



Продукт Platform V Content Management (SDC)

Компонент Сервис шаблонов форм UI (UFTM)

Руководство по установке

Содержание

Руководство по установке продукта Platform V Content Management (SDC) **Error! Bookmark not defined.**

В руководстве приведены инструкции по установке компонента Сервис шаблонов форм UI (UFTM) продукта Platform V Content Management (SDC). . **Error! Bookmark not defined.**

Термины и определения	3
Системные требования.....	3
Системное программное обеспечение	3
* Применимо для обеспечения обратной совместимости.	9
Аппаратные требования	9
Установка.....	10
Настройка СУБД	10
Миграция конфигурации компонента UFTM в репозиторий	12
Настройка конфигурации компонента UFTM.....	12
Установка дистрибутива компонента.....	15
Обновление.....	16
Удаление	16
Проверка работоспособности	16
Откат	28
Часто встречающиеся проблемы и пути их устранения	29
Чек-лист валидации установки	29

Руководство по установке продукта Platform V Content Management (SDC)

В руководстве приведены инструкции по установке компонента Сервис шаблонов форм UI (UFTM) продукта Platform V Content Management (SDC).

Термины и определения

Список терминов и определений приведен в одноименном разделе документа [«Описание функциональных характеристик»](#).

Системные требования

Настройки безопасности окружения и перечень платформенных (дополнительных внешних) продуктов, используемых для установки, настройки и контроля в конечной информационной системе (далее — ИС), выбираются клиентом при разработке конечной ИС, исходя из характера обрабатываемой в ней информации и иных требований информационной безопасности (далее — ИБ), предъявляемых к ней.

Системное программное обеспечение

Ниже представлены категории системного программного обеспечения (далее — ПО), которые **обязательны** или **опциональны** для установки, настройки, контроля и функционирования компонента UFTM. В каждой категории перечислены все поддерживаемые продукты сторонних правообладателей. Отдельно обозначены варианты, которые рекомендует АО «СберТех» (маркировка «Рекомендовано» в столбце «Продукт, функциональная совместимость с которым подтверждена»). Клиенту необходимо выбрать один из продуктов в каждой категории, исходя из условий использования конечной ИС.

Категория ПО	Обязательность установки (да/нет) *	Наименование ПО	Версия	Продукт, функциональная совместимость с которым подтверждена **	Описание
Операционная система	Да	ОС Альт 8 СП	altlinux-sp8-5.4.113 - release	Рекомендовано	ОС контейнеров для запуска модулей компонента
		Red Hat Enterprise Linux Server	7.6	Опционально	

Категория ПО	Обязательность установки (да/нет) *	Наименование ПО	Версия	Продукт, функциональность совместимость с которым подтверждена **	Описание
Среда контейнеризации ***	Да	Kubernetes (k8s)	1.21	Рекомендовано	Платформа контейнеризации для запуска компонентов сервиса
		Red Hat OpenShift	4.6.28	Опционально	
Java-машина	Да	OpenJDK	11	Рекомендовано	Окружение для работы модулей компонента
СУБД	Да	PostgreSQL	14.2	Рекомендовано. Правообладателем АО «СберТех» также рекомендована СУБД, основанная на PostgreSQL – Platform V Pangolin SE (PSQ), см. раздел «Платформенные зависимости»	СУБД для хранения данных компонента
Брокер сообщений	Да	Kafka	2.7.0	Рекомендовано	Событийный обмен сообщениями между модулями компонента
Сервис интеграции и оркестрации микросервисов в облаке	Да	Istio	1.12	Рекомендовано. Правообладателем АО «СберТех» также рекомендован сервис интеграции и оркестрации микросервисов в облаке, основанный на Istio – Platform V Synapse Service Mesh (SSM), см. раздел	Сервис интеграции микросервисов в облаке

Категория ПО	Обязательность установки (да/нет) *	Наименование ПО	Версия	Продукт, функциональность совместимость с которым подтверждена **	Описание
				«Платформенные зависимости»	
		Red Hat OpenShift Service Mesh	2.0.6-2	Опционально	

Примечание:

*

- **Да** — категория ПО обязательна для функционирования сервиса (это означает, что сервис не может выполнять свои основные функции без установки данной категории ПО).
- **Нет** — категория ПО необязательна для функционирования сервиса (это означает, что сервис может выполнять свои основные функции без установки данной категории ПО).

**

- **Рекомендовано** — рекомендованный правообладателем АО «СберТех» продукт.
- **Опционально** — альтернативный по отношению к рекомендованному правообладателем АО «СберТех» продукт.

Здесь и далее поддерживаемой средой контейнеризации является Kubernetes (использование OpenShift – опционально). В инструкциях по настройке, в именах переменных и параметрах системы могут встречаться названия систем контейнеризации, которые одинаковы и применимы для обеих сред контейнеризации.

Платформенные зависимости

Для настройки, контроля и функционирования компонента реализована интеграция с программными продуктами, правообладателем которых является АО «СберТех»:

Наименование продукта	Код	Версия продукта	Код и наименование компонента	Обязательность установки (да/нет) **	Описание	Аналог других производителей ***
Platform V Pangolin SE	PS Q	5.1.0	PSQL Platform V Pangolin	Нет	Система управления базами	PostgreSQL 14.2

Наименование продукта	Код	Версия продукта	Код и наименование компонента	Обязательность установки (да/нет) **	Описание	Аналог других производителей ***
					данных, основанная на PostgreSQL	
Platform V DevOps Tools	DOT	1.2	CDJE Deploy tools	Да	Сервис для развертывания и обновления компонентов Платформы и приложений потребителей, для настройки и обслуживания инфраструктуры Платформы	Сервис успешно прошел испытания и подтвердил свою работоспособность с компонентом CDJE. С аналогами других производителей ей не тестировался
Platform V Frontend Std	#FS	4.1.3	* AUDE Аудит ЕФС	Нет	Сервис для сбора и хранения событий аудита	Сервис успешно прошел испытания и подтвердил свою работоспособность с компонентом AUDE. С аналогами других производителей ей не тестировался
Platform V Frontend Std	#FS	4.1.3	* LOGE Журналирование ЕФС	Нет	Сервис для хранения логов	Сервис успешно прошел испытания и подтвердил свою работоспособность с компонентом LOGE. С аналогами других производителей ей не тестировался

