библиотека для работы с геометрическими объектами (`GeometryCpp`);
- библиотека для работы с системами координат (`CoordinateSystemCpp`);
- библиотека для расчета дистанционного зондирования (`RemoteSensingCpp`);
- библиотека для создания динамической библиотеки (`RemoteSensingPythonBinding`);
- библиотека для Python-интерфейса (`RemoteSensingPythonInterface`).

Библиотека `GeometryCpp` представляет собой набор функций и методов для работы с геометрическими примитивами, такими как вектор, матрица, кватернион, сфера и эллипсоид.

Библиотека `CoordinateSystemCpp` обеспечивает функционал для перехода между систем координат ITRF, GCRS и TEME и функционал для перехода между системами времен TT, TAI, UTC, UT1, TCG, TCB, TDB и GPS.

Библиотека `RemoteSensingCpp` предназначена для расчета зон покрытия при дистанционном зондировании Земли, вычисления ключевых показателей эффективности при решении задачи зондирования, а также оценки загруженности космического аппарата.

Библиотека `RemoteSensingPythonBinding` предназначена для формирования динамической библиотеки, содержащей функционал всех вышеперечисленных библиотек.

Библиотека `RemoteSensingPythonInterface` представляет собой интерфейс для работы с динамической библиотекой.

2.2.7.2. Связи между библиотеками

Библиотека `GeometryCpp` использует стороннюю библиотеку `Eigen`.

Библиотека `CoordinateSystemCpp` использует библиотеку `GeometryCpp`.

Библиотека `RemoteSensingCpp` использует библиотеки `GeometryCpp` и `CoordinateSystemCpp`.

Библиотека `RemoteSensingPythonBinding` использует библиотеки `GeometryCpp`, `CoordinateSystemCpp` и `RemoteSensingCpp`.

Библиотека `RemoteSensingPythonInterface` использует все перечисленные библиотеки.

2.2.7.3. Входные данные

Входными данными для модуля дистанционного зондирования Земли являются:

– Положения космических аппаратов дистанционного зондирования в системе GCRS;

- Скорости космических аппаратов дистанционного зондирования в системе GCRS;
- Кватернионы вращения Земли на весь интервал моделирования;
- Положения Солнца на весь интервал моделирования;
- Средства зондирования для каждого космического аппарата (оптическая камера, ИК – камера, радиолокатор);
- Предельные углы крена для каждого средства зондирования;
- Углы захвата для каждого средства зондирования;
- Линейное разрешение в зачетных условиях для каждого средства зондирования;
- Описание зачетных условий для каждого средства зондирования;
- Список целевых областей;
- Граница каждой целевой области (в виде многоугольника на карте);
- Предельные углы места для каждой целевой области;
- Предельные углы Солнца для каждой целевой области;
- Предельные минимальные линейные разрешения для каждой целевой области;
- Список целевых точек наблюдения;
- Предельные углы места для каждой целевой точки;
- Предельные углы Солнца для каждой целевой точки;
- Предельные минимальные линейные разрешения для каждой целевой точки.
- Ориентация средств зондирования космического аппарата;
- При отсутствии ориентации средств зондирования - параметры оптимизации средств зондирования.

2.2.7.4. Выходные данные

Выходными данными модуля движения в околоземном пространстве являются:

– КРІ по каждой целевой области: максимальная, минимальная и средняя периодичности наблюдения, максимальный, минимальный и средний угол места наблюдения, максимальный, минимальный и средний угол солнца, максимальное, минимальное и среднее линейное разрешение, процент покрытия области в зависимости от времени;

– КРІ по каждой целевой точке: максимальная, минимальная и средняя периодичности наблюдения, максимальный, минимальный и средний угол места наблюдения, максимальный, минимальный и средний угол солнца, максимальное, минимальное и среднее линейное разрешение, количество осмотров;

– Расписание работы каждого средства наблюдения на каждом космическом аппарате;

– В случае проведения оптимизации по ориентации средства зондирования возвращается наилучшая ориентация средства зондирования.

2.2.7.5. Возможные исключения при работе

Для обработки ошибок в течение работы модуля дистанционного зондирования используются исключения.

– Исключение `GeometryException` используется в случае, когда невозможна интерполяция значения, когда невозможно считывание или запись в файл матрицы или вектора, когда невозможно произвести процедуры с геометрическими примитивами: сферой, эллипсоидом, прямой и плоскостью;

– Исключение `CoordinateSystemException` используется в случае, когда невозможно считать файл с эфемеридами, когда невозможно считать файл с параметрами вращения Земли, когда производятся арифметические операции с временами в разных шкалах, когда невозможно сконвертировать одну шкалу времени к другой или когда невозможно подсчитать матрицу перехода между системами координат GCRS, ITRS или TEME (запрашиваемое время значительно превосходит временной интервал информации, указанной в файле с данными о вращении Земли);

– Исключение `RemoteSensingException` используется в случае, когда граница целевой области наблюдения состоит из 2 и менее точек, когда бинарный файл с

результатами не корректен, когда неверно указаны углы захвата оптического средства.

2.2.8. Модуль мониторинга космического пространства

Назначение модуля – рассчитывать решение целевой задачи для космических аппаратов, которые наблюдают за другими космическими объектами.

2.2.8.1. Состав

Модуль дистанционного зондирования земли состоит из следующих библиотек:

- библиотека для работы с геометрическими объектами (GeometryCpp);
- библиотека для работы с системами координат (CoordinateSystemCpp);
- библиотека для работы с солнечным давлением (SolarPressure);
- библиотека для расчета дистанционного зондирования (ControlSystemImmitation);
- библиотека для создания динамической библиотеки (ConstrolSystemImmitationPythonBinding);
- библиотека для Python-интерфейса (ControlSystemImmitationPythonInterface).

Библиотека GeometryCpp представляет собой набор функций и методов для работы с геометрическими примитивами, такими как вектор, матрица, кватернион, сфера и эллипсоид.

Библиотека CoordinateSystemCpp обеспечивает функционал для перехода между систем координат ITRF, GCRS и TEME и функционал для перехода между системами времен TT, TAI, UTC, UT1, TCG, TCB, TDB и GPS.

Библиотека SolarPressure обеспечивает функционал для определения функции тени и прямого солнечного давления, оказываемого на космический объект.

Библиотека ControlSystemImmitation предназначена для проведения моделирования мониторинга космического пространства, расчета видимости тел

в космическом пространстве спутниками наблюдателями и наземными станциями наблюдения.

Библиотека RemoteSensingPythonBinding предназначена для формирования динамической библиотеки, содержащей функционал всех вышеперечисленных библиотек.

Библиотека RemoteSensingPythonInterface представляет собой интерфейс для работы с динамической библиотекой.

2.2.8.2. Связи между библиотеками

Библиотека GeometryCpp использует стороннюю библиотеку Eigen.

Библиотека CoordinateSystemCpp использует библиотеку GeometryCpp.

Библиотека SolarPressure использует библиотеки GeometryCpp, CoordinateSystemCpp.

Библиотека ControlSystemImmitation использует библиотеки GeometryCpp, CoordinateSystemCpp и SolarPressure.

Библиотека RemoteSensingPythonBinding использует библиотеки GeometryCpp, CoordinateSystemCpp, ControlSystemImmitation и SolarPressure.

Библиотека RemoteSensingPythonInterface использует все перечисленные библиотеки.

2.2.8.3. Входные данные

Входными данными для модуля дистанционного зондирования Земли являются:

- Положения космических аппаратов наблюдателей в системе GCRS;
- Скорости космических аппаратов наблюдателей GCRS;
- Расположение наземных станций наблюдения в системе ITRS;
- Кватернион вращения Земли на весь интервал моделирования;
- Положение Солнца на весь интервал моделирования;
- Положение Луны на весь интервал моделирования;
- Положения целей наблюдения в системе GCRS;
- Скорости целей наблюдения в системе GCRS;

– Средства мониторинга на каждый аппарат-наблюдатель (оптическое средство или радиолокатор);

– Характеристики средств мониторинга. Для оптического средства – поля видимости и максимальная звездная величина для наблюдения. Для радиолокационного средства – поля зрения и минимальная эффективная площадь рассеяния тела, наблюдаемого с расстояния 1000 км;

– Список ключевых индексов полезности, которые необходимо вычислить;

– Характеристики целей наблюдения – эффективная площадь рассеяния, характерный линейный размер;

– Режим расчета – потенциальная или реальная видимость;

– В случае реальной видимости – программа наблюдения.

2.2.8.4. Выходные данные

Выходными данными модуля движения в околоземном пространстве являются:

– КРІ по каждой цели наблюдения: максимальная, минимальная и средняя звездная величина наблюдения, количество наблюдений, максимальное, минимальное и среднее время ненаблюдения, список аппаратов-наблюдателей и наземных станций наблюдения, осматривавших цель за интервал моделирования;

– КРІ по каждому аппарату наблюдателю и наземной станции: минимальная и средняя звездная величина наблюдения, количество наблюдений, список наблюдаемых целей, количество наблюдаемых целей в зависимости от времени, двумерные карты видимости целей.

2.2.8.5. Возможные исключения при работе

Для обработки ошибок в течение работы модуля дистанционного зондирования используются исключения.

– Исключение `GeometryException` используется в случае, когда невозможна интерполяция значения, когда невозможно считывание или запись в файл матрицы или вектора, когда невозможно произвести процедуры с геометрическими примитивами: сферой, эллипсоидом, прямой и плоскостью;

– Исключение `CoordinateSystemException` используется в случае, когда невозможно считать файл с эфемеридами, когда невозможно считать файл с параметрами вращения Земли, когда производятся арифметические операции с временами в разных шкалах, когда невозможно сконвертировать одну шкалу времени к другой или когда невозможно подсчитать матрицу перехода между системами координат GCRS, ITRS или TEME (запрашиваемое время значительно превосходит временной интервал информации, указанной в файле с данными о вращении Земли);

– Исключение `SolarPressureException` используется в случае, если координаты космического объекта поданы таким образом, что находятся под поверхностью какого-либо из небесных тел;

– Исключение `ConstrolSystemImmitationException` используется в случае, когда для расчета с потенциальной видимостью используются объекты для расчета с реальной видимостью и наоборот, когда размеры массивов входных данных о положении Луны, Солнца и вращении Земли не согласованы (разные длины массивов), когда для проверки геометрической видимости используется неправильные геометрические примитивы

2.2.9. Блок расчета маршрутизации данных

Задачей модуля является решение задачи маршрутизации целевых сообщений среди участников сети: космическими аппаратами (КА) и наземными станциями. Решением задачи является список маршрутов для каждого сообщения с подробной информацией о передачах.

2.2.9.1. Входные данные.

Для решения задачи маршрутизации необходима информация про следующие сущности: соединения, по которым осуществляется связь; целевые сообщения (пакеты); узлы связи; алгоритм маршрутизации и его настройки; возможные комбинации целевых узлов.

Соединения.

Под соединением подразумевается канал связи, через который может быть передана информация.

В модуль маршрутизации информация о соединениях приходит от модуля Available Communication Graph. О каждом соединении известна следующая информация:

- Время начала соединения
- Время конца соединения
- Задержка соединения
- Отправитель (информация про узел и антенну или виртуальный индекс антенны-отправителя)
- Приемник (информация про узел и антенну или виртуальный индекс антенны-приемника)
- Стоимость передачи по соединению
- Скорость передачи по соединению

Пакеты.

Пакет представляет из себя объем информации, который генерируется на одном из узлов сети и должен быть доставлен на другой (т.н. целевой узел). Стоит отметить, что целевой узел может быть не один (т.е. допустимо доставить пакет в любой из указанных узлов).

