

# Настройка контейнерных компонентов

Версия 8.0



Эта документация предоставляется с ограничениями на использование и защищена законами об интеллектуальной собственности. За исключением случаев, прямо разрешенных в вашем лицензионном соглашении или разрешенных законом, вы не можете использовать, копировать, воспроизводить, переводить, транслировать, изменять, лицензировать, передавать, распространять, демонстрировать, выполнять, публиковать или отображать любую часть в любой форме или посредством любые значения. Обратный инжиниринг, дизассемблирование или декомпиляция этой документации, если это не требуется по закону для взаимодействия, запрещены.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления и не может гарантировать отсутствие ошибок. Если вы обнаружите какие-либо ошибки, сообщите нам о них в письменной форме.

# Содержание

<b>Настроить глобальный поиск</b>	<b>4</b>
Настроить сервис глобального поиска (версия 3.0)	4
<b>Настроить массовый поиск дублей</b>	<b>18</b>
Компоненты сервиса массового поиска дублей	18
Настроить переменные окружения	19
Запустить контейнеры	19
Проверить успешность запуска контейнеров	20
Проверить логирование	20
Подключить функциональность массового поиска дублей в Creatio	20
Обновить версию массового поиска дублей	25
Рекомендуемые работы по обслуживанию сервиса	25
<b>Часто задаваемые вопросы по глобальному поиску и дедупликации</b>	<b>26</b>
Как узнать, какая версия глобального поиска используется?	26
Как запустить переиндексацию сайта?	26
Как включить логирование работы сервиса глобального поиска?	26
Какие метрики или системы слежения можно использовать для мониторинга работы глобального поиска?	27
Как настроить доступ к Elasticsearch по паролю?	27
Как добавить в индекс новый объект или изменить настройки индексируемых полей существующих объектов?	27
Как развернуть Elasticsearch на нескольких серверах с указанием одного URL? Как настроить кластеризацию?	29
Что делать, если не отображается настройка “Правила поиска дублей”?	29
Можно ли использовать сервисы глобального поиска и массового поиска дублей одновременно с двумя приложениями Creatio?	29
Как сортируются результаты поиска?	29
Поддерживается ли аутентификация Windows в микросервисах?	30
<b>Сервис машинного обучения</b>	<b>30</b>
Подготовка к установке сервиса машинного обучения	30
Установить сервис машинного обучения	32
Обновить компоненты сервиса машинного обучения	34
<b>Настроить сервис синхронизации Exchange Listener</b>	<b>34</b>
Определить конфигурацию сервиса Exchange Listener	35
Способы развертывания сервиса Exchange Listener	36
Настроить сервис Exchange Listener на стороне Creatio	41
Диагностика настроек Exchange Listener	42

# Настроить глобальный поиск

ПРОДУКТЫ: **ВСЕ ПРОДУКТЫ**

Сервис глобального поиска (Global Search Service) создан для интеграции ElasticSearch с Creatio и выполняет следующие функции:

- **Регистрирующую:**
  - Подписывает клиента, создавая индекс в ElasticSearch и сохраняет связь индекс-приложение.
  - Отключает клиента, по требованию удаляя индекс в ElasticSearch.
- **Транспортную:**
  - Участвует в процессе индексации — забирает данные из базы данных.

Для настройки глобального поиска необходимы два отдельных физических или виртуальных сервера (“сервер 1” и “сервер 2”) с установленной ОС Linux. Для расчета требований к серверам воспользуйтесь [калькулятором системных требований](#).

Развертывание компонентов глобального поиска выполняется на ПО Docker. С перечнем поддерживаемых ОС вы можете ознакомиться в [документации Docker](#).

**На заметку.** В зависимости от потребностей вашей компании можно использовать Docker Community Edition (CE) или Enterprise Edition (EE). Подробную информацию вы найдете в [документации Docker](#).

**Важно.** Для настройки глобального поиска необходимы базовые знания администрирования ПО Docker и ОС Linux.

Последовательность действий при настройке глобального поиска зависит от того, какую версию сервиса вы будете использовать. Для последней версии Creatio рекомендуется всегда использовать новейшую версию сервиса глобального поиска.

Для поддержания работоспособности сервиса и возможности быстрого восстановления данных после отказа, например, в случае перебоев с электроэнергией, рекомендуется раз в сутки выполнять резервное копирование Elasticsearch.

Если у вас возникнут вопросы в ходе настройки, то рекомендуем ознакомиться со статьей [Часто задаваемые вопросы по глобальному поиску и дедупликации](#).

## Настроить сервис глобального поиска (версия 3.0)

**На заметку.** Настройки ниже актуальны для сервиса глобального поиска версии 3.0. Если вам необходимо настроить сервис глобального поиска более ранней версии, то ознакомьтесь с

описанием настроек в [документации Creatio версии 7.16](#).

Чтобы **обновить** глобальный поиск с версии 2.0 на версию 3.0, необходимо удалить все docker volume версии 2.0 на серверах 1 и 2 при помощи команды [docker-compose down -v](#) и повторно выполнить установку и настройку всех сервисов.

## Компоненты глобального поиска

Развертывается на сервере 1:

- [elasticsearch](#) — поисковый движок.

Развертываются на сервере 2:

- [postgres](#) — база данных конфигурирования компонентов глобального поиска.
- [rabbitmq](#) — брокер сообщений.
- [redis](#) — хранилище данных, используемое для кеширования и быстрого действия.
- [gs-web-api](#) — web-сервис конфигурирования компонентов глобального поиска.
- [gs-web-indexing-service](#) — web-сервис для обработки запросов точечного индексирования данных из системы.
- [gs-search-service](#) — web-сервис поиска данных, проху для elasticsearch.
- [gs-scheduler](#) — планировщик задач индексации данных из Creatio в ElasticSearch.
- [gs-worker](#) — компонент индексирования данных из Creatio в ElasticSearch по задачам планировщика.
- [gs-worker-replay](#) — компонент, обрабатывающий результаты индексации (результаты работы gs-worker-a).
- [gs-worker-single](#) — компонент точечной индексации данных бизнес-процессов в ElasticSearch по запросу из