

**Описание функциональных характеристик
продукта «Platform V DataSpace»**

Код продукта: АРТ

Описание функциональных характеристик компонента DataSpace Core (DSPC)

Описание функциональных характеристик

О документе

Документ содержит описание функциональных характеристик компонента DataSpace Core продукта Platform V DataSpace.

Основные понятия

В таблице приведены основные аббревиатуры и сокращения:

Аббревиатура, сокращение	Расшифровка
ACID	Atomicity, Consistency, Isolation, Durability — свойства транзакции БД обеспечивать атомарность, целостность, изолированность и постоянство изменений
API	Application programming interface. Набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом)
SQL	Structured Query Language . Декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных
XML	eXtensible Markup Language . Расширяемый язык разметки
АС	Автоматизированная система
БД	База данных
СУБД	Система управления базами данных

В таблице приведены основные термины и определения:

Термин	Определение
DataSpace Core	Серверная часть компонента DataSpace Core программного продукта Platform V DataSpace
Java	Объектно-ориентированный язык программирования Java
Kafka	Apache Kafka (рекомендован Продукт Platform V Corax)
SDK	Клиентская часть программного продукта DataSpace
Клиент	Внешняя ссылка на сущность Клиент в мастер-системе ведения клиентов (идентификатор сущности Клиент во внешней системе)
Компонент	Компонент DataSpace Core продукта Platform V DataSpace
Рабочий артефакт	Продукт работы DataSpace Client Tools - набор скомпилированных классов java и других ресурсов, собранный в JAR-архив и

	предназначенный для кастомизации DataSpace для работы с предметной моделью потребителя
Разъединение (разливание) клиентов	Разъединение клиентских профилей вследствие ошибочной дедубликации. Деактивированный ранее (дедублицированный) профиль восстанавливается, создается новый профиль клиента, в который переносятся оставшиеся данные ошибочного профиля, данные ошибочного профиля деактивируются

Назначение

DataSpace Core (далее также DataSpace или компонент) — компонент продукта Platform V DataSpace, дающий разработчикам frontend- и backend-приложений возможность сформировать набор сервисов для работы с данными в терминах описанной ими модели данных своей предметной области и использовать их для выполнения следующих операций: создание, изменение, удаление, объединение в транзакционные группы, гибкий поиск объектов и получение их свойств. Объектная модель описывается в интуитивно понятном XML-формате, знания СУБД и SQL не требуются. Для управления объектами в разрабатываемых приложениях предусмотрен ограниченный набор шаблонных конструкций языка Java, а также простой GraphQL-интерфейс.

Цель создания

Использование сервиса DataSpace ускоряет процесс разработки бизнес-приложений уровня предприятия (line-of-business applications (LOB)), в том числе продуктовых фабрик.

Основные функции (сущностные)

DataSpace предоставляет следующие базовые функции:

Функция	Сервис, предоставляющий функцию	Аргументы	Результат
Генерация рабочих артефактов	DataSpace Client Tools	Проект, содержащий подготовленную разработчиком предметную модель	Рабочий артефакт, Скрипты миграции БД, DataSpace Java SDK
Исполнение пакета команд работы с хранимыми данными (CRUD)	DataSpace Core	Пакет команд работы с данными в терминах предметной модели — создание/изменение/получение/удаление	Выполнены проверки целостности, состояние объектов (определяются составом модели) изменено,

Функция	Сервис, предоставляющий функцию	Аргументы	Результат
			данные получены
Исполнение поискового запроса	DataSpace Core	Условия поиска в терминах предметной модели	Набор отобранных по условиям поиска объектов
Обработка событий слияния/различия клиентов	DataSpace StateMachine	Данные события — тип слияния, id агрегата, id передающей стороны, id получающей стороны	Все данные (или только идентифицированный агрегат) перепривязаны от передающей стороны к принимающей стороне в соответствии с типом слияния
Загрузка локальных справочников	DataSpace Core	Данные версии локальных справочников в формате JSON	Загружена новая версия данных справочников
Исполнение поискового запроса по всем шардам	DataSpace Multisearch	Условия поиска в терминах предметной модели	Набор отобранных по условиям поиска объектов
Выполнение задач инициализации реплики	DataSpace Migration	Список партиций	Поток прочитанных из БД объектов по числу партиций
Выполнение задач технического контроля данных	DataSpace Migration	Список сущностей для контроля	Набор прочитанных из БД объектов по списку сущностей