В модуль маршрутизации информация о соединениях подается извне. О каждом пакете известна следующая информация:

- Узел, на котором пакет был сгенерирован
- Время генерации
- Приоритет пакета
- Целевой узел
- Размер пакета

Узлы связи.

Узел связи представляет из себя упрощенную информацию об участниках связи. Для решения задачи маршрутизации требуется следующая информация:

- Идентификатор узла
- Объем памяти
- Возможность ретрансляции

Алгоритм маршрутизации и его настройки.

В модуле маршрутизации реализованы обертки над двумя видами алгоритмов: динамическими и статическими. Для статических алгоритмов, вся информация о соединениях и пакетах обрабатывается сразу, т.е. заранее известно, какие аппараты смогут связываться между собой.

Динамическим алгоритмам данные о возникновении/разрыве соединений и генерации пакетов подаются по мере их появления. Для принятия решений об отправке сообщений такие алгоритмы используют т.н. служебную информацию, которая должна передаваться между узлами. Состав этой информации зависит от алгоритма.

В качестве настроек динамическим алгоритмам подаются следующие параметры:

– Флаг мгновенного распространения служебной информации. Данная настройка позволяет существенно повысить скорость выполнения расчета, а при малом размере пакета служебной информации точность расчета снизится незначительно.

– Размер пакета с служебной информацией.

– Частота генерации служебных пакетов.

Возможные комбинации целевых узлов.

Как было указано выше, для некоторых узлов допустимо несколько целевых узлов. Для удобства передачи информации о нескольких целевых узлах для пакета было выбрано следующее решение: на этапе инициализации расчета должен быть подан словарь возможных комбинаций. Ключом этого словаря является целое отрицательное число, а значением – массив идентификаторов узлов.

Динамические же алгоритмы принимают решения о маршрутизации исходя из информации, имеющейся в конкретный момент времени. Поэтому процесс решения задачи маршрутизации с применением подобного рода алгоритмов представляет собой пошаговое моделирование всей системы.

2.2.9.2. Решение задачи маршрутизации

В случае использования статических алгоритмов маршрутизации (CGR и RDTN) в модуле происходит преобразование данных, получаемых извне. Никаких изменений в данных при этом не происходит, вся информация сразу подается алгоритму.

Динамические же алгоритмы принимают решения о маршрутизации исходя из информации, имеющейся в конкретный момент времени. Поэтому процесс решения задачи маршрутизации с применением подобного рода алгоритмов представляет собой пошаговое моделирование всей системы.

Процесс решения состоит из следующих шагов:

1) Солвер (система-обертка для некоего алгоритма маршрутизации) получает на вход полную информацию о соединениях, пакетах, узлах связи. Важный момент: информация хранится в солвере, но не доступна алгоритму, принимающему решение.

2) Происходит инициализация общих требуемых служебных переменных: времена начала и конца расчета, массив размеров доступной памяти, массив информации о пересылках, массив текущих пересылок.

3) Происходит инициализация специальных для алгоритма требуемых служебных переменных.

4) После инициализации всех требуемых для расчета переменных запускается цикл «по времени».

Регистрация новых пакетов. Подробнее процесс отслеживания положений пакетов будет описан ниже.

Проверка возникновения новых (т.е. таких, которые не были активны на предыдущем шаге по времени, но доступны на текущем) соединений. Передача информации о таких соединениях в алгоритм.

Проверка возникновения закрытых (т.е. таких, которые были активны на предыдущем шаге по времени, но не доступны на текущем) соединений. Передача информации о таких соединениях в алгоритм.

Проверка выполнения условия генерации служебных пакетов. Если условие выполняется, то происходит их генерация и рассылка.

Передача в алгоритм информации о новых пакетах. После решения алгоритма происходит рассылка отправка пакета по указанному алгоритмом соединению.

Изменение значения «текущего времени» на величину шага по времени.

Отслеживание и перемещение информации о пакетах организовано следующим образом.

1. После решения алгоритма на отправку некоего пакета по соединению, в соответствующий элемент массива текущих пересылок добавляется массив из трех элементов: номер пакета, время предполагаемого начала передачи, время предполагаемого завершения передачи. Время завершения передачи больше времени начала на величину отношения размера пакета к скорости передачи соединения.

Если на момент добавления в ячейке нет информации о других пакетах, то время предполагаемого начала передачи совпадает со значением текущего шага по времени, а время предполагаемого завершения передачи.

Если в ячейке есть информация о других пакетах, то время предполагаемого начала передачи совпадает со временем предполагаемого завершения последнего добавленного пакета.

2. Как было упомянуто выше, в начале каждого шага цикла по времени происходит регистрация новых пакетов. Новым, в данном случае, является пакет, который либо был сгенерирован на предыдущем шаге по времени, либо был прислан на один узел с другого. Стоит отметить, что при генерации пакета происходит учет текущей свободной памяти узла. Пакет будет «передан» узлу только в случае, если объем текущей свободной памяти больше размера пакета. В противном случае, пакет считается потерянным.

Контроль пересылок происходит путем прохода по массиву текущих пересылок следующим образом:

Для элемента массива определяется время окончания. Это либо значение текущего времени, либо время завершения соединения (выбирается наименьшее из этих двух значений)

Для каждой записи о пакете в ячейке выполняется сравнение значения времени окончания и предполагаемого завершения передачи.

Если время окончания больше, то пакет считается переданным. На этом этапе происходит увеличение объема свободной памяти на узле-отправителе. Если объем свободной памяти узла-приемника меньше размера переданного пакета, то передача не засчитывается и пакет считается потерянным.

В массив информации о пересылках записывается отметка о выполнении передачи с указанием начального и конечного узла, а также временах начала и конца передачи. Если при этом конечный узел является целевым, то пакет считается доставленным и в дальнейших пересылках не участвует.

После этих действий, запись о передаче пакета удаляется.

Если время окончания совпадает со временем завершения соединения, то все записи о пакетах в ячейке, оставшиеся после проверок предыдущего пункта, удаляются.

2.2.9.3. Выходные данные

После обработки входных данных модуль для каждого пакета выдает его маршрут. Маршрут представляет из себя информацию про:

– Сеансы связи. По каждому сеансу известно: идентификатор спутника-отправителя, идентификатор антенны-отправителя, идентификатор спутника-приемника, идентификатор антенны-приемника, время начала сеанса, время завершения сеанса

– Время доставки пакета по маршруту

– Стоимость доставки пакета по маршруту

– Таким образом среди выходных данных появляется сущность «маршрут пакета».

2.2.10. Модуль расчета интегральных характеристик связи

Модуль представляет собой набор функций, анализирующих информацию о пакетах, соединениях, узлах, возможных комбинациях целевых узлов, маршрутах пакетов (эти данные получаются от модуля маршрутизации).

2.2.10.1. Входные данные

Модуль принимает на вход входные и выходными данными блока расчета маршрутизации 2.2.9. Модуль может решать множество задач по определению различных характеристик задачи маршрутизации, для удобства восприятия, содержимое модуля описано в таблице 2-1.

Таблица 2-1 – Общий список задач, которые может определять модуль расчета интегральных характеристик

Задача	Входные данные	Выходные данные
Метод вычисления количества переданной и принятой информации за период.	<ul style="list-style-type: none">• Идентификаторы узлов• Точки времени• Период• Маршруты сообщений	Суммарное количество информации, переданной через узлы с заданными идентификаторами за период
Метод вычисления количества сеансов за период	<ul style="list-style-type: none">• Идентификаторы узлов• Точки времени• Период• Маршруты сообщений	Количество сеансов для узлов с заданными идентификаторами. Прим. Если на вход подана группировка (т.е. несколько узлов, то пересылки внутри

		группировки не учитываются)
Метод вычисления времени продолжительности и сеанса связи	<ul style="list-style-type: none"> •Идентификаторы узлов •Маршруты сообщений 	<p>Статистика времен продолжительности сеансов для указанных узлов (мин, макс, среднее, дисперсия)</p> <p>Прим. Если на вход подана группировка (т.е. несколько узлов, то пересылки внутри группировки не учитываются)</p>
Метод вычисления времени ожидания сеанса связи	<ul style="list-style-type: none"> •Идентификаторы узлов •Маршруты сообщений 	<p>Статистика времен продолжительности сеансов для указанных узлов (мин, макс, среднее, дисперсия)</p> <p>Прим. Если на вход подана группировка (т.е. несколько узлов, то пересылки внутри группировки не учитываются)</p>

<p>Метод вычисления времен продолжительност и возможной связи</p>	<ul style="list-style-type: none"> •Соединения 	<p>Статистика времен продолжительности возможной связи (мин, макс, среднее, дисперсия).</p> <p>Прим. Возможная связь – между узлами есть соединение.</p>
<p>Метод вычисления отношения объема потерянной информации к объему сгенерированной</p>	<ul style="list-style-type: none"> •Маршруты сообщений 	<p>Отношение объема потерянной информации к объему сгенерированной.</p> <p>Прим. Пакет считается потерянным, если он был потерян при передаче/генерации или не дошел до адресата.</p>
<p>Метод вычисления объема сгенерированной информации</p>	<ul style="list-style-type: none"> •Пакеты •Идентификатор ы узлов 	<p>Объем сгенерированной информации на указанных узлах</p>
<p>Метод вычисления пропускной способности</p>	<ul style="list-style-type: none"> •Идентификатор ы узлов •Маршруты •Пакеты 	<p>Статистика о пропускной способности узла или группы узлов (мин, макс, среднее, дисперсия)</p>

		Под пропускной способностью подразумевается отношение суммарного объема переданной информации через узел ко всему времени расчета
Метод вычисления пропускной способности от времени	<ul style="list-style-type: none"> • Идентификаторы узлов • Маршруты • Пакеты • Соединения • Точки времени 	Скорость передачи данных через узел или набор узлов в указанные точки времени.
Метод вычисления кол-ва одновременных сеансов связи от времени	<ul style="list-style-type: none"> • Маршруты • Точки времени 	Количество одновременных сеансов связи в указанные точки времени
Метод вычисления объема принятой и объема переданной информации	<ul style="list-style-type: none"> • Маршруты • Точки времени • Идентификаторы узлов 	Суммарный принятый/ переданный к указанным точкам времени объем информации на узлах
Метод вычисления заполненности памяти	<ul style="list-style-type: none"> • Идентификаторы узлов • Маршруты • Точки времени 	Заполненность (абсолютная и относительная) памяти

		узла в указанные точки времени.
Метод вычисления загрузки узла	<ul style="list-style-type: none"> • Идентификатор узла • Маршруты • Пакеты • Соединения • Точки времени 	<p>Загруженность узла в указанные моменты времени</p> <p>Прим.</p> <p>Загруженность – это отношение текущей скорости передачи данных к максимальной возможной.</p>
Метод вычисления количества КА в зоне расчетной видимости	<ul style="list-style-type: none"> • Соединения • Точки времени 	<p>Количество аппаратов, находящихся в зоне видимости наземных станций для указанных точек времени.</p>
Метод вычисления среднего значения количества одновременных сеансов	<ul style="list-style-type: none"> • Точки времени 	<p>Среднеинтегральное значение одновременных сеансов связи</p>
Метод вычисления оперативности	<ul style="list-style-type: none"> • Пакеты • Маршруты 	<p>Статистика оперативности (мин, макс, среднее, дисперсия)</p> <p>Оперативность – разница времени приема пакета на</p>

		<p>целевом узле и времени генерации пакета.</p> <p>Прим. Потерянные пакеты не учитываются</p>
<p>Метод вычисления оперативности от времени</p>	<ul style="list-style-type: none"> • Пакеты • Маршруты • Точки времени 	<p>Статистика оперативности (мин, макс, среднее, дисперсия) для пакетов, которые были доставлены между указанными точками по времени</p>
<p>Метод вычисления оперативности доставки пакетов по регионам</p>	<ul style="list-style-type: none"> • Маршруты • Информация о принадлежности пакета к региону 	<p>Статистика оперативности (мин, макс, среднее, дисперсия) по регионам.</p>
<p>Метод вычисления срока выполнения заявки.</p>	<ul style="list-style-type: none"> • Маршруты • Информация о принадлежности пакета к заявке 	<p>Минимальное время доставки пакета по заявке</p>
<p>Метод вычисления циклограммы реализованных связей.</p>	<ul style="list-style-type: none"> • Маршруты 	<p>Циклограмма реализованных связей.</p>
<p>Метод вычисления циклограммы</p>	<ul style="list-style-type: none"> • Соединения 	<p>Циклограмма потенциальных связей.</p>

потенциальных связей.		
--------------------------	--	--

3. НАСТРОЙКА ПРОГРАММЫ

Демонстратор программного комплекса «Интеграл-Д» поставляется в виде набора docker образов, данных для первоначального заполнения различных баз данных, конфигурационных файлов (включая переменные окружения) и скриптов, которые позволяют развернуть серверную компоненту в рамках одной рабочей станции. При подъеме docker контейнеров связь между ними происходит путем сетевого взаимодействия. Возможны и другие варианты при корректном изменении переменных окружения среды (расположены в файле .env) и настроек сервера nginx (conf/nginx/default.conf). Дальнейшее описание приводится для случая развёртывания в рамках одного компьютера.