Наименование продукта	Код	Версия продукта	Код и наименование компонента	Обязательность установки (да/нет) **	Описание	Аналог других производителей ***
Platform V Frontend Std	#FS	4.1.3	* MONE Прикладной мониторинг ЕФС	Нет	Сервис для сбора прикладных и инфраструктурных метрик и отправки их в целевую систему хранения	Сервис успешно прошел испытания и подтвердил свою работоспособность с компонентом MONE. С аналогами других производителей не тестировался
Platform V Frontend Std	#FS	4.1.3	* CFGE Управление параметрам и ЕФС	Нет	Сервис обеспечивает хранение, управление и предоставление по запросу параметров конфигурации и библиотек, сервисов Платформы и прикладных приложений, разработанных на Платформе	Сервис успешно прошел испытания и подтвердил свою работоспособность с компонентом CFGE. С аналогами других производителей не тестировался
Platform V Frontend Std	#FS	4.2	ASYT Platform V Asynchronous Tasks	Нет	Сервис регистрации и выполнения периодических задач	Сервис успешно прошел испытания и подтвердил свою работоспособность с компонентом ASYT. С аналогами других производителей не тестировался
Platform V Frontend Std	#FS	4.2	OTTS One-Time	Нет	Сервис для аутентификации и	Сервис успешно прошел

Наименование продукта	Код	Версия продукта	Код и наименование компонента	Обязательность установки (да/нет) **	Описание	Аналог других производителей ***
			Password (OTP)/ОТТ		авторизации межсервисных взаимодействий	испытания и подтвердил свою работоспособность с компонентом OTTS. С аналогами других производителей ей не тестировался
Platform V Frontend Std	#FS	4.2	STDE ЕФС.Stand In	Нет	Сервис представляет собой технологический сервис, предназначенный для предоставления сервисов Топологии	Сервис успешно прошел испытания и подтвердил свою работоспособность с компонентом STDE. С аналогами других производителей ей не тестировался
Platform V Frontend Std	#FS	4.2	FPRT Защита от внутренних отказов ЕФС	Нет	Сервис платформы ЕФС, обеспечивающий предоставление информации о здоровье экземпляров сервисов для средств мониторинга и балансировки	Сервис успешно прошел испытания и подтвердил свою работоспособность с компонентом FPRT. С аналогами других производителей ей не тестировался
Platform V Synapse Service Mesh	SSM	2.10	IGEG Граничный прокси	Нет	Компонент для управления трафиком, приходящим и исходящим из проекта	Istio proxy 1.12

Наименование продукта	Код	Версия продукта	Код и наименование компонента	Обязательность установки (да/нет) **	Описание	Аналог других производителей ***
					одной системы	
			POLM Управление политиками	Нет	Панель управления с открытым исходным кодом, служащая для взаимодействия, мониторинга и обеспечения безопасности контейнеров в среде контейнеризации Kubernetes	Istio control plane 1.12

Примечание:

*** Применимо для обеспечения обратной совместимости.**

**

- **Да** — компонент или продукт необходим для функционирования сервиса (это означает, что сервис не может выполнять свои основные функции без установки данного компонента).
- **Нет** — необязательный для функционирования сервиса компонент или продукт (это означает, что сервис может выполнять свои основные функции без установки данного компонента).

*** Рекомендуется установка программного продукта, правообладателем которого является АО «СберТех», при этом не исключена возможность (допускается правообладателем) использования аналога других производителей. Аналоги, в отношении которых продукт успешно прошел испытания и подтвердил свою работоспособность, указаны в разделе «Системное программное обеспечение».

Аппаратные требования

Для проекта (namespace), в котором развертывается компонент, должны быть выданы следующие квоты по ресурсам:

- Ядра ЦПУ — 16
- Оперативная память — 32 GB

Установка

Установка компонента UFTM автоматизированным способом производится с помощью инструментов продукта Platform V DevOps Tools. Конфигурация компонента UFTM хранится в git-репозитории и мигрируется из дистрибутива компонента UFTM.

Порядок выполнения установки:

1. Настройка СУБД.
2. Миграция конфигурации компонента UFTM в репозиторий.
3. Настройка конфигурации компонента UFTM.
4. Установка дистрибутива компонента UFTM.

Настройка СУБД

Для настройки СУБД необходимо выполнить следующие действия в указанном порядке:

1. Создать схему БД.
2. Создать пользователя для взаимодействия со схемой БД из компонента UFTM.
3. Создать пользователя для выполнения liquibase-скриптов.
4. Создать табличные пространства.
5. Выдать права на изменение схемы пользователю для выполнения liquibase-скриптов.
6. Выдать полные права пользователю для взаимодействия со схемой БД из компонента UFTM на создаваемые пользователем для выполнения liquibase-скриптов объекты.

Параметры, используемые во время выполнения SQL-скриптов настройки БД

database_name_placeholder	= <имя БД>
schema_name_placeholder	= uftm_<суффикс_блока>
user_name_placeholder	= uftm_<суффикс_блока>
user_password_placeholder	= '<пароль пользователя БД>'
ts_data_placeholder	= uftm_ts_data
ts_data_location_placeholder	= '<путь до каталога табличного пространства>'
ts_idx_placeholder	= uftm_ts_idx
ts_idx_location_placeholder	= '<путь до каталога табличного пространства>'
liquibase_user_name_placeholder	= '<имя пользователя БД для выполнения liquibase-скриптов>'
liquibase_user_password_placeholder	= '<пароль пользователя БД для выполнения liquibase-скриптов>'

Создание схемы БД

```
CREATE SCHEMA :schema_name_placeholder;
```

Создание пользователя для взаимодействия со схемой БД из компонента UFTM

```
CREATE USER :user_name_placeholder WITH ENCRYPTED PASSWORD :user_password_placeholder;
ALTER USER :user_name_placeholder VALID UNTIL 'INFINITY';
ALTER USER :user_name_placeholder SET search_path = :schema_name_placeholder;
GRANT CONNECT ON DATABASE :database_name_placeholder TO :user_name_placeholder;
GRANT ALL ON SCHEMA :schema_name_placeholder TO :user_name_placeholder;
GRANT USAGE ON SCHEMA :schema_name_placeholder TO :user_name_placeholder;
```