Генерация рабочих артефактов

Компонент предусматривает моделирование предметной области в интуитивно понятном XML-формате. Предметная модель включает в себя: - объектную модель — сущности, отношения агрегации, отношения наследования, ссылки, встраиваемые типы; - модель статусов — набор статусов и допустимых переходов между ними для сущностей; - локальные справочники — набор данных локальных справочников.

Сформированная модель предметной области используется для генерации: - рабочих артефактов DataSpace (скомпилированных Java-классов в исполняемом JAR-архиве); - скриптов миграции БД (Liquibase *changelog.xml*); - DataSpace Java SDK — клиентской библиотеки в виде JAR-архива, дающей возможность простого программирования операций с данными в терминах предметной области.

Для этого модель помещается в проект, созданный с использованием maven-архетипа DataSpace, после чего выполняется сборка проекта.

Варианты сборки определяются параметрами, заданными в конфигурации сборки проекта.

Исполнение пакета команд

Функция исполнения пакета команд предназначена для управления данными.

Пакет представляет собой набор команд, где каждая команда — одна атомарная операция.

Список поддерживаемых команд: * создание сущности; * изменение сущности; * удаление сущности; * получение сущности; * блокировка сущности; * разблокировка сущности.

При исполнении пакета команд гарантируется: * последовательное выполнение команд; * согласованное изменение данных в рамках одной транзакции; * откат изменений в случае ошибок.

Для пакета команд обеспечиваются свойства ACID с уровнем изоляции read committed в текущей БД. В репликах БД гарантируется изменение только committed данных.

Примечание

В DataSpace реализована возможность идемпотентного выполнения пакета команд: повторный идентичный вызов, сделанный один или несколько раз подряд, не изменяет состояние системы. Применительно к DataSpace повторение вызова в рамках одного пакета не изменяет состояние агрегата.

Исполнение поискового запроса

Функция исполнения поискового запроса предназначена для поиска и выборки данных.

Поисковый запрос представляет собой описание того, какие данные нужно найти и вернуть пользователю.

Имеются следующие виды поисковых запросов: * поиск сущностей; * слияние поисков разнородных сущностей; * выборка свойств.

При запросе данных поисковым запросом обеспечена согласованность по чтению получаемого набора данных. При постраничной вычитке запроса согласованность по чтению между запросами страниц не обеспечивается.

Поиск сущностей

Примечание

Поиск может выполняться как по данным в одном shard (в одной БД), так и по данным из всех shards в случае мультишардовой конфигурации потребителя или DataSpace. Мультишардовая конфигурация - предполагает развертывание нескольких экземпляров сервиса DataSpace со своими БД, каждый из которых отвечает за свое подмножество данных.

Данный вид поискового запроса позволяет выбрать объекты одного типа.

При исполнении поискового запроса имеется возможность задать: * критерии поиска; * настройки пагинации: * ограничение на количество элементов; * смещение; * запрос общего количества элементов; * настройки сортировки; * перечень возвращаемых свойств.

Слияние поисков разнородных сущностей

Данный вид поискового запроса позволяет объединить поиски сущностей разных типов. При этом: * критерии поиска задаются для каждого типа сущности индивидуально; * значения конкретных свойств запрашиваются для каждого типа сущности индивидуально; * настройки пагинации и настройки сортировки задаются в рамках всего объединения.

Запрос расчетных полей

Данный вид поискового запроса позволяет получить набор расчетных значений на основе поиска сущностей одного типа. Как и при поиске сущностей, доступна возможность задать:

- критерии поиска;
- настройки пагинации;
- настройки сортировки.