3.1. Первоначальная установка демонстратора программного комплекса «Интеграл-Д»

1. Скопировать в папку `integral_d` следующие файлы:
 1. `docker-compose.yml`
 - `load_images.sh`
 - `migrate.sh`
 - `setup_tiles.sh`
 - `db.tar.gz`
 - `integral_d.tar.gz`
 - `.env`
 - папку `conf`
 - папку `data`
 2. Перейти в папку `integral_d` (команда `cd`)
 3. Запустить скрипт загрузки образов ``load_images.sh``
 4. Перезапустить контейнеры ``sudo docker-compose up -d``. Выполнение команды требует прав администратора.
 5. Применить миграции: запустить ``sudo migrate.sh``

6. Проверить корректность файлов карт, запустив скрипт ``check_data_consistency.sh``. В случае несовпадения хэша или отсутствия файла будет сообщение об ошибке. В конце работы скрипт запишет "All files checked.". Обязательно выполнять при ошибках выполнения в `setup_tiles.sh`
7. Регенерация карт: запустить ``sudo setup_tiles.sh``. Длительная процедура. Может занимать несколько часов. В случае ошибок нужно проверить целостность файлов, представленных в папке `data`.
8. Создать пользователя с правами администратора в соответствии с разделом 3.5.
9. Провести проверку доступности сервера в соответствии с разделом 4. Если 80 порт занят, то рекомендуется выключить конфликтующее ПО и провести проверку. Если это невозможно, то данный пункт можно пропустить и переходить к пункту 10. Но это осложнит дальнейший поиск возможных ошибок.
10. Произвести настройку веб сервера в соответствии с разделом 3.4
11. Провести повторную проверку в соответствии с разделом 4.

3.2. Обновление версии демонстратора программного комплекса «Интеграл-Д»

1. Скопировать в папку `integral_d` следующие файлы:
 - `db.tar.gz`.
 - `integral_d.tar.gz`.
2. Скопировать файлы настроек. Если в файлы настроек вносились изменения, то их надо внести во вновь предоставленные файлы. Рекомендуется сохранять файлы настроек. Если данные файлы не изменились по сравнению с прошлой версией, то копировать их не надо.
 - `docker-compose.yml`. Для сохранения локальных настроек рекомендуется использование механизма `docker-compose.override.yml` <https://docs.docker.com/compose/extends/>

- .env
 - папку conf
3. Скопировать скрипты для запуска. Изменения возможны
 - load_images.sh
 - migrate.sh
 - setup_tiles.sh
 4. Перейти в папку `integral_d` (команда `cd`). Для обновления уже существующей установки имя папки должно совпадать с именем папки, в которой производился первоначальный вызов `docker-compose up -d`. Обходить данное требование можно с использованием переменной окружения `COMPOSE_PROJECT_NAME`. Тогда команда на запуск должна выглядеть следующим образом `sudo COMPOSE_PROJECT_NAME=integral_d docker-compose up -d`, если первоначальный запуск производился из папки `integral_d`.
 5. Запустить скрипт загрузки образов `sudo load_images.sh`
 6. Перезапустить контейнеры `sudo docker-compose up -d`
 7. Применить миграции: запустить `sudo migrate.sh`
 8. Регенерация карт: запустить `sudo setup_tiles.sh`. Данное действие требуется только если были обновлены карты.

3.3. Сборка сообщений об ошибках

Для сбора сообщений об ошибках предусмотрена интеграция с сервером Sentry. Для отправки сообщений необходимо установить значение переменной окружения `SENTRY_DSN` равной клиентскому ключу `DSN`, который можно узнать в настройках проекта на сервере сбора ошибок:

1. Перейти в раздел `Settings` (см. рисунок Рисунок 3-1 цифра 1).
2. Выбрать категорию `Projects` (см. рисунок Рисунок 3-1 цифра 2).
3. Выбрать интересующий проект. В данном примере `example_for_instruction` (см. рисунок Рисунок 3-1 цифра 3).
4. Открыть `Client Keys (DSN)` (см. рисунок 3-2 цифра 4).

5. Скопировать ссылку, представляющую ключ DSN (см. рисунок Рисунок 3-1 цифра 1)

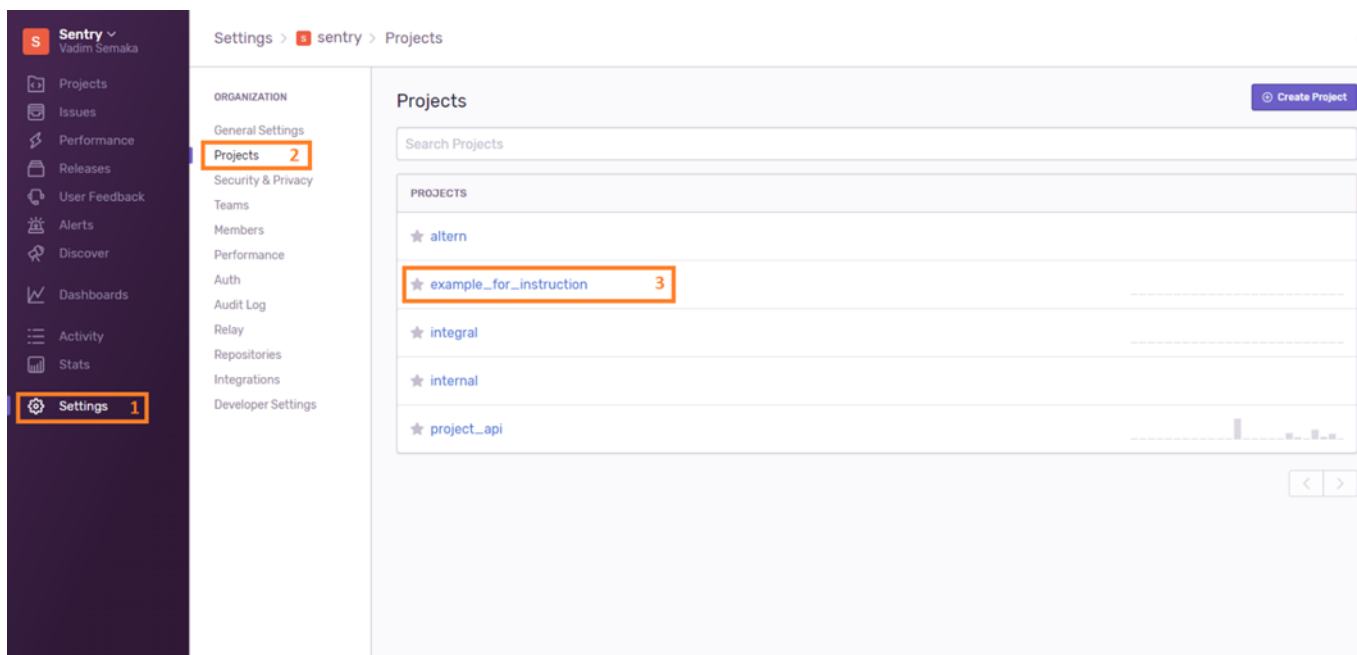


Рисунок 3-1 – Иллюстрация последовательности действий при сборе сообщений об ошибках

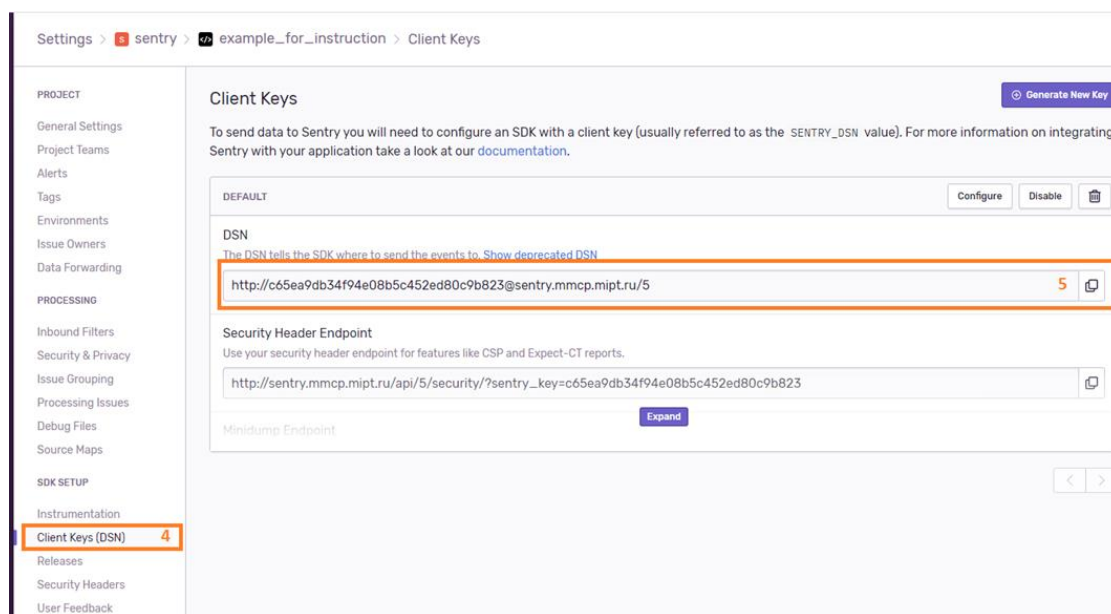


Рисунок 3-2 - Иллюстрация последовательности действий при сборе сообщений об ошибках

Для каждого из контейнеров установка данной переменной окружения происходит независимо от других. Для установки переменной окружения надо:

1. Открыть файл `docker-compose.yml` в любом текстовом редакторе (см. рисунок 3-3, цифра 1).