Создание пользователя для выполнения liquibase-скриптов

```
CREATE USER :liquibase_user_name_placeholder WITH ENCRYPTED PASSWORD :liquibase_user_password_placeholder;
ALTER USER :liquibase_user_name_placeholder VALID UNTIL 'INFINITY';
ALTER USER :liquibase_user_name_placeholder SET search_path = :schema_name_placeholder;
GRANT CONNECT ON DATABASE :database_name_placeholder TO :liquibase_user_name_placeholder;
GRANT ALL ON SCHEMA :schema_name_placeholder TO :liquibase_user_name_placeholder;
GRANT USAGE ON SCHEMA :schema_name_placeholder TO :liquibase_user_name_placeholder;
```

Создание табличных пространств

Перед созданием табличных пространств необходимо создать каталоги, в которых будут располагаться файлы табличных пространств. Каталоги должны быть пустыми и принадлежать пользователю ОС, под которым запущен экземпляр СУБД.

```
CREATE TABLESPACE :ts_data_placeholder OWNER :liquibase_user_name_placeholder
LOCATION :ts_data_location_placeholder;
```

```
CREATE TABLESPACE :ts_idx_placeholder OWNER :liquibase_user_name_placeholder
LOCATION :ts_idx_location_placeholder;
```

Выдача прав на изменение схемы пользователю для выполнения liquibase-скриптов

Пользователь для выполнения liquibase-скриптов должен обладать всем набором прав на изменение созданной схемы БД. Пример выдачи прав на изменение схемы:

```
GRANT ALL PRIVILEGES ON SCHEMA :schema_name_placeholder TO :liquibase_user_name_placeholder;
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA :schema_name_placeholder TO :liquibase_user_name_placeholder;
GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA :schema_name_placeholder TO :liquibase_user_name_placeholder;
GRANT ALL PRIVILEGES ON ALL FUNCTIONS IN SCHEMA :schema_name_placeholder TO :liquibase_user_name_placeholder;
GRANT ALL PRIVILEGES ON ALL ROUTINES IN SCHEMA :schema_name_placeholder TO :liquibase_user_name_placeholder;
```

```
GRANT ALL PRIVILEGES ON ALL PROCEDURES IN SCHEMA :schema_name_placeholder TO :liquibase_user_name_placeholder;
```

Выдача полных прав пользователю для взаимодействия со схемой БД из компонента UFTM на создаваемые объекты БД

```
ALTER DEFAULT PRIVILEGES FOR USER :liquibase_user_name_placeholder IN SCHEMA :schema_name_placeholder GRANT ALL PRIVILEGES ON TABLES TO :user_name_placeholder;
```

```
ALTER DEFAULT PRIVILEGES FOR USER :liquibase_user_name_placeholder IN SCHEMA :schema_name_placeholder GRANT ALL PRIVILEGES ON SEQUENCES TO :user_name_placeholder;
```

```
ALTER DEFAULT PRIVILEGES FOR USER :liquibase_user_name_placeholder IN SCHEMA :schema_name_placeholder GRANT ALL PRIVILEGES ON FUNCTIONS TO :user_name_placeholder;
```

```
ALTER DEFAULT PRIVILEGES FOR USER :liquibase_user_name_placeholder IN SCHEMA :schema_name_placeholder GRANT ALL PRIVILEGES ON ROUTINES TO :user_name_placeholder;
```

Миграция конфигурации компонента UFTM в репозиторий

Для миграции конфигурации компонента UFTM в репозиторий необходимо в Deploy job выполнить следующее:

1. Нажать кнопку **Собрать с параметрами**.
2. Указать значения для параметров:
 1. CONFIG_DIR
 2. SUBSYSTEM
 3. COMPONENTS
 4. SECTOR
 5. OSE_CLUSTERS
 6. DISTRIB_VERSION
 7. PARAMS:
 1. В секции «Репозиторий с настройками ФП» выбрать ветку в репозитории с конфигурацией компонента UFTM. При необходимости создайте ветку в репозитории с конфигурацией компонента UFTM, руководствуясь документацией продукта Platform V DevOps Tools.
 2. Выбрать плейбук MIGRATION_FP_CONF.
3. Нажать кнопку **Собрать**.

Назначение плейбуков описано в документации продукта Platform V DevOps Tools.

Настройка конфигурации компонента UFTM

В разделе описана настройка минимального набора параметров, необходимого для работы компонента UFTM. Полный список доступных для настройки параметров приведен в разделе «Конфигурирование» документа [«Руководство по системному администрированию»](#).

Настройка интеграции с СУБД

Руководствуясь документацией продукта Platform V DevOps Tools необходимо заполнить следующие параметры в репозитории с конфигурацией стенда (common-репозитории):

Название	Описание	Значение	Конфигурационный файл
<code>jdbc.uftm_postgres.url</code>	URL подключения к БД	URL подключения к БД	<code>_global.jdbc.conf</code>
<code>jdbc.liquibase_postgres.user</code>	Имя пользователя БД для выполнения liquibase-скриптов	Имя пользователя БД, который обладает всем набором прав на изменение схемы БД	<code>_passwords.conf</code>
<code>jdbc.liquibase_postgres.password</code>	Пароль пользователя БД, имя которого указано в параметре <code>jdbc.liquibase_postgres.user</code>	Пароль пользователя БД, который обладает всем набором прав на изменение схемы БД	<code>_passwords.conf</code>
<code>DB_SCHEMA_SUFFIX</code>	Суффикс имени схемы БД	Суффикс имени схемы БД, например, имя блока или контура, на который устанавливается компонент UFTM	<code>common.conf.yml</code>
<code>jdbc.uftm_postgres.user</code>	Имя пользователя БД для взаимодействия со схемой БД из компонента UFTM	Имя пользователя БД, который обладает набором прав для взаимодействия со схемой БД	<code>_passwords.conf</code>
<code>jdbc.uftm_postgres.password</code>	Пароль пользователя БД, имя которого указано в параметре <code>jdbc.uftm_postgres.user</code>	Пароль пользователя БД, который обладает набором прав для взаимодействия	<code>_passwords.conf</code>