Кроме того, имеется возможность задать условие получения только distinct (различающихся) наборов значений.

Обработка событий слияния/разливания клиентских данных

Компонент позволяет реализовать сценарии обработки событий дедубликации (слияния) или разъединения (разливания) клиентов или каких-либо других внешних идентификаторов.

В объектной модели объявляются внешние (по отношению к предметной модели, ведущиеся в других АС) типы, ссылки на которые должны обрабатываться сценарием слияния. Доступны следующие возможности: - запуск сценария слияния через вызов API — по определенным в модели правилам выполняется поиск всех ссылок с идентификатором передающей стороны и их замена на идентификатор принимающей стороны; - запуск сценария слияния путем подписки на события в Kafka, формируемые внешним источником; - запуск сценария передачи агрегата с заданным идентификатором — аналогично слиянию, но поиск и замена производятся не по всей БД, а только по заданному агрегату; - запуск сценария разливания — перепривязка ранее слитых идентификаторов к исходному идентификатору передающей стороны и к новому идентификатору принимающей стороны.

Обработка задач инициализации реплики и технического контроля данных

Компонент в качестве опции поддерживает передачу данных в хранилище данных посредством сервиса Platform V Archiving.

Обрабатываются запросы от сервиса ТСА: - на инициализацию реплики — в ответ передаются все объекты предметной модели, сохраненные в БД; - на технический контроль данных — в ответ передаются объекты по списку запрошенных сущностей, полученные из БД.

Сценарии использования

DataSpace представляет собой набор инструментов для ускорения реализации бизнес-приложений и сокращения затрат на разработку в части работы с хранимыми данными.

Процесс разработки бизнес-приложений с использованием DataSpace включает следующие сценарии использования (СИ):

- СИ01 Разработка модели данных.
- СИ02 Выпуск сервиса DataSpace и DataSpace Java SDK.
- СИ03 Разработка бизнес-приложения.
- СИ04 Использование бизнес-приложения.

СИ01 Разработка модели данных

Акторы: Разработчик (модели данных), Система.

Предусловия: на основе архетипа DataSpace создан Maven-проект бизнес-приложения.

Постусловия:

1. В проекте пользовательского приложения созданы файлы предметной модели.
2. Сформирован артефакт DataSpace Java SDK.

Триггер: Разработка предметной модели по производственной необходимости.

Шаги основного сценария:

1. Подготовить модель данных.

Примечание

Модель данных включает в себя описание объектов предметной области и возможных состояний этих объектов. В случае необходимости разработчик производит локальные тесты сервера DataSpace Core через DataSpace Java SDK и API.

2. Собрать рабочий артефакт DataSpace для кастомизации DataSpace Core под модель данных.
3. Собрать артефакт DataSpace Java SDK.

Альтернативные сценарии:

- При наличии ошибок в модели данных процесс сборки прервется. Разработчику необходимо устранить ошибку и повторить сборку.
- При выявлении нарушений обратной совместимости модели данных процесс сборки прерывается. Разработчику необходимо доработать модель данных для соблюдения обратной совместимости и повторить сборку.

СИ02 Выпуск сервиса DataSpace и DataSpace Java SDK

Актеры: Инженер (инженер DevOps), Система.

Предусловия:

1. Собран рабочий артефакт DataSpace для кастомизации DataSpace Core для подготовленной модели.
2. Стенд для развертывания сервиса.
3. Доступ к серверу БД.

Постусловия:

1. На стенде (ИФТ+) развернут API DataSpace для подготовленной модели.
2. Сервис DataSpace развернут и доступен для эксплуатации.

Шаги основного сценария:

1. Произвести первоначальную настройку стенда:
 1. Создать и сконфигурировать аккаунты и секреты.
 2. Настроить балансировщик.
 3. Создать базу данных или схему для приложения.
2. Установить DataSpace на стенд.

Альтернативные сценарии:

- В случае ошибок при развертывании выполнить откат версии DataSpace до предыдущей версии.
- В случае обратно несовместимых изменений схемы данных выполнить откат схемы данных до предыдущей версии.