2. Найти интересующий контейнер (project_api, slow_worker, fast_worker, mapapi, frontend). В данном примере это project_api (см. рисунок 3-3, цифра 2).
3. Найти секцию environment (см. рисунок 3-3, цифра 3).
4. Внутри секции environment найти переменную окружения SENTRY_DSN и вставить скопированную ссылку (из пункта 5 предыдущей инструкции по получению DSN key)

```
project_api: 1
  image: registry.gitlab.mmcp.mipt.ru/integral/web_api/project_api:master
  depends_on:
    - mongo
    - postgres
    - redis
  networks:
    - backend
    - frontend
  environment: 2
    SENTRY_DSN: http://c65ea9db34f94e08b5c452ed80c9b823@sentry.mmcp.mipt.ru/5 3
    BACKEND_CORS_ORIGINS:
    BACKEND_CORS_ORIGIN_REGEX:
  env_file: .env
  environment:
    PYTHONPATH: /app
fast_worker:
  image: registry.gitlab.mmcp.mipt.ru/integral/web_api/project_api:master
  depends_on:
    - mongo
    - postgres
    - redis
  networks:
    - backend
  environment:
    PYTHONPATH: /app
  command: celery -A integral_d.core_api.main_celery.app worker -n fast_worker@%h --loglevel=info -Q fast
  environment:
    SENTRY_DSN:
    BACKEND_CORS_ORIGINS:
    BACKEND_CORS_ORIGIN_REGEX:
  env_file: .env
slow_worker:
  image: registry.gitlab.mmcp.mipt.ru/integral/web_api/project_api:master
  depends_on:
    - mongo
    - postgres
    - redis
  networks:
    - backend
  environment:
    PYTHONPATH: /app
  command: celery -A integral_d.core_api.main_celery.app worker -n slow_worker@%h --loglevel=info -Q slow
  environment:
    SENTRY_DSN:
```

Рисунок 3-3 – иллюстрация к установке переменной окружения

3.4. Изменение порта веб-сервера. Настройка SSL.

По умолчанию предусмотрена работа веб-сервера на 80 порту и приёма соединений по незащищенному протоколу http. При наличии доступа к серверу из сети интернет строго необходимо настроить доступ по протоколу https в соответствии с https://nginx.org/ru/docs/http/configuring_https_servers.html. Данная настройка не производится по умолчанию ввиду того, что каждая организация

владеет своей парой сертификата и закрытого ключа. Это конфиденциальные данные, передача которых ведет к их компрометации. В случае наличия доступа только из локальной сети или VPN сети использование протокола http также несёт некоторые риски безопасности, поэтому рекомендуется использовать протокол https.

При размещении сертификата и ключа в какой-либо подпапке `conf/nginx` настройку SSL следует производить исходя из того, что папка `conf/nginx` будет примонтирована в `/etc/nginx/conf.d/`. При размещении сертификата и ключа в любой другой папке следует её примонтировать к контейнеру разместив в файле `docker-compose.yml` в разделе volumes контейнера frontend строку аналогичную строке

`- ./conf/nginx:/etc/nginx/conf.d`, где вместо `./conf/nginx` следует указать путь до папки с сертификатами на хостовой машине, а `/etc/nginx/conf.d`, соответствующий путь в контейнере куда она будет примонтирована. Эта строка размещается в дополнение к строке `- ./conf/nginx:/etc/nginx/conf.d`. Важно соблюсти отступы. Для файла ключа рекомендуются установить права доступа 600 (chmod) и владельцем root(chown).

Если была произведена перенастройка протокола на https или 80 порт занят другим приложением, то следует изменить настройки проброса портов между контейнером frontend и хост машиной. Для этого надо в настройках контейнера frontend в секции ports изменить используемые порты. На рисунке 3-4 проиллюстрирована ситуация, когда все входящие соединения на 9090 порт хост машины перенаправляются на 80 порт контейнера frontend. В случае использования https следует перенаправлять входящие соединения, приходящие на 443 порт (заменить 9090 на 443). Второе число (80 в примере) должно быть согласовано с настройками nginx, заданными в файле `./conf/nginx/default.conf`, см. 3-5.

```

frontend:
  image: registry.gitlab.mmcp.mipt.ru/integral/alternative_ui/reactwebinterface:latest
  networks:
    - frontend
    - web
  ports:
    - 9090:80
  restart: always
  volumes:
    - ./conf/nginx/:/etc/nginx/conf.d/

```

Рисунок 3-4 – Перенаправление порта с хост машины в контейнер

```

server {
  listen 80;

  location / {
    root /usr/share/nginx/html;
    index index.html index.htm;
    try_files $uri $uri/ /index.html;
  }
}

```

Рисунок 3-5 – Настройка прослушиваемого порта сервера nginx

3.5. Создание пользователя с правами админа

Для создания пользователя с правами администратора необходимо выполнить команду `docker-compose exec project_api integral_admin_user add`. После этого Вам в интерактивном режиме (см. рисунок 3-6) будет предложено заполнить учетную информацию. При вводе first name, middle name, last name следует использовать только символы русского и латинского алфавита. Спецсимволы и пробелы не допускаются (пользователь будет создан, но не удастся авторизоваться).

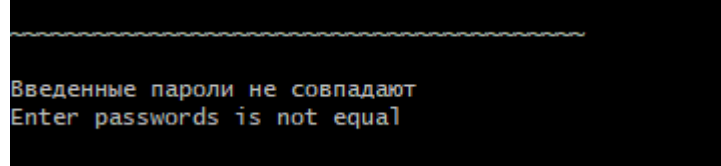
```

Enter admin email: semaka.viu@mipt.ru
Enter admin first name: admin
Enter admin middle name: admin
Enter admin last name: admin
Введите пароль администратора(Enter admin password):
Подтвердите пароль администратора(Confirm admin password):
Пользователь успешно создан(User creating is success)

```

Рисунок 3-6 – иллюстрация задания пароля супер пользователя

Если ввести несовпадающие пароли, то будет выдано соответствующее предупреждение, см. рисунок 3-7.



```
Введенные пароли не совпадают  
Enter passwords is not equal
```

Рисунок 3-7 – предупреждение при несовпадающих паролях

3.6. Одновременная работа разных версий демонстратора программного комплекса «Интеграл-Д»

Одновременная работа разных версий на одном сервере не предусмотрена. Скрипты, идущие в комплекте, настроены на запуск docker контейнеров следующих образов «»

3.7. Откат версии демонстратора программного комплекса «Интеграл-Д»

При применении скрипта migrate.sh происходит необратимое изменение баз данных. При замене docker образов на предыдущие версии работа с обновленной базой данных может оказаться невозможной или некорректной. Данный сценарий использования не рекомендуется и может приводить к ошибкам в расчетах.

4. ПРОВЕРКА ПРОГРАММЫ

В данном разделе предполагается, что установка была произведена с параметрами по умолчанию. Если происходило переназначение портов и/или протоколов, то проверяемые адреса стоит скорректировать с учетом реальных настроек.

Рекомендуется проверку выполнять, открыв браузер в режиме инкогнито. В случае наличия ошибок в настройках браузер может кэшировать неправильные версии страниц. В случае использования режима инкогнито (приватного режима) достаточно будет только закрыть и открыть браузер заново для сброса кэша.

В первый раз рекомендуется проходить пункты проверки 4.1 – 4.8 последовательно. В последующем можно выполнять сразу пункты 4.6 и 4.8, а в случае возникновения каких-либо проблем исполнения приступать к более детальному анализу.

4.1. Проверка доступности контейнера project_api

Откройте в браузере адрес <http://127.0.0.1/api/>. Если все настроено верно, то Вы увидите надпись "Hello Core Api", см. рисунок 4-1.

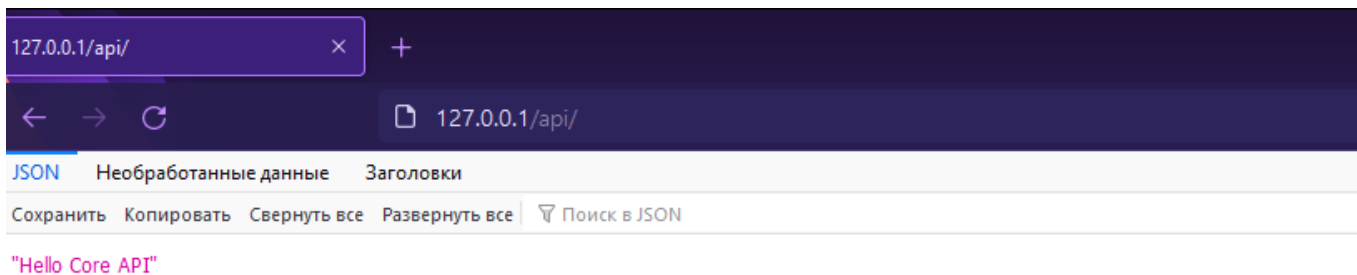


Рисунок 4-1 – Приветственная надпись, если все было сделано правильно

Если вместо этого Вы получаете 502 ошибку (см. рисунок 4-2), то возможно контейнер ещё не запустился и стоит подождать. Если это продолжается дольше 5 минут, то следует перейти к разделу анализ логов project_api.

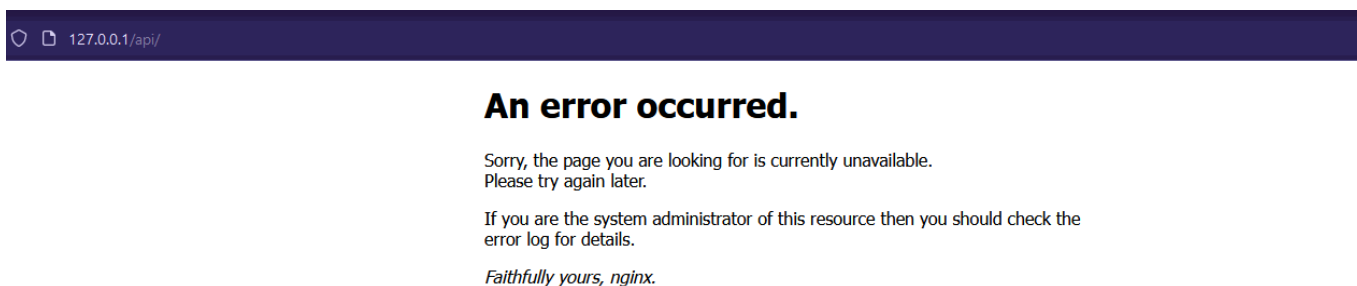


Рисунок 4-2 – иллюстрация 502 ошибки.

Если Вас перенаправляет на основную страницу интерфейса, то неверно настроено проксирование запросов для адресов вида <http://127.0.0.1/api/>.