Название	Описание	Значение	Конфигурационный файл
		ия со схемой БД	

Настройка интеграции с Компонентом OTT

Настройка сертификатов

1. Сгенерировать ключевую пару.
2. Создать запрос на выпуск сертификата с именем субъекта, равного значению параметра `ufs-dynamic-content.ose.egress.ott.module.id`
3. Отправить запрос на выпуск сертификата в удостоверяющий центр.
4. Импортировать в хранилище сертификатов, путь к которому указан в параметре `ssl.ose.istio.keyStore.egress.KeyStoreFromFile`:
 1. сертификат удостоверяющего центра.
 2. выпущенный сертификат с `alias`, определяемым параметром `ufs-dynamic-content.ose.egress.ott.client.cert.alias`.

Порядок выпуска и импорта сертификатов в хранилище описан в разделе «Управление ключами и сертификатами» документа [«Руководство по системному администрированию»](#).

Настройка интеграции с брокером Kafka

Создание topic

Создать в Kafka следующие topic:

Название	Описание	Параметр с названием topic	Конфигурационный файл, в котором описан параметр
<code>ufs-ul-sbercms-content-data</code>	Загрузка пакетов изменений шаблонов из мастер-системы	<code>ufs-dynamic-content-upload-service.kafka.upload-topic-name</code>	<code>ufsdyncontent-upload-service.conf</code>
<code>ufs-sbercms-content-acknowledgement</code>	Уведомление мастер-системы о статусе обработки пакета изменений шаблонов	<code>ufs-dynamic-content-upload-service.kafka.notification-topic-name</code>	<code>ufsdyncontent-upload-service.conf</code>

Настройка сертификатов

1. Сгенерировать ключевую пару.
2. Создать запрос на выпуск сертификата с именем субъекта, равного значению параметра `ufs-dynamic-content.ose.egress.ott.module.id`.
3. Отправить запрос на выпуск сертификата в удостоверяющий центр.
4. Импортировать в хранилище сертификатов, путь к которому указан в параметре `ufs-dynamic-content-upload-service.kafka.ssl.keyStore.filePath`:
 1. сертификат удостоверяющего центра.

2. выпущенный сертификат с alias, определяемым параметром `ufs-dynamic-content-upload-service.kafka.ssl.keyStore.certAliases`.

Порядок выпуска и импорта сертификатов в хранилище описан в разделе «Управление ключами и сертификатами» документа [«Руководство по системному администрированию»](#).

Настройка параметров

В репозитории с конфигурацией компонента UFTM необходимо заполнить следующие параметры:

Название	Описание	Значение	Конфигурационный файл
<code>ufs-dynamic-content-upload-service.kafka.bootstrap.servers</code>	Адреса серверов Kafka внешней CMS	Адреса серверов Kafka, перечисленные через запятую	<code>ufsdyncontent-upload-service.conf</code>
<code>ufs-dynamic-content-upload-service.ose.istio.egress.se.tcp.cms-kafka-port.number</code>	Порт серверов Kafka внешней CMS	Порт серверов Kafka, перечисленных в параметре <code>ufs-dynamic-content-upload-service.kafka.bootstrap.servers</code>	<code>ufsdyncontent-upload-service.istio.conf</code>

Установка дистрибутива компонента

Для установки дистрибутива компонента UFTM необходимо в Deploy job выполнить следующее:

1. Нажать кнопку **Собрать с параметрами**.
2. Указать значения для параметров:
 1. CONFIG_DIR
 2. SUBSYSTEM
 3. COMPONENTS
 4. SECTOR
 5. OSE_CLUSTERS
 6. DISTRIB_VERSION
 7. PARAMS:
 1. В секции «Репозиторий с настройками ФП» выбрать ветку в репозитории с конфигурацией компонента UFTM.
 2. Выбрать следующие плейбуки:
 1. DB_UPDATE
 2. NGINX_MM_DEPLOY
 3. FP_CONF_CHECK
 4. OPENSIFT_DEPLOY
 5. OPENSIFT_INGRESS_EGRESS_DEPLOY
 6. IMPORT_SUP_PARAMS
 7. IMPORT_LOGGER_PARAMS

8. IMPORT_TASK_SCHEDULER_PARAMS.

3. Нажать кнопку **Собрать**.

Назначение плейбуков описано в документации продукта Platform V DevOps Tools.

Обновление

1. Информирование о выпуске обновлений и поставка обновлений производится в рамках единого релизного цикла фронтальной части Платформы.
2. Необходимые настройки сервиса будут осуществлены при установке дистрибутива компонента UFTM через Platform V DevOps Tools. Дополнительных настроек не требуется.
3. Обновление представляет собой удаление предыдущей версии релиза компонента UFTM и установку изменений, внесенных в рамках новой версии релиза.

Удаление

Удаление компонента UFTM представляет собой удаление всех объектов и ресурсов в пространстве имен (namespace), где установлен компонент, а также самого пространства имен в среде контейнеризации.