СИ03 Разработка бизнес-приложения

Актеры: Разработчик (бизнес-приложение), Система.

Предусловия: имеется доступ к стенду с развернутым сервисом DataSpace + SDK.

Постусловия: бизнес-приложение, использующее сервис DataSpace + SDK, разработано.

Шаги основного сценария:

1. Разработчик разрабатывает бизнес-приложение, которое реализует автоматизированные бизнес-процессы.

2. В рамках автоматизируемых бизнес-процессов изменения состояний объектов реализованы как вызовы DataSpace через API (GraphQL и/или JSON-RPC 2.0) или с использованием Java SDK.

Альтернативные сценарии не предусмотрены.

СИО4 Использование бизнес-приложения

Актеры: Пользователь, Приложение, Система (стенд DataSpace).

Предусловия:

1. Сервис DataSpace + SDK развернуты и доступны.
2. Бизнес-приложение доступно.
3. Пользователь авторизован в приложении.

Постусловия:

1. Пользователь выполнил шаги бизнес-процесса.
2. В результате выполнения шагов бизнес-процесса Приложение создало требуемые объекты или изменило свойства существующих объектов.
3. Состояния объектов записаны в БД DataSpace.

Триггер: Необходимость использования бизнес-процесса.

Пользователь последовательно выполняет шаги сценария. Действия пользователя с объектами DataSpace реализованы следующим образом:

1. Пользователь в бизнес-приложении создает объекты, производит их поиск или изменяет их свойства.
2. Бизнес-приложение преобразует действия пользователя в команды работы с данными и передает их серверу DataSpace через DataSpace Java SDK или DataSpace API.
3. Сервер DataSpace обрабатывает команды работы с данными:
 - Проверяет атрибуты команд.
 - Выполняет полученные команды.
 - Обновляет информацию в базе данных.
 - Возвращает результат выполнения команд бизнес-приложению.

Альтернативные сценарии не предусмотрены.

Описание функциональных характеристик компонента DS Lab (DSLБ)

Описание функциональных характеристик

О документе

Документ содержит описание функциональных характеристик компонента DS Lab продукта Platform V DataSpace.

Основные понятия

В таблице приведены основные аббревиатуры и сокращения:

Аббревиатура, сокращение	Расшифровка
БД	База данных
СУБД	Система управления базами данных
API	Application programming interface. Набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом)
JWT	JSON Web Token
JWKS	JSON Web Key Sets. набор веб-ключей JSON
XML	eXtensible Markup Language . Расширяемый язык разметки

В таблице приведены основные термины и определения:

Термин	Определение
Продукт	Platform V DataSpace
Продуктовая фабрика	Бизнес-приложение или набор бизнес-сервисов (сценариев), доступных клиентам
DataSpace Core	Компонент DataSpace Core программного продукта Platform V DataSpace
DS Lab	Компонент DS Lab продукта Platform V DataSpace
IAM-service	Сервис Identity and Access Management. Система идентификации и управления доступом, предоставляющая JWT, например Platform V IAM SE или KeyCloak
Java	Объектно-ориентированный язык программирования Java
SDK	Клиентская часть программного продукта DataSpace
Сервис DataSpace	Развернутый сервер DataSpace Core с моделью данных потребителя, обслуживающий его запросы

Назначение

Цель создания

Продукт Platform V DataSpace представляет собой набор инструментов для ускорения реализации бизнес-приложений и сокращения затрат на разработку в части работы с хранимыми данными.

Компонент DS Lab продукта Platform V DataSpace предоставляет разработчикам, архитекторам и аналитикам инструмент для создания и управления сервисами DataSpace для работы с хранимыми данными в терминах моделей данных своих предметных областей.

Компонент DS Lab представляет собой web-приложение для графического проектирования модели данных потребителя DataSpace и для создания кастомизированного сервиса DataSpace по спроектированной модели данных и в соответствии с заданной конфигурацией. Сервис разворачивается в окружении потребителя DataSpace.