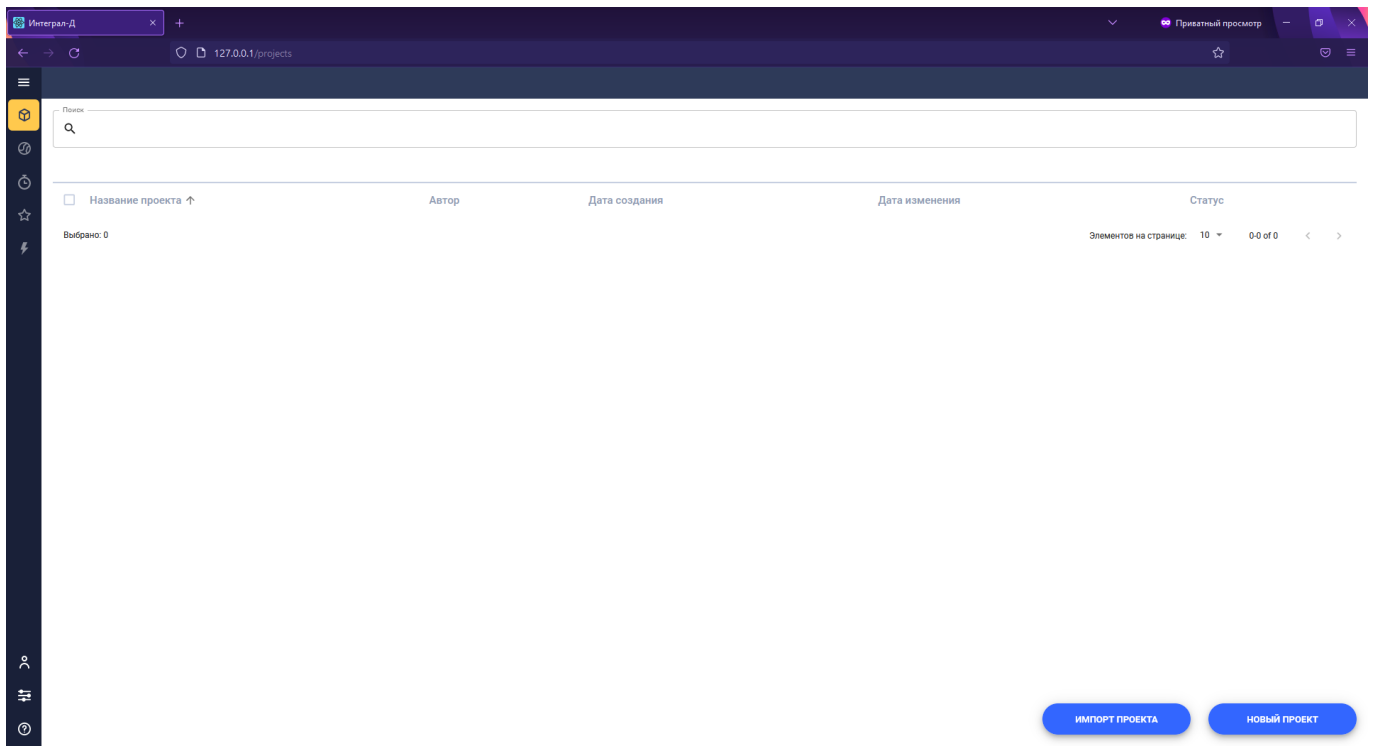


Рисунок 4-3 – стартовая страница интерфейса программного комплекса.

Проверьте, что в секции `server` файла `./conf/nginx/default.conf` действительно присутствует проксирование соответствующих запросов

```
location /api/ {  
    proxy_pass http://api;  
    proxy_set_header Host    $host;  
    proxy_set_header X-Real-IP $remote_addr;  
    rewrite ^/api/$ / break ;  
}
```

Если оно присутствует, но не срабатывает, то следует проанализировать логи контейнера `frontend` на наличие сообщений об ошибках в конфигурации `nginx`. Смотрите раздел получение логов контейнера.

Если Вы получаете 404 ошибку (см. рисунок 4-4), то это свидетельствует о недоступности данной страницы. Данная страница используется в чисто диагностических целях. Комплекс «Интеграл-Д» при этом может оставаться работоспособным. Можно продолжить проверку работоспособности.

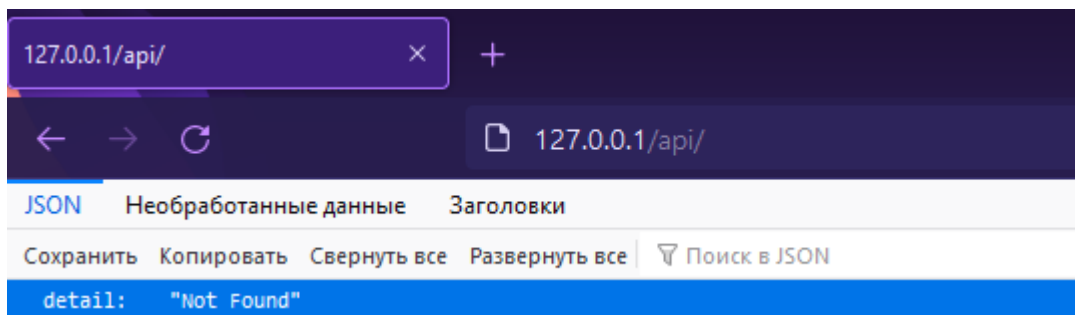


Рисунок 4-4 – иллюстрация ошибки 404.

Проверьте наличие строки `rewrite ^/api/$ / break ;` в подсекции `location /api/` файла `./conf/nginx/default.conf`

4.2. Проверка доступности функционала организаций

Откройте адрес <http://127.0.0.1/api/v1/organizations>. Если Вы видите JSON (для первой установки пустой, см. рисунок 4-5), то функционал доступен. Переходите к следующему пункту.

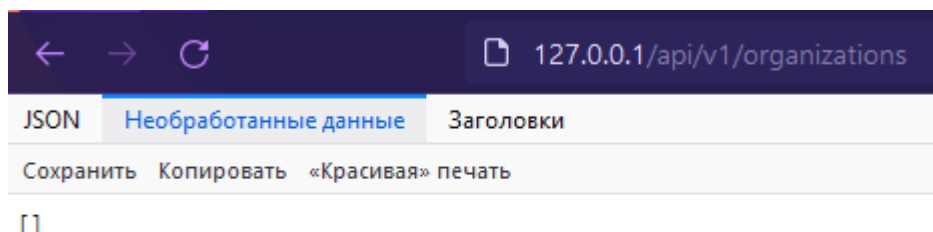


Рисунок 4-5 – иллюстрация JSON

Если Вы видите сообщение `Internal Server Error` (см. рисунок 4-6), то вероятно была пропущена выполнение команды `migrate.sh`. Вернитесь к инструкции по установке/обновлению и выполните все действия, описанные в ней.

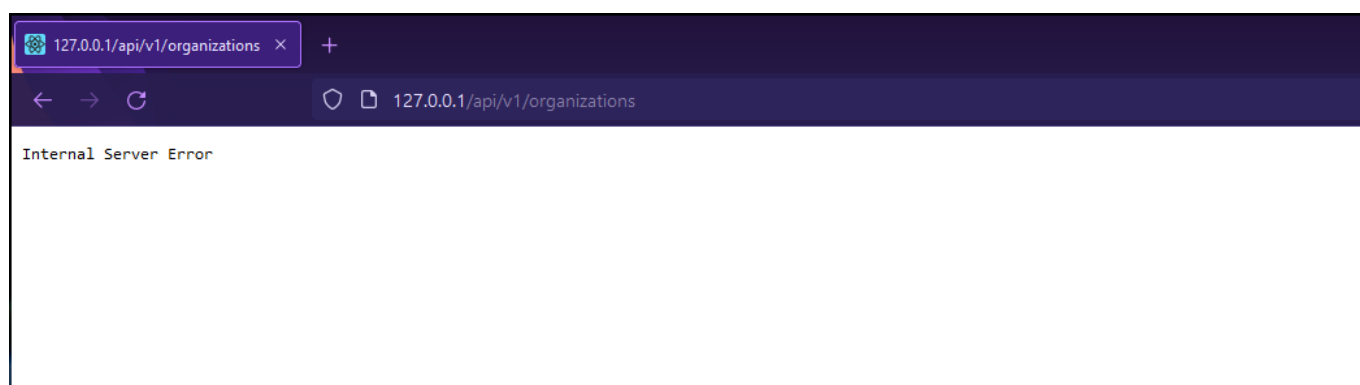


Рисунок 4-6 – иллюстрация сообщения `Internal Server Error`

Если Вас перенаправляет на основную страницу интерфейса, то неверно настроено проксирование запросов для адресов вида <http://127.0.0.1/api/>. Пути решения смотрите в пункте 4.1.

В случае, если поведение не описывается не одним из этих случаев, следует переходить к анализу логов контейнера project_api.

4.3. Проверка доступности функционала проектов

Перейдите по ссылке <http://127.0.0.1/api/v1/constellations/projects/>. Если Вы видите JSON, то функционал доступен. Переходите к следующему пункту.

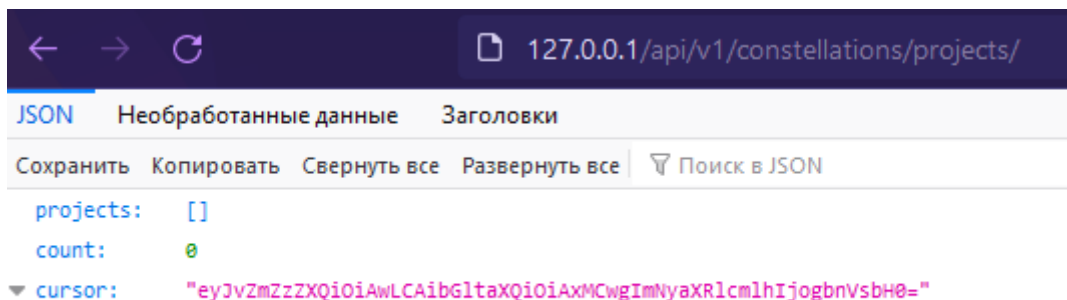


Рисунок 4-7 – переход к следующему пункту установки при наличии JSON

Если Вас перенаправляет на основную страницу интерфейса, то неверно настроено проксирование запросов для адресов вида <http://127.0.0.1/api/>. Пути решения смотрите в пункте 4.1

В случае, если поведение не описывается ни одним из этих случаев, следует переходить к анализу логов контейнера project_api.

4.4. Проверка доступности функционала проверки авторизации

Перейдите по ссылке <http://127.0.0.1/users/me>. Если Вы видите JSON, сообщающий об отсутствии логина (Unauthorized), или JSON, содержащий сведения о Вашей учетной записи, то функционал доступен, иллюстрация приведена на рисунке 4-8 либо 4-9. Переходите к следующему пункту.

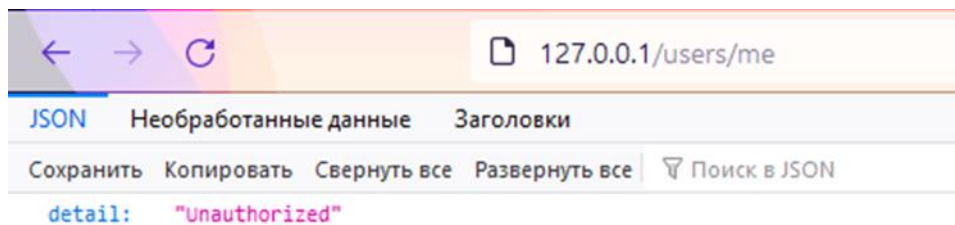


Рисунок 4-8 – Информация об отсутствии аутентификации

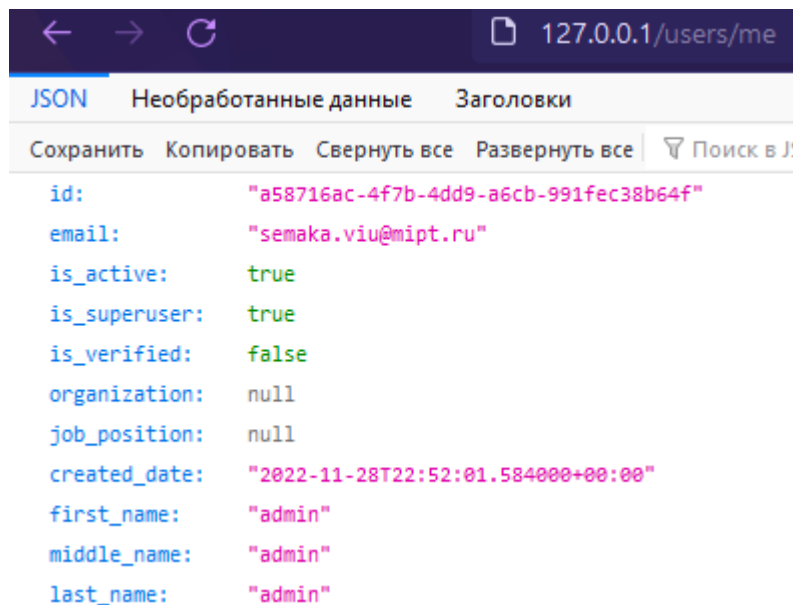


Рисунок 4-9 – Информация об учетной записи

Если Вас перенаправляет на основную страницу интерфейса, то неверно настроено проксирование запросов для адресов вида <http://127.0.0.1/users/>. Проверьте, что в секции server файла `./conf/nginx/default.conf` действительно присутствует проксирование соответствующих запросов.

```
location /users/ {  
    proxy_pass http://api;  
    proxy_set_header Host $host;  
    proxy_set_header X-Real-IP $remote_addr;  
}
```

Если оно присутствует, но не срабатывает, то следует проанализировать логи контейнера frontend на наличие сообщений об ошибках в конфигурации nginx. Смотрите раздел получение логов контейнера.