Проверка работоспособности

1. Проверка публикации нового шаблона через Kafka

Шаг	Действие	Ожидаемый результат
1	Отправить сообщение в топик ufs-ul-sbercms-content-data. Поле "parentRequestId" в теле запроса не обязательно, заполняется сервисом из заголовка parent_request_id перед загрузкой в БД. Где для параметра changeType необходимо указать значение PUBLISH. Пример запроса указан ниже	В KafkaTool отображается корректное сообщение
2	Через некоторое время (значение по умолчанию 5 мин) проверить в топике ufs-sbercms-content-acknowledgement наличие сообщения об обработке запроса, указанного в параметре Компонента СУП dynamic-content.scheduled.change-packages-processing	В KafkaTool отображается сообщение со статусом PROCESSED, не содержащее ошибок
3	Отправить POST-запрос на endpoint сервиса выдачи шаблонов: "protocol://host:port/ufs-dynamic-content-service/v1/templates/search-by-cms-template-id", содержащий в теле запроса ["cdb70d6c-e7f4-4538-a29b-c47d46206a70"]. Где идентификатор шаблона должен иметь значение из параметра template.id, отправленного сообщения на шаге 1	В ответе ожидается корректный шаблон, значения полей которого совпадают с отправленным сообщением на шаге 1

```
{  
  "totalChunkAmount": 1,  
  "currentChunkNumber": 1,  
  "correlationId": "edcc7f00-035e-4746-8eba-fbf96a38874b",  
  "packageCreationDate": 1637330099,  
  "changes": [  
    {  
      "changeType": "PUBLISH",
```



```

"template": {
  "id": "cdb70d6c-e7f4-4538-a29b-c47d46206a70",
  "version": 1,
  "templateType": "CORP_LETTER_DISIGN",
  "locale": "ru-RU",
  "isDefault": false,
  "body": "BPJFQcY1AJSQb7utnXjZgsEprAunTWPeHp37XmxT58iUrB8Cw810rtwEyHWY
xWiDts74LkSHtKwKyBBPwiFZUoJD8oRMI5JnKHIXKZ6npS8QQxadjSaKZ3E11FF0jkxNHsJKk28Sg
MXc3pmF0FAZAM",
  "context": {
    "subsystemCode": "CORPLETTER",
    "tenantCode": "SBBOL",
    "tags": [
      "MASS_COMM_PRODUCT_CHANGE"
    ],
    "clients": [
      {
        "code": "WEB",
        "versionRange": {
          "fromVersion": {
            "major": 1,
            "minor": 0,
            "patch": 0
          },
          "toVersion": {
            "major": 1,
            "minor": 0,
            "patch": 1
          }
        }
      }
    ]
  }
}

```

2. Проверка изменения шаблона через Kafka

Шаг	Действие	Ожидаемый результат
1	Отправить сообщение в топик <code>ufs-ul-sbercms-content-data</code> с измененным телом шаблона. Поле parentRequestId в теле запроса не обязательно, заполняется сервисом из заголовка <code>parent_request_id</code> перед загрузкой в БД. Где для параметра <code>changeType</code> необходимо указать значение <code>PUBLISH</code> . Пример запроса указан ниже	В KafkaTool отображается корректное сообщение
2	Через некоторое время (значение по умолчанию 5 мин) проверить в топике <code>ufs-sbercms-content-acknowledgement</code> наличие сообщения об обработке запроса,	В KafkaTool отображается сообщение со статусом

Шаг	Действие	Ожидаемый результат
	указанного в параметре Компонента СУП <code>dynamic-content.scheduled.change-packages-processing</code>	PROCESSED, не содержащее ошибок
3	Отправить POST-запрос на endpoint сервиса выдачи шаблонов: <code>protocol://host:port/ufs-dynamic-content-service/v1/templates/search-by-cms-template-id</code> , содержащий в теле запроса <code>["cdb70d6c-e7f4-4538-a29b-c47d46206a70"]</code> . Где идентификатор шаблона должен иметь значение из параметра <code>template.id</code> , отправленного сообщения на шаге 1	В ответе ожидается корректный шаблон, которого совпадают с отправленным сообщением на шаге 1

```

{
  "totalChunkAmount": 1,
  "currentChunkNumber": 1,
  "correlationId": "edcc7f00-035e-4746-8eba-fbf96a38874b",
  "packageCreationDate": 1637330099,
  "changes": [
    {
      "changeType": "PUBLISH",
      "template": {
        "id": "cdb70d6c-e7f4-4538-a29b-c47d46206a70",
        "version": 1,
        "templateType": "CORP_LETTER_DISIGN",
        "locale": "ru-RU",
        "isDefault": false,
        "body": "EditedBody",
        "context": {
          "subsystemCode": "CORPLETTER",
          "tenantCode": "SBBOL",
          "tags": [
            "MASS_COMM_PRODUCT_CHANGE"
          ],
          "clients": [
            {
              "code": "WEB",
              "versionRange": {
                "fromVersion": {
                  "major": 1,
                  "minor": 0,
                  "patch": 0
                },
                "toVersion": {
                  "major": 1,
                  "minor": 0,
                  "patch": 1
                }
              }
            }
          ]
        }
      }
    }
  ]
}

```

```

}
]
}

```

3. Проверка деактивации шаблона через Kafka

Шаг	Действие	Ожидаемый результат
1	Отправить сообщение в топик <code>ufs-ul-sbercms-content-data</code> с измененным телом шаблона. Поле <code>parentRequestId</code> в теле запроса не обязательно, заполняется сервисом из заголовка <code>parent_request_id</code> перед загрузкой в БД. Где для параметра <code>changeType</code> необходимо указать значение <code>UNPUBLISH</code> . Пример запроса указан ниже	В KafkaTool отображается корректное сообщение
2	Через некоторое время (значение по умолчанию 5 мин) проверить в топике <code>ufs-sbercms-content-acknowledgement</code> наличие сообщения об обработке запроса, указанного в параметре Компонента СУП <code>dynamic-content.scheduled.change-packages-processing</code>	В KafkaTool отображается сообщение со статусом <code>PROCESSED</code> , не содержащее ошибок
3	Отправить POST-запрос на endpoint сервиса выдачи шаблонов: <code>protocol://host:port/ufs-dynamic-content-service/v1/templates/search-by-cms-template-id</code> , содержащий в теле запроса <code>["cdb70d6c-e7f4-4538-a29b-c47d46206a70"]</code> . Где идентификатор шаблона должен иметь значение из параметра <code>template.id</code> , отправленного сообщения на шаге 1	В ответе ожидаем корректный ответ с пустым списком шаблонов

```

{
  "totalChunkAmount": 1,
  "currentChunkNumber": 1,
  "correlationId": "edcc7f00-035e-4746-8eba-fbf96a38874b",
  "packageCreationDate": 1637330099,
  "changes": [
    {
      "changeType": "UNPUBLISH",
      "template": {
        "id": "cdb70d6c-e7f4-4538-a29b-c47d46206a70",
        "version": 1,
        "templateType": "CORP_LETTER_DISIGN",
        "locale": "ru-RU",
        "isDefault": false,
        "body": "EditedBody",
        "context": {
          "subsystemCode": "CORPLETTER",
          "tenantCode": "SBBOL",
          "tags": [
            "MASS_COMM_PRODUCT_CHANGE"
          ],
          "clients": [
            {
              "code": "WEB",

```

```

    "versionRange": {
      "fromVersion": {
        "major": 1,
        "minor": 0,
        "patch": 0
      },
      "toVersion": {
        "major": 1,
        "minor": 0,
        "patch": 1
      }
    }
  }
]
}

```

4. Проверка публикации некорректного шаблона через Kafka

Шаг	Действие	Ожидаемый результат
1	Отправить сообщение в топик <code>ufs-ul-sbercms-content-data</code> с измененным телом шаблона. Поле <code>parentRequestId</code> в теле запроса не обязательно, заполняется сервисом из заголовка <code>parent_request_id</code> перед загрузкой в БД. Пример запроса указан ниже	В KafkaTool отображается корректное сообщение
2	Через некоторое время (значение по умолчанию 5 мин) проверить в топике <code>ufs-sbercms-content-acknowledgement</code> наличие сообщения об обработке запроса, указанного в параметре Компонента СУП <code>dynamic-content.scheduled.change-packages-processing</code>	В KafkaTool отображается сообщение со статусом <code>FAILED</code> , содержащее ошибки
3	Отправить POST-запрос на endpoint сервиса выдачи шаблонов: <code>protocol://host:port/ufs-dynamic-content-service/v1/templates/search-by-cms-template-id</code> , содержащий в теле запроса <code>["cdb70d6c-e7f4-4538-a29b-c47d46206a70"]</code> . Где идентификатор шаблона должен иметь значение из параметра <code>template.id</code> , отправленного сообщения на шаге 1	В ответе ожидаем корректный ответ с пустым списком шаблонов

```

{
  "totalChunkAmount": 1,
  "currentChunkNumber": 1,
  "correlationId": "92c974cb-fac7-4e55-b770-61ed68a65c44",
  "changes": [
    {
      "changeType": "PUBLISH",
      "template": {
        "id": "e5fc6c04-b2a2-4f20-b113-f3ee0c6b0c29",
        "version": 1,

```

```

    "isDefault": false,
    "actualityPeriod": {},
    "body": "pE2PNrxijMauZeakrfA8tnfKBhZkt77Jg3NhcaD4l3kYZF6ZnPY1ful9VjzT
RXxb3zADSHvoFKJYxVJEj1QAKvqwnQ1sFPE7gJNkpXGq70Ppf9gAyRIHGomyfKXXQL8WCyJ93g1osT
6GSyRFthdn4Wr"
  }
}
]
}

```

5. Переключение уровня логирования событий

Шаг	Действие	Ожидаемый результат	Тестовые данные
1	Убедиться в наличии параметров журналирования для подсистемы DYNAMIC_CONTENT в АРМ Компонента Журналирование, вкладка Администрирование параметров	В разделе Активные параметры присутствуют: logger.local.enabled logger.local.level logger.local.stdout.appender.enabled logger.remote.enabled logger.remote.level logger.name.level.	logger.local.level = ERROR logger.remote.level = ERROR
2	Проверить переключение уровня логирования в локальном логге сервиса, изменив значения параметров	В консоли инструмента управления контейнеризованными приложениями (Kubernetes или OpenShift (опционально)) во вкладке Logs появляются записи с уровнями DEBUG/INFO/WARN/ERROR	logger.local.enabled = true logger.local.level = DEBUG logger.local.stdout.appender.enabled = true
3	Вернуть значение на уровень ERROR	В консоли инструмента управления контейнеризованными приложениями (Kubernetes или OpenShift (опционально)) во вкладке Logs появляются записи только с уровнем ERROR	logger.local.level = ERROR
4	Проверить переключение уровня логирования в Компоненте Журналирование, изменив значения параметров	В АРМ Компонента Журналирование во вкладке Таблица появляются записи с уровнями DEBUG/INFO/WARN/ERROR	logger.remote.enabled = true logger.remote.level = DEBUG
5	Вернуть значение на уровень ERROR	В АРМ Компонента Журналирование во вкладке Таблица появляются записи только с уровнем ERROR	logger.remote.level = ERROR

6. Применение параметров Компонента СУП без перезапуска приложения

Шаг	Действие	Ожидаемый результат	Тестовые данные
1	В АРМ Компонента Журналирование установить уровень логирования DEBUG	Значения параметров изменились	logger.local.enabled = true logger.local.level = DEBUG logger.local.stdout.appende r.enabled = true
2	Проверить переключение уровня логирования в локальном логе сервиса	В консоли инструмента управления контейнеризированными приложениями (Kubernetes или OpenShift (опционально)) во вкладке Logs появляются записи с уровнями DEBUG/INFO/WARN/ERROR	-
3	В АРМ Компонента Мониторинг для ear-файла DYNAMIC_CONTENT включить сбор метрик «Карта сервисов для ufs-dynamic-content-service»	Для необходимых метрик Фильтр вашей роли находится в состоянии Выбран (стоит галочка).	Метрика: Карта сервисов -> Использование параметра Компонента СУП -> SERVICE_MAP/PARAM_READ_REQUEST
4	В АРМ Компонента СУП изменить значение параметра <code>ufsparams.module.monitoring.enabled</code> для конфигурации UFS-DYNAMIC-CONTENT_SERVICE - Включение отправки метрики мониторинга при запросе параметра в Компоненте СУП. (true - метрики отправляются, false - метрики не отправляются)	Для параметра <code>ufsparams.module.monitoring.enabled</code> установлено значение true.	ufsparams.module.monitoring.enabled = true
5	Проверить в логах сервиса, что значение параметра изменилось	В консоли инструмента управления контейнеризированными приложениями (Kubernetes или OpenShift (опционально)) во вкладке Logs появляются записи с новым значением параметра	-
6	В АРМ Компонента Мониторинг проверить появление метрики	На графике видны всплески со значениями	Метрика: Карта сервисов -> Использование параметра Компонента СУП -> SERVICE_MAP/PARAM_READ_REQUEST
7	В АРМ Компонента СУП изменить значение параметра <code>ufsparams.module.monitoring.enabled</code> для	Для параметра <code>ufsparams.module.monitoring.enabled</code>	ufsparams.module.monitoring.enabled = false

Шаг	Действие	Ожидаемый результат	Тестовые данные
	конфигурации UFS-DYNAMIC-CONTENT_SERVICE - Включение отправки метрики мониторинга при запросе параметра в Компоненте СУП (true - метрики отправляются, false - метрики не отправляются)	ing.enabled установлено значение false	
8	В АРМ Компонента Журналирование установить уровень логирования ERROR	Значения параметров изменились	logger.local.enabled = true logger.local.level = ERROR logger.local.stdout.appende r.enabled = true

7. Фильтрация шаблонов по локали и типу и версии МП через REST-запрос

Шаг	Действие	Ожидаемый результат	Тестовые данные
1	Отправить POST-запрос на endpoint protocol://host:port/ufs-dynamic-content-service/v1/templates	В ответе ожидаем список шаблонов с указанными в теле запроса "locale" и "clientType"	Тело запроса представлено ниже

```
{
  "subsystemCode": "CREDIT_CARD",
  "locale": "ru",
  "clientType": "Android",
  "templateTypes": [
    "CARD"
  ]
}
```

8. Фильтрация шаблонов по подсистеме и типу и версии МП через REST-запрос

Шаг	Действие	Ожидаемый результат	Тестовые данные
1	Отправить POST-запрос на endpoint protocol://host:port/ufs-dynamic-content-service/v1/templates	В ответе ожидаем список шаблонов с указанными в теле запроса "subsystemCode" и "clientType"	Тело запроса представлено ниже

```
{
  "subsystemCode": "CREDIT_CARD",
  "locale": "ru",
  "clientType": "Android",
  "templateTypes": [
    "CARD"
  ]
}
```

9. Возврат fallback-данных (нулевой результат фильтрации) через REST-запрос

Шаг	Действие	Ожидаемый результат	Тестовые данные
1	Отправить POST-запрос на endpoint protocol://host:port/ufs-dynamic-content-service/v1/templates	{ "success": true, "body": { "templates": { } } }	Тело запроса представлено ниже

```
{
  "subsystemCode": "CREDIT_CARD",
  "locale": "locale",
  "clientType": "Android",
  "templateTypes": [
    "CARD"
  ]
}
```

10. Логирующие изменения

Шаг	Действие	Ожидаемый результат	Тестовые данные
1	Убедиться в наличии параметров журналирования для подсистемы DYNAMIC_CONTENT в АРМ Компонента Журналирование, вкладка Администрирование параметров	В разделе Активные параметры присутствуют: logger.local.enabled, logger.local.level, logger.local.stdout.appender.enabled, logger.remote.enabled, logger.remote.level, logger.name.level. Значения по умолчанию: logger.local.level = ERROR, logger.remote.level = ERROR	-
2	Выполнить загрузку шаблона с последующим запросом контента от сервиса выдачи шаблонов	Шаблон успешно загружен, сервис выдачи шаблонов корректно выдает контент	Для логирования действий необходим уровень DEBUG. logger.local.stdout.appender.enabled = true, logger.local.level = DEBUG, logger.remote.level = DEBUG
3	Проверить наличие записей в локальных логах сервисов	В консоли инструмента управления контейнеризированными приложениями (Kubernetes или OpenShift (опционально)) во вкладках Logs для сервисов появляются записи, соответствующие уровню логирования, указанному в параметрах. Есть информация о выполненной загрузке и выдаче контента	-
4	Проверить наличие записей в АРМ Компонента Журналирование	В АРМ Компонента Журналирование во вкладке Таблица появляются записи, соответствующие уровню логирования, указанному в	-

Шаг	Действие	Ожидаемый результат	Тестовые данные
		параметрах. Есть информация о выполненной загрузке и выдаче контента	

11. Мониторинг событий

Шаг	Действие	Ожидаемый результат	Тестовые данные
1	В АРМ Компонента Мониторинг для ear-файла DYNAMIC_CONTENT включить сбор прикладных метрик для сервисов «ufs-dynamic-content-service, ufs-dynamic-content-upload-service»	Для необходимых метрик Фильтр роли находится в состоянии Выбран (стоит галочка)	-
2	Выполнить загрузку шаблона с последующим запросом контента от сервиса выдачи шаблонов	Шаблон успешно загружен, сервис выдачи шаблонов корректно выдает контент	-
3	В АРМ Компонента Мониторинг для ufs-dynamic-content-upload-service проверяем наличие метрики	Метрика присутствует на графике	APPLICATION/CHANGE_PACKAGE_PERSISTED
4	В АРМ Компонента Мониторинг для ufs-dynamic-content-service проверяем наличие метрик	Метрики присутствуют на графиках	APPLICATION/CONTENT_REQUEST_IS_RECEIVED, APPLICATION/CONTENT_REQUEST_IS_SUCCESSFULLY_FINISHED

12. Проверка доступности сервиса

Шаг	Действие	Ожидаемый результат	Тестовые данные
1	Отправить GET-запрос на endpoint protocol://host:port/ufs-dynamic-content-service/healthcheck	Получен ответ: { "success": true, "body": "OK" }	-

13. Запрос шаблонов по идентификаторам шаблонов в CMS

Шаг	Действие	Ожидаемый результат	Тестовые данные
1	Отправить POST-запрос на endpoint: "protocol://host:port/ufs-dynamic-content-service/v1/templates/search-by-cms-template-id", содержащий в теле запроса ["18f49b0b-167f-459c-b479-2286f46a1c11"]	В ответе ожидаем список шаблонов с указанными в теле запроса идентификаторами	-

14. Проверка ОТТ

Шаг	Действие	Ожидаемый результат	Тестовые данные
1	Проверить в логах ottsidecar-egress-a, что token замаскированы (допустимы повторяющиеся записи для токена с полным отображением)	INFO [fbt-refresh-thread] c.s.o.b.a.c.i.t.RootTokenTask - Вернулся токен с сервера ОТТ ; INFO [fbt-refresh-thread] c.s.o.b.a.c.i.TokenService - Токен для вызова [JwsTokenCreateReq{tokenClaims=JwsTokenClaims(invoker=ci02649924_as_efs_ufsdyncontent_emp,receiver=null,url=null,urlEvent=null, domain=ott, encKeyParam=null,action=null,anonymous=null, tokenType=fallback,reqId=0f8db145-d539-4697-8c31-fad2584b93a7),namedAttrs={Attributes=[ott:appurn:oasis:names:tc:xacml:1.0:subject-category:access-subjecthttp://www.w3.org/2001/XMLSchema#string[ci02649924_as_efs_ufsdyncontent_emp],ott:token:typeurn:oasis:names:tc:xacml:3.0:attribute-category:environmenthttp://www.w3.org/2001/XMLSchema#string[fallback],ott:realmurn:oasis:names:tc:xacml:3.0:attribute-category:environmenthttp://www.w3.org/2001/XMLSchema#string[ott]]}}] успешно получен. Ответ от сервера ОТТ - [TokenData[token='eyJhbGciOiJIUzI1NiJ9....truncated.ngGw1_6cnpfCO6Z57pI7uUUaYQ762S2ONqqpV3wnAK6txDvYt_mx_Hh-6mI_fxD9hWsB1bAE7mKtRpJzRbC0A';implCode='fallback';reqId='0f8db145-d539-4697-8c31-fad2584b93a7']]	-
2	Убедиться, что в АРМ Компонента Журналирование на вкладке «Администрирование параметров» присутствуют параметры	В разделе Активные параметры присутствуют: logger.local.enabled logger.local.level logger.local.stdout.appender.enabled logger.remote.enabled logger.remote.level logger.name.level Значения по умолчанию: logger.remote.level = ERROR	-

Шаг	Действие	Ожидаемый результат	Тестовые данные
	журналирования для подсистемы DYNAMIC_CONTENT		
3	Переключить уровень логирования в АРМ Компонента Журналирование, изменив значения параметров	В АРМ Компонента Журналирование во вкладке Таблица появляются записи с уровнями DEBUG/INFO/WARN/ERROR.	logger.remote.enabled = true logger.remote.level = DEBUG
4	Отправить POST-запрос на эндпоинт: "protocol://host:port/ufs-dynamic-content-service/v1/templates/search-by-cms-template-id", с указанием в теле запроса несуществующих идентификаторов ["11111111-1111-1111-1111-111111111111"]	Получен ответ: {"success": true, "body": {"templates": {}}}	protocol = https host = nginx_mm port = 9443
5	В АРМ Компонента Журналирование, во вкладке «Таблица», найти запись о запросе и убедиться, что незамаскированный	Искомая запись найдена, незамаскированный токен отсутствует	-

Шаг	Действие	Ожидаемый результат	Тестовые данные
	токен отсутствует		
6	Вернуть первоначальное значение параметров в Компоненте Журналирование	В АРМ Компонента Журналирование во вкладке «Таблица» появляются записи, соответствующие установленному уровню	-

15. Аудит изменений

Шаг	Действие	Ожидаемый результат	Тестовые данные
1	Выполнить загрузку шаблона с последующим запросом контента от сервиса выдачи шаблонов	Шаблон успешно загружен, сервис выдачи шаблонов корректно выдает контент	-
2	Проверить наличие событий в АРМ Компонента Аудит	<p>Найдены события:</p> <p>SAVE_TEMPLATE_CHANGE_PACKAGE_CHUNK: PACKAGE_ID - Идентификатор пакета изменений шаблонов в мастер-системе SUBSYSTEM_CODE - Код подсистемы-владельца пакета изменений шаблонов CURRENT_CHUNK_NUMBER - Номер текущего чанка пакета изменений шаблонов</p> <p>APPLY_TEMPLATE_CHANGE_PACKAGE: PACKAGE_ID - Идентификатор пакета изменений шаблонов в мастер-системе SUBSYSTEM_CODE - Код подсистемы-владельца пакета изменений шаблонов</p>	-

Откат

Откат к предыдущей версии компонента UFTM представляет собой удаление изменений, внесенных в рамках установленной версии:

1. Выполнить установку предыдущей версии компонента UFTM. В проекте Kubernetes или Openshift (опционально) из deploy конфигураций удалить объекты «ufs-dynamic-content-service-» и «ufs-dynamic-content-upload-service-<номер релиза>», где <номер релиза> — номер релиза, в который устанавливался продукт (например, «r201»).
2. После установки дистрибутива компонента UFTM предыдущей версии необходимо проверить правильность функционирования приложения в соответствии с разделом «Использование программного продукта», документа «Руководство прикладного разработчика»

Примечание: Откат изменений в настройках параметров осуществляется в Deploy Tools и [Компоненте СУП](#) вручную. Для получения детальной информации о работе с настройкой параметров обратитесь к документации на данные продукты.

Часто встречающиеся проблемы и пути их устранения

У сервиса нет известных проблем.

Чек-лист валидации установки

Среда развертывания обеспечивает наличие:

- Инфраструктура: Kubernetes (k8s) или опционально Openshift Enterprise (OSE).
- Кластер Kafka для внешнего взаимодействия (Kafka внешней CMS).
- СУБД PostgreSQL или Platform V Pangolin SE.

На площадке развертывания установлены следующие инфраструктурные компоненты фронтальной части Платформы:

- Компонент Аудит.
- Компонент Журналирование.
- Компонент Мониторинг.
- Компонент САЗ.
- Компонент СУП.
- Компонент ОТТ.
- Компонент Управляющий агент.
- Компонент ЕФС.Stand In.

Успешно пройдены тест кейсы раздела «Проверка работоспособности».

В процессе установки выполнены следующие действия:

- Создана схема БД с именем `uftm_<суффикс_блока>`.
- Установлен дистрибутив Platform V Content Management — Сервис шаблонов форм UI.
- (Опционально) Применены параметры подключения к указанным выше инфраструктурным компонентам фронтальной части Платформы.