Компонент DS Lab включает в себя: * Визуальный редактор модели данных и конфигурации DataSpace * Инструмент выпуска и развертывания DataSpace в окружении потребителя в спроектированной конфигурации * Консоль для работы с хранимыми данными в созданном сервисе DataSpace посредством GraphQL API

Информация о созданных сервисах хранится в базе данных компонента DS Lab.

Допускается внесение изменений в спроектированную модель данных, после чего ранее созданный сервис DataSpace может быть пересоздан с сохранением данных.

Создаваемый при помощи DS Lab сервис DataSpace предназначен в первую очередь для цели разработки приложения потребителя.

Для выпуска промышленного дистрибутива DataSpace необходимо взять разработанные пользователем DS Lab модель данных, GraphQL-запросы и правила доступа и использовать их с набором инструментов компонента DataSpace Core.

Основные функции

Название функции	Потребитель функции	Аргументы функции	Результат
Создание проекта DataSpace	Разработчик модели данных	Название модели данных	Проект DataSpace в репозитории DS Lab, в котором выполняется проектирование модели данных
Создание модели данных предметной области	Разработчик модели данных	Проект DataSpace, сущности и связи предметной области, модель жизненного цикла сущностей	Модель данных предметной области в виде объектной модели и статусной модели в репозитории DS Lab

Создание (выпуск) DataSpace	Разработчик модели данных	Параметры размещения DataSpace, параметры подключения к БД потребителя	Собранный и развернутый в окружении потребителя сервис DataSpace, кастомизированный под модель данных и конфигурацию потребителя. Схема данных в БД потребителя, созданная в соответствии с моделью данных. Endpoint сервиса DataSpace для работы с данными.
Изменение модели данных предметной области	Разработчик модели данных	Проект DataSpace, изменяемые элементы модели предметной области	Схема данных в БД потребителя, мигрированная под измененную модель данных. Пересобранный и развернутый сервис DataSpace, кастомизированный под измененную модель данных.
Экспорт модели данных	Разработчик модели данных	Проект DataSpace, имя файла экспорта	Модель данных текущего проекта экспортирована в XML-файл с заданным именем
Импорт модели данных	Разработчик модели данных	Проект DataSpace, XML-файл с импортируемой моделью	Модель данных текущего проекта замещена на модель из XML-файла
Загрузка данных справочника	Разработчик модели данных	Проект DataSpace, JSON-файл импорта данных справочников	Данные из JSON-файла, загруженные в справочники в БД сервиса DataSpace
Проектирование GraphQL запросов к данным и работа с хранимыми данными	Разработчик модели данных	Проект DataSpace	Данные, полученные в результате выполнения query. Измененные данные в БД в результате выполнения mutation.
Управление разрешениями на доступ к данным сервиса	Разработчик модели данных	Проект DataSpace	Правила доступа к данным, сохраненные в репозитории DS Lab и примененные к развернутому сервису DataSpace
Мониторинг работы сервиса	Разработчик модели	Проект DataSpace	Dashboard с показателями работы сервиса DataSpace

	данных		
Удаление проекта	Разработчик модели данных	Проект DataSpace	Сервис DataSpace удален из инфраструктуры потребителя

Сценарии использования

Процесс разработки бизнес-приложений с использованием DataSpace включает следующие сценарии использования (СИ):

- СИ01 Разработка модели данных предметной области.
- СИ02 Создание (выпуск) сервиса DataSpace.
- СИ03 Проектирование GraphQL запросов к данным и работа с хранимыми данными.
- СИ04 Работа с данными через API.
- СИ05 Мониторинг ресурсов и производительности сервиса DataSpace.
- СИ06 Внесение изменений в модель данных.
- СИ07 Настройка правил доступа к данным.

СИ01 Разработка модели данных предметной области

Актеры: Разработчик модели данных предметной области.

Предусловия: настроен доступ пользователя к проекту в DS Lab (во внешнем IAM-service).

Постусловия:

4. В пользовательском проекте в DS Lab создана модель данных предметной области.
5. Модель данных прошла валидацию.
6. Сохранена резервная копия модели данных в XML-файле.