В случае, если поведение не описывается ни одним из этих случаев, следует переходить к анализу логов контейнера project_api.

4.5. Проверка доступности функционала аутентификации

Перейдите по ссылке <http://127.0.0.1/auth/login>. Если Вы видите JSON, содержащий "Method Not Allowed", то функционал доступен. Переходите к следующему пункту.

Если Вас перенаправляет на основную страницу интерфейса, то неверно настроено проксирование запросов для адресов вида <http://127.0.0.1/auth/>. Проверьте, что в секции server файла `./conf/nginx/default.conf` действительно присутствует проксирование соответствующих запросов

```
location /auth/ {
    proxy_pass http://api;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
}
```

Если оно присутствует, но не срабатывает, то следует проанализировать логи контейнера frontend на наличие сообщений об ошибках в конфигурации nginx. Смотрите раздел получение логов контейнера.

В случае, если поведение не описывается ни одним из этих случаев, следует переходить к анализу логов контейнера project_api.

4.6. Проверка работоспособности

Перейдите по ссылке <http://127.0.0.1>. Вы должны увидеть страницу логина, см. рисунок 4-10, если ранее не была произведена авторизация из используемого браузера. Использование режима инкогнито позволяет это обеспечить.

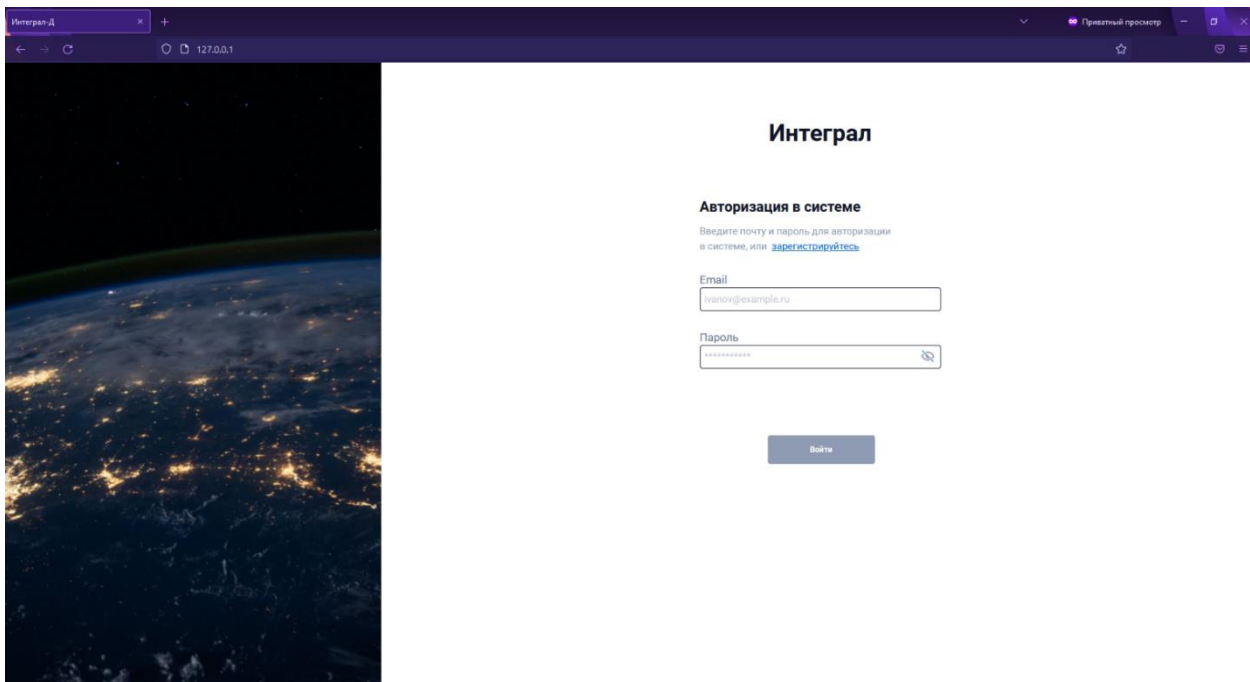


Рисунок 4-10 – страница логина программного комплекса

Если Вы видите страницу с надписью: «НЛО перехватило ваши данные», то контейнер `project_apr` по каким-либо причинам недоступен. Это нормальное явление при старте (сразу после выполнения команды ``sudo docker-compose up -d``). Если это продолжается дольше 5 минут, то следует перейти к разделу анализ логов `project_apr`.

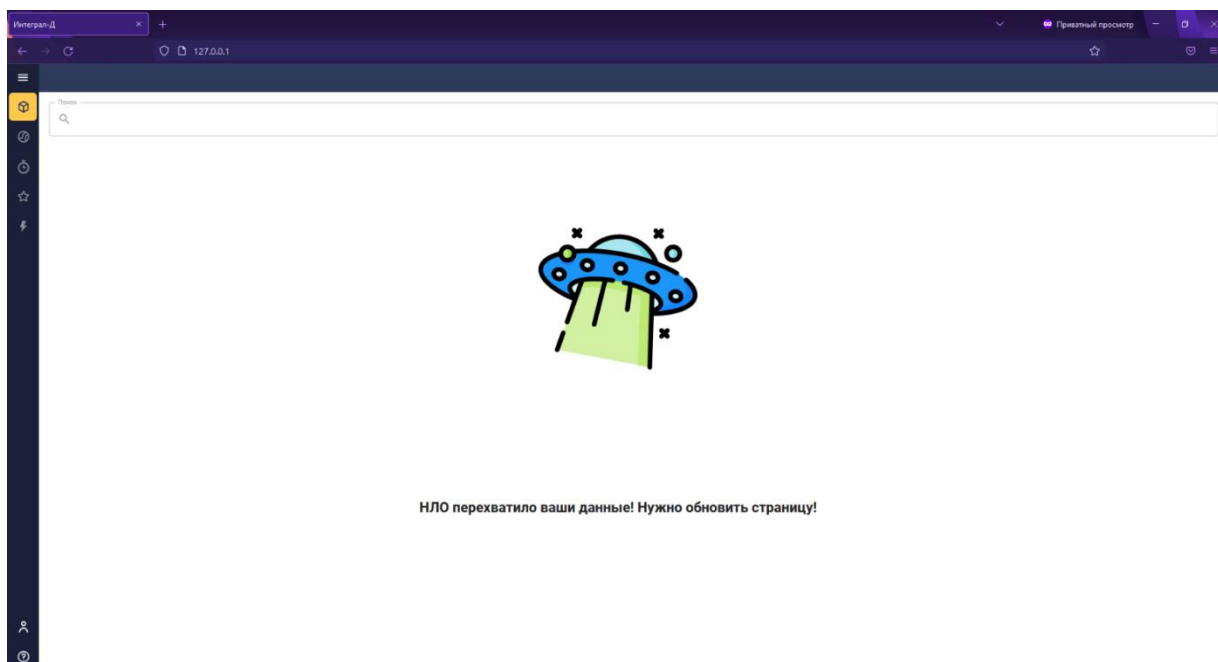


Рисунок 4-11 – иллюстрация недоступности `project_apr`

Если Вы видите начальную страницу, см. рисунок 4-12, хотя ещё ни разу не авторизовались с текущего браузера, то стоит перейти к проверке доступности функционала проверки авторизации (/users/me).

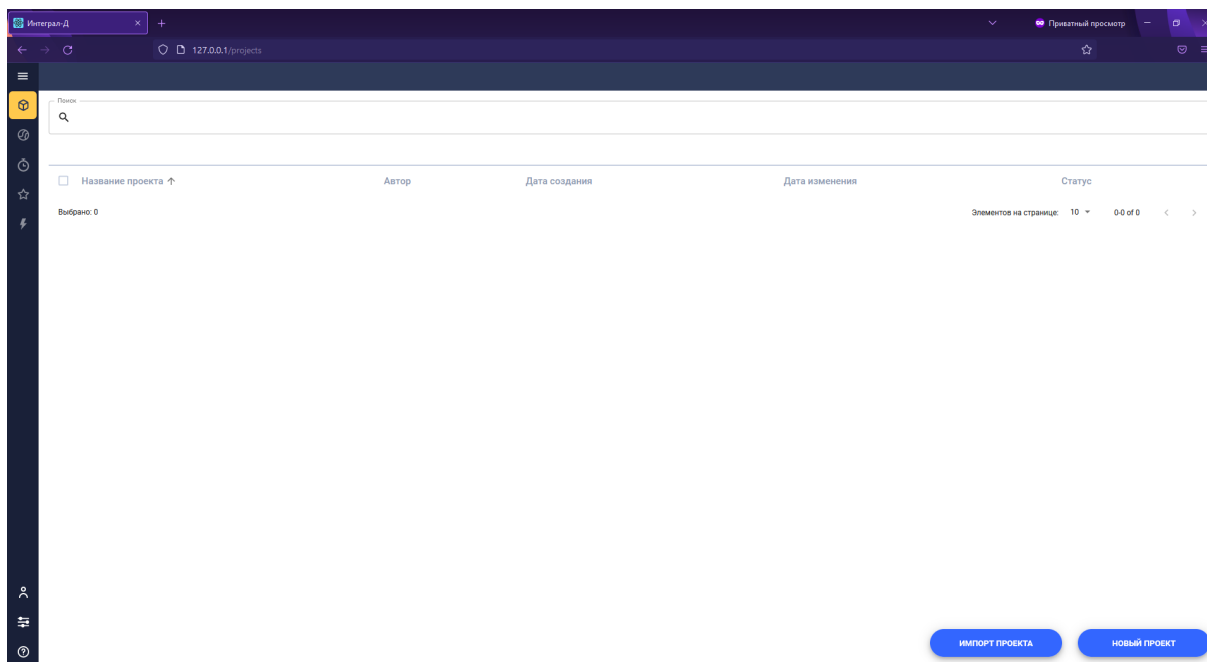


Рисунок 4-12 – иллюстрация начальной страницы комплекса

Если Вы видите сообщение, что соединение было сброшено, см. рисунок 4-13, то это означает, что сервер nginx не стартовал совсем. Следует анализировать логи контейнера frontend. Один из случаев, когда по каким-либо причинам в момент запуска контейнера frontend выключен какой-либо из контейнеров, к которому проксируются запросы. О такой ситуации будет свидетельствовать сообщение в логах, подобное ``nginx: [emerg] host not found in upstream "project_api:8000" in /etc/nginx/conf.d/default.conf:2``. В таком случае сначала следует запустить контейнер, от которого зависит frontend, а потом только сам frontend. Также можно все контейнеры запустить одновременно командой ``sudo docker-compose up -d``. Эта ошибка возникнет при ручном включении/отключении отдельных контейнеров, если перепутать порядок их включения.

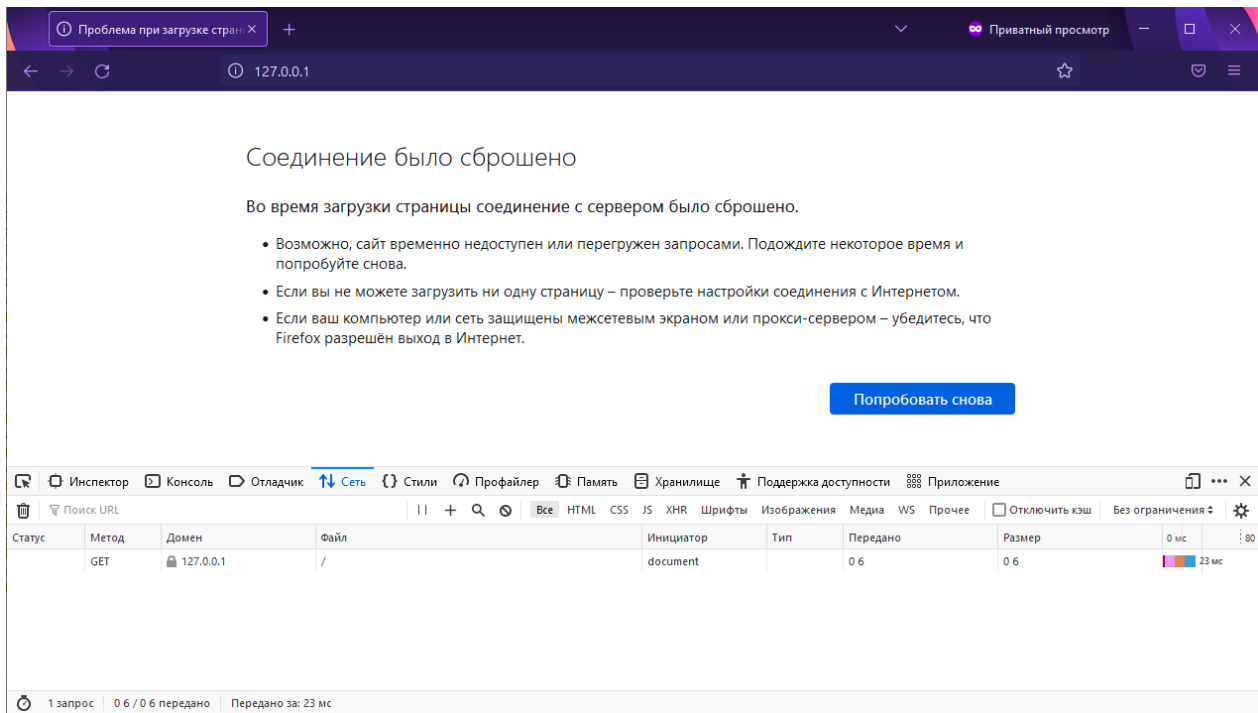


Рисунок 4-13 – Иллюстрация сброса соединения

4.7. Проверка карт

Перейдите по ссылке <http://127.0.0.1/tiles/tile/1/1/1.png>. Если Вы видите картинку с изображением Австралии, Африки и Антарктиды, сервер карт работает нормально, см. рисунок 4-14.

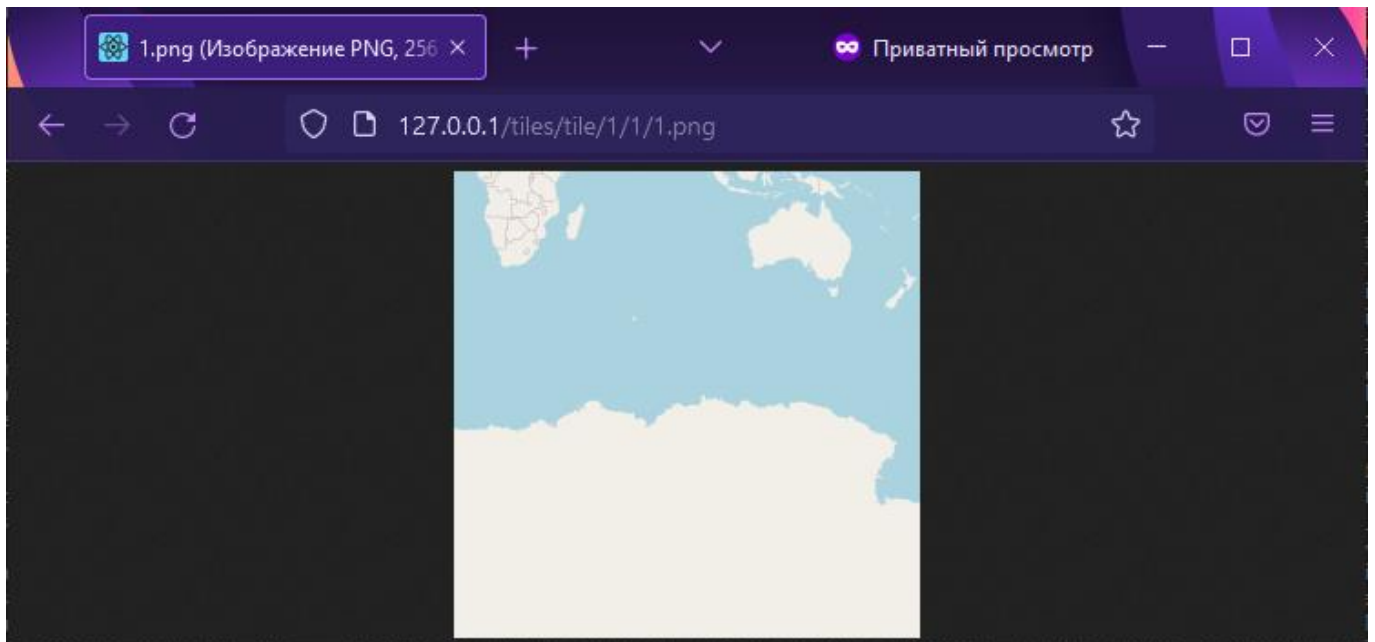


Рисунок 4-14 – иллюстрация работы сервиса карт

Если Вы получаете сообщение, что запрашиваемый адрес не существует, см. рисунок 4-15, то следует запустить генерацию карт ``sudo setup_tiles.sh``.

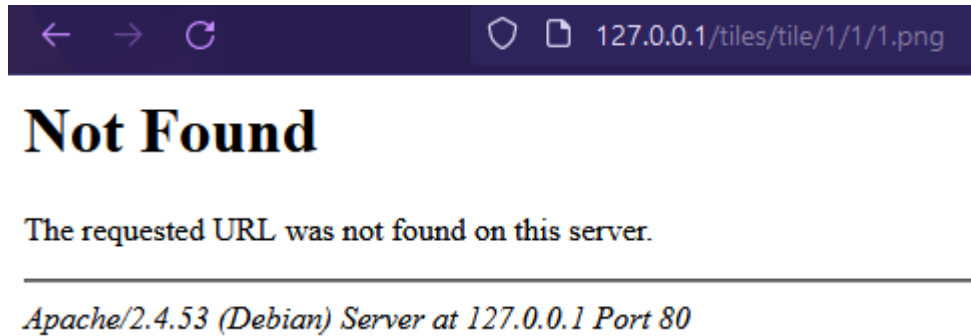


Рисунок 4-15 – Иллюстрация отсутствия допуска к сервису карт.

Если Вас перенаправляет на основную страницу интерфейса, то неверно настроено проксирование запросов для адресов вида <http://127.0.0.1/tiles/>. Проверьте, что в секции `server` файла ``./conf/nginx/default.conf`` действительно присутствует проксирование соответствующих запросов

```
location /tiles/ {  
    proxy_pass http://tiles/;  
    proxy_set_header Host $host;  
    proxy_set_header X-Real-IP $remote_addr;  
}
```

Если оно присутствует, но не срабатывает, то следует проанализировать логи контейнера `frontend` на наличие сообщений об ошибках в конфигурации `nginx`. Смотрите раздел получение логов контейнера.

В случае, если поведение не описывается ни одним из этих случаев, следует переходить к анализу логов контейнера `project_api`.

4.8. Создание проекта

Нажмите на кнопку `новый проект`. Должно открыться окно с параметрами нового проекта, результат смотри на рисунке 4-16.

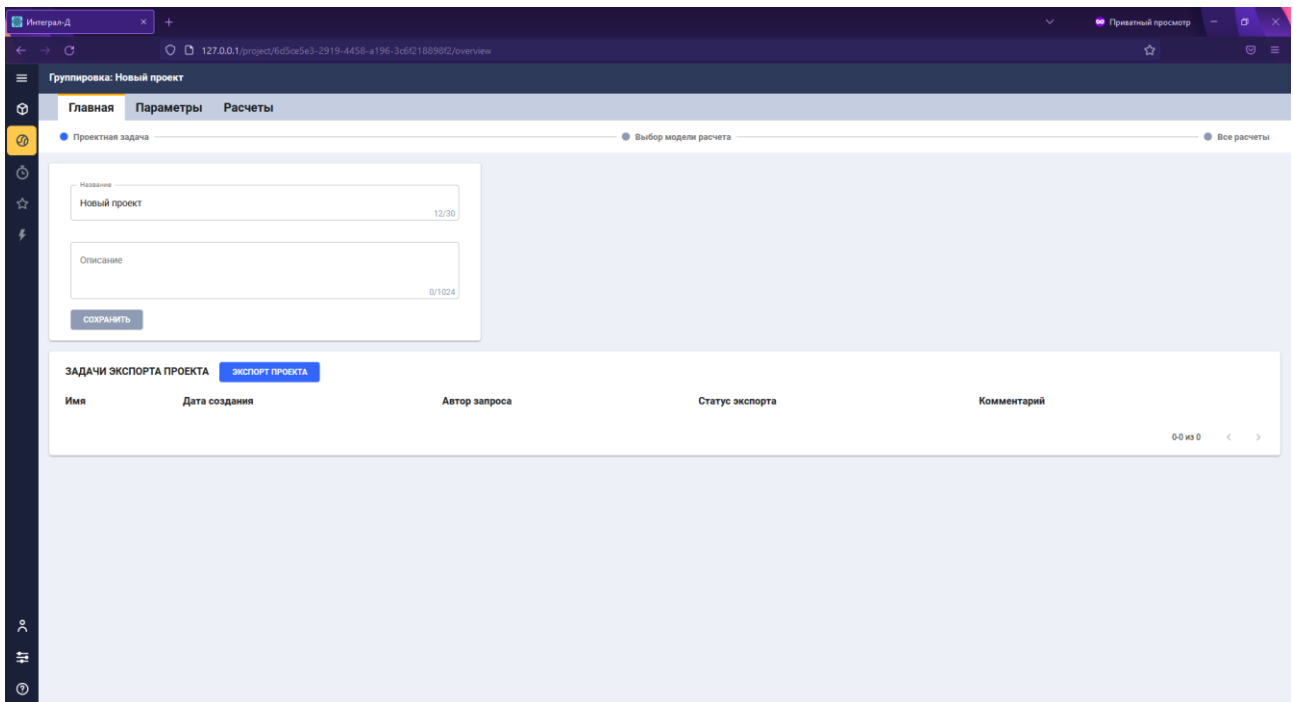


Рисунок 4-16 – Иллюстрация начала создания проектов в программном комплексе

В случае, когда интерфейс не реагирует явным образом на нажатие кнопки **новый проект**, следует анализировать результат выполнения запроса.

Если в качестве ответа выдается ошибка **500 Internal Server Error**, то следует анализировать логи контейнера `project_api`. Если был пропущен шаг ``sudo migrate.sh``, то сервер будет как раз выдавать эту ошибку с кодом **500**, см. рисунок 4-17.

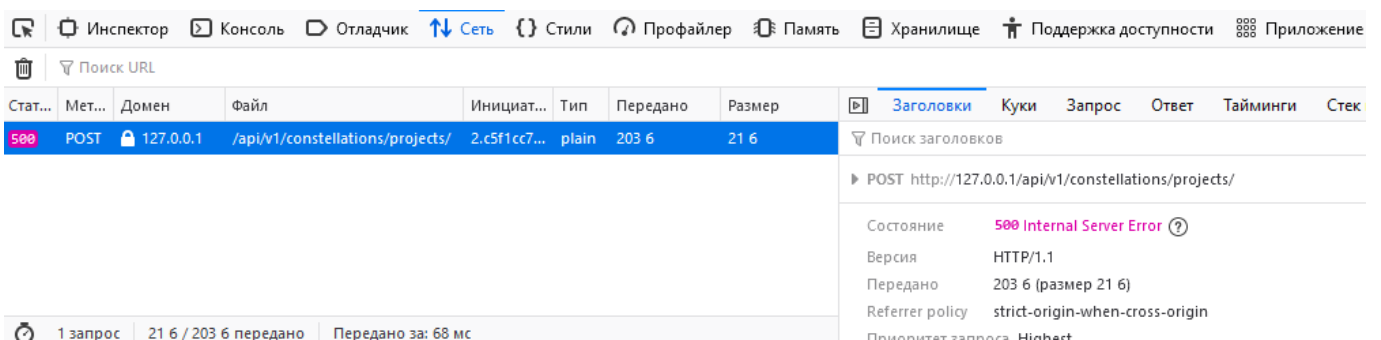


Рисунок 4-17 – иллюстрация ошибки с кодом **500** при попытке инициировать проект в программном комплексе.

Если в качестве ответа выдается ошибка 502 Bad Gateway (см. рисунок 4-18), то контейнер `project_api` не запущен или неправильно настроен `nginx`. Следует перейти к пунктам 4.1–4.3.

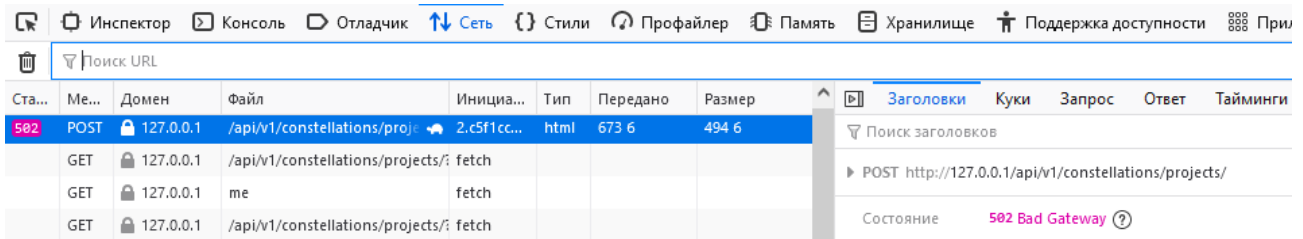


Рисунок 4-18 – Иллюстрация ошибки 502.

Если в качестве ответа выдается ошибка 405 Not Allowed (см. рисунок 4-19), то проблемы с проверкой прав пользователя на создание проекта, следует воспроизвести проверку, описанную в пункте 4.4 и удостовериться, что авторизация работает.

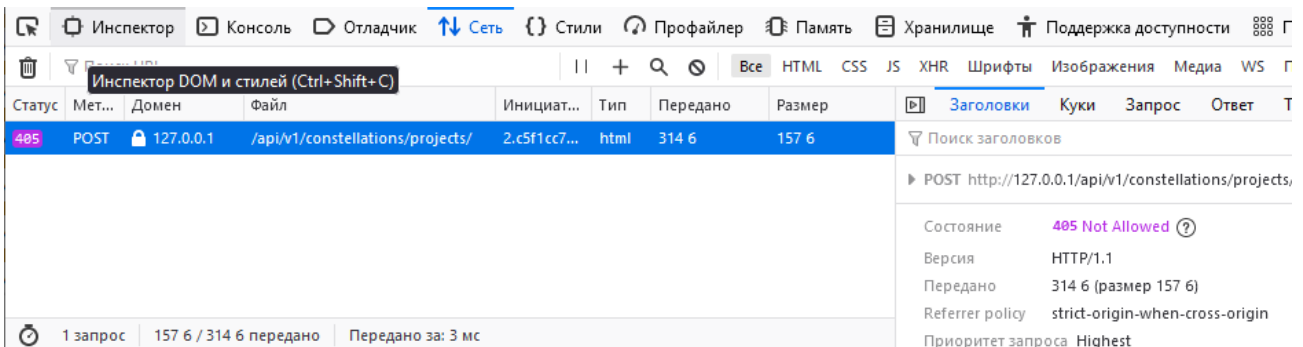
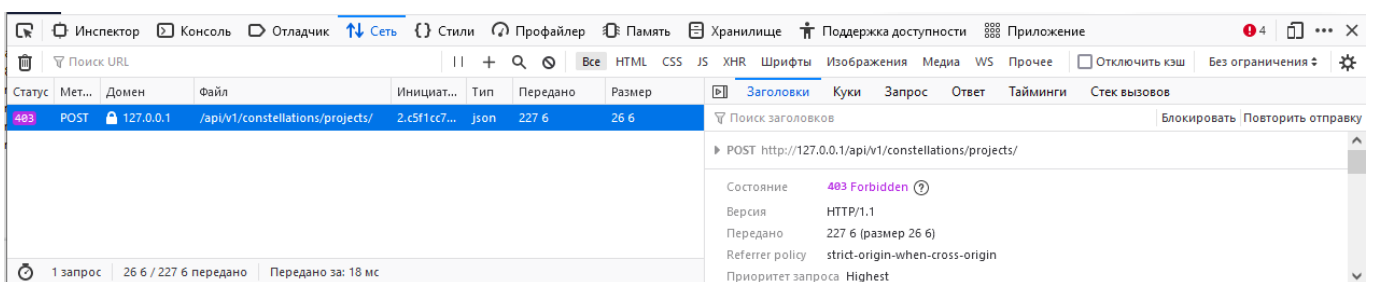


Рисунок 4-19 – иллюстрация ошибки 405

Если в качестве ответа выдается ошибка 403 Forbidden (см. рисунок 4-20), то пользователю запрещено создавать новые проекты. Все пользователи, кроме администраторов (создаваемых через консоль как описано в пункте 3.5) изначально не имеют никаких прав ни на просмотр проектов, ни на их создание. Все права выделяются администратором для группы, а пользователь получает права тех групп, в которые его добавил администратор. Данный процесс описан в инструкции пользователя.



4.9. Получение логов контейнера

Получить логи контейнера можно различными способами:

1. Команда `\`sudo docker logs integral_d_project_api_1\`` позволяет смотреть события, произошедшие до её запуска
2. Команда `\`sudo docker attach integral_d_project_api_1\`` позволяет в режиме реального времени смотреть вывод контейнера. Удобна для анализа событий, происходящих в реальном времени. Особенно, когда мы сами их можем инициировать

4.10. Анализ логов project_api

Получите логи любым удобным способом из описанных в предыдущем пункте.

1. Нормальное функционирование. Логи содержат информацию о старте и/или информацию о поступивших http запросах, см. рисунок 4-21.

```

asctime: "2022-11-29 00:32:03,992", "levelname": "INFO", "message": "Started server process [6]", "color_message": "Started server process [\u001b[36m\u001b[0m]"
asctime: "2022-11-29 00:32:03,992", "levelname": "INFO", "message": "Waiting for application startup."
asctime: "2022-11-29 00:32:04,000", "levelname": "INFO", "message": "Logger configured"
asctime: "2022-11-29 00:32:04,001", "levelname": "INFO", "message": "Application startup complete."
asctime: "2022-11-29 00:32:04,001", "levelname": "INFO", "message": "Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)", "color_message": "Uvicorn running on [\u001b[1m"
asctime: "2022-11-29 00:32:29,087", "levelname": "INFO", "message": "192.168.16.3:53564 - \"GET /api/v1/constellations/projects/?limit=20&order=name HTTP/1.0\" 401"}
asctime: "2022-11-29 00:32:29,088", "levelname": "INFO", "message": "192.168.16.3:53566 - \"GET /users/me HTTP/1.0\" 401"}
asctime: "2022-11-29 00:32:29,090", "levelname": "INFO", "message": "192.168.16.3:53568 - \"GET /api/v1/constellations/projects/?limit=10&order=name HTTP/1.0\" 401"}

```

Рисунок 4-21 – Логи с информацией о старте и/или информацию о поступивших http запросах

2. Логи содержат информацию о доступности баз данных, см. 4-22.

```

External services is unavailable - sleeping
2022/11/28 22:07:16 Waiting for: tcp://mongo:27017
2022/11/28 22:07:16 Waiting for: tcp://postgres:5432
2022/11/28 22:07:16 Waiting for: tcp://minio:9000
2022/11/28 22:07:16 Waiting for: tcp://redis:6379
2022/11/28 22:07:16 Connected to tcp://minio:9000
2022/11/28 22:07:16 Connected to tcp://postgres:5432
2022/11/28 22:07:26 Problem with dial: dial tcp: i/o timeout. Sleeping 1s
2022/11/28 22:07:26 Timeout after 10s waiting on dependencies to become available: [tcp://mongo:27017 tcp://postgres:5432 tcp://minio:9000 tcp://redis:6379]

```

Рисунок 4-22 – Логи с информацией о доступности баз данных

Запросы для проверки выполняются раз в 10 секунд. В данном примере недоступны базы данных mongo и redis. Это может быть вызвано несколькими причинами, которые следует проверять по очереди:

- Надо убедиться, что соответствующий контейнер/сервер запущены. При необходимости перезапустить.

– Произошла ошибка при старте соответствующего контейнера/базы данных => Надо смотреть логи проблемного контейнера и решать возникшую там проблему.

– Сеть настроена неверна и база данных недоступна => Проверить доступность контейнера/сервера с помощью команды ping.

– В переменных окружения указан неверный адрес базы данных => Проверить, что база данных доступна по заданному адресу

– Ошибка в аутентификационных данных => Проверить, что с логином/паролем, которые передаются внутрь контейнера действительно можно авторизоваться в БД.

3. В логах содержится ошибки о несуществующих таблицах и отношениях => Вероятно, миграции не применялись => В случае новой установки лучше повторить весь процесс установки. В случае обновления, выполнить `sudo migrate.sh`. Пример соответствующего сообщения

```
sqlalchemy.exc.ProgrammingError: (psycopg2.errors.UndefinedTable)
relation \"organization\" does not exist
```

5. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Для проведения испытаний требуется на АРМ оператора: Веб-браузер с поддержкой HTML 2.0, JavaScript 1.0, виртуальной машины Java™ (JVM), Java Runtime Environment (JRE) версии 7, WebGL 2 и поддержка cookie. Например, Mozilla Firefox версии 24 или выше, Chrome 49.0.2623.112 или выше, Safari 7 или выше.

Для серверной компоненты Программный комплекс «Интеграл-Д» с полным набором модулей Docker.

Минимальные требования к компонентам АРМ оператора для испытания ДПКМ «Интеграл-Д»: клавиатура, мышь, процессор Intel Core i3-4350, 8 ГБ оперативной памяти, видеокарта GeForce GT 1030, монитор с разрешением

1280x720 пикселей. Рекомендуются процессор Intel Core i5-8400, видеокарта GeForce GTX 1060 и монитор с разрешением 1920x1080 пикселей.

Минимальные требования для развертывания серверной части: клавиатура, мышь, процессор Intel Core i5-8400, 16 ГБ оперативной памяти, не менее 100 ГБ свободного пространства на жестком диске, монитор с разрешением 1280x720 пикселей.