Триггер: Разработка предметной модели по производственной необходимости.

Шаги основного сценария:

Все действия относятся к пользовательскому интерфейсу DS Lab, подробное описание приведено в документе [Руководство оператора](#).

7. Создать проект (модель).
8. Спроектировать модель данных в визуальном редакторе.

Примечание

Модель данных включает в себя описание объектов предметной области и возможных состояний этих объектов.

9. Описать статусную модель сущностей модели данных.
10. Выполнить валидацию модели данных.
11. Экспортировать спроектированную модель данных в XML-файл.

Альтернативные сценарии:

- Разработчик может импортировать модель данных из XML-файла (экспортированного ранее или созданного вручную). Подробное описание приведено в документе [Руководство оператора](#).
- При наличии ошибок в модели данных разработчику необходимо исправить модель и повторить валидацию.
- При выявлении нарушений обратной совместимости модели данных процесс разработчику необходимо устранить обратно-несовместимые изменения в модели данных и повторить валидацию (со списком несовместимых изменений можно ознакомиться в разделе “Несовместимые изменения” документа “Порядок внесения изменений в модель данных” компонента DataSpace Core продукта Platform V DataSpace).

СИ02 Создание (выпуск) сервиса DataSpace

Актеры: Разработчик модели данных предметной области.

Предусловия:

12. В DS Lab подготовлена и прошла валидацию модель данных.
13. В контейнерной среде исполнения (кластер Kubernetes) для развертывания сервиса доступна достаточная квота вычислительных ресурсов.
14. Получена учетная запись в контейнерной среде исполнения с правами администратора кластера или пространства имен (namespace) для развертывания сервиса.
15. Подготовлена БД и доступ к серверу БД - сетевое подключение, схема данных, учетная запись и пароль или сертификат для БД с правами на создание объектов схемы данных.

Постусловия:

16. В контейнерной среде исполнения (в кластере Kubernetes) развернут сервис DataSpace для подготовленной модели.
17. Опубликованы API endpoints созданного сервиса DataSpace (JSON-RPC, GraphQL, gRPC).
18. В maven-репозитории доступны артефакты DataSpace Java SDK для подготовленной модели.

Шаги основного сценария:

Все действия относятся к пользовательскому интерфейсу DS Lab, подробное описание приведено в документе [Руководство оператора](#).

19. Войти в проект, содержащий выпускаемую модель.
20. Задать параметры окружения для выпускаемого сервиса DataSpace.
21. Нажать кнопку “Выпустить”.
22. Дождаться завершения процесса выпуска.

Альтернативные сценарии:

- При наличии ошибок в модели данных разработчику необходимо устранить ошибку и повторить выпуск.

СИ03 Проектирование GraphQL запросов к данным и работа с хранимыми данными.

Актеры: Разработчик (бизнес-приложение).

Предусловия:

23. Сервис DataSpace выпущен и развернут.
24. Пользователь получил доступ к проекту в DS Lab.

Постусловия: Разработаны запросы к данным бизнес-приложения.

Шаги основного сценария:

25. Разработчик разрабатывает бизнес-приложение, которое реализует автоматизированные бизнес-процессы.
26. В рамках автоматизируемых бизнес-процессов разработчик использует GraphQL-консоль DS Lab для проектирования запросов на изменение (mutation) и поиск (query) данных. Детально работа с GraphQL-запросами описана в документе “Протокол GraphQL компонента DataSpace Core” компонента DataSpace Core продукта Platform V DataSpace.
27. Разработанные запросы используются в коде бизнес-приложения.

Альтернативные сценарии:

- Разработчик может загрузить данные справочников через API сервиса DataSpace, как описано в документе [Руководство оператора](#).

СИ04 Работа с данными через API

Актеры: Бизнес-приложение.

Предусловия:

28. Сервис DataSpace развернут и доступен.
29. Бизнес-приложение доступно.
30. Бизнес-приложение передает в заголовке owner-id код тенанта.

Постусловия:

31. Пользователь бизнес-приложения выполнил в нем действия.

32. В результате действий пользователя Бизнес-приложение в соответствии с его бизнес-логикой создало требуемые объекты, изменило состояние существующих объектов или получило набор хранимых данных.
33. Состояния объектов записаны в БД сервиса DataSpace.

Триггер: Необходимость использования бизнес-процесса.

Пользователь последовательно выполняет шаги сценария. Действия пользователя с объектами DataSpace реализованы следующим образом:

34. Пользователь в бизнес-приложении создает объекты, производит их поиск или изменяет их свойства.
35. Бизнес-приложение преобразует действия пользователя в команды работы с данными и передает их серверу DataSpace через DataSpace Java SDK или DataSpace API.
36. Сервис DataSpace обрабатывает команды работы с данными:
 - Проверяет атрибуты команд.
 - Выполняет полученные команды.
 - Обновляет информацию в базе данных.
 - Возвращает результат выполнения команд бизнес-приложению.

Альтернативные сценарии:

- в случае ошибок API повторить операцию (для mutation задать ключ идемпотентности во избежание дублирования обработки данных).

СИ05 Мониторинг ресурсов и производительности сервиса DataSpace

Актеры: Разработчик модели данных предметной области.

Предусловия:

37. Проект (модель) выпущен в DS Lab.
38. В контейнерной среде исполнения (кластер Kubernetes) развернут компонент Prometheus, настроенный на сбор метрик сервисов DataSpace проекта.
39. Развернут компонент Grafana, настроенный на отображение данных из компонента Prometheus.

Постусловия:

1. Получена информация в dashboard проекта.

Шаги основного сценария:

Детально мониторинг описан в документе “Мониторинг работы сервиса”.

1. Открыть проект (модель) в DS Lab.
2. Открыть вкладку **Детали** и перейти по ссылке “**Ссылка на dashboard.**”.
3. Проанализировать отображаемые данные мониторинга.

Альтернативные сценарии:

- Мониторинг работоспособности встроенными средствами контейнерной среды исполнения, анализ логов.

СИ06 Внесение изменений в модель данных

Актеры: Разработчик модели данных предметной области.

Предусловия:

1. В DS Lab выпущена модель данных.

Постусловия:

1. В контейнерной среде исполнения (в кластере Kubernetes) развернут сервис DataSpace с обновленной моделью.
2. В maven-репозитории доступны артефакты DataSpace Java SDK для обновленной модели.

Шаги основного сценария:

Все действия относятся к пользовательскому интерфейсу DS Lab, подробное описание приведено в документе “Руководство оператора”.

1. Войти в проект, содержащий обновляемую модель.
2. Внести изменения в модель.
3. Нажать кнопку “Выпустить”.
4. Дождаться завершения процесса выпуска.

Альтернативные сценарии:

- При наличии ошибок в модели данных разработчику необходимо устранить ошибку и повторить выпуск.

СИ07 Настройка правил доступа к данным

Актеры: Разработчик модели данных предметной области.

Предусловия:

1. В DS Lab выпущена модель данных.

Постусловия:

1. Выгружен файл с описанием правил доступа к данным.
2. Сервис DataSpace осуществляет контроль доступа в соответствии с настроенными правилами.

Шаги основного сценария:

Все действия относятся к пользовательскому интерфейсу DS Lab, подробное описание приведено в документе “Система разграничения прав для GraphQL-запросов”.

1. Войти в проект, содержащий выпущенную модель.
2. Нажать кнопку **Загрузить JWKS** для загрузки **JWKS в формате JSON**. для проверки JWT.
3. Нажать кнопку **Добавить разрешение** для добавления правил доступа.
4. При необходимости изменения тел запросов для ранее созданных правил использовать кнопку **Перезагрузить запросы**.
5. Нажать кнопку **Экспорт разрешений** для выгрузки правил доступа в файл.

Альтернативные сценарии:

- Загрузить ранее сохраненные правила доступа из файла при помощи кнопки **Импорт разрешений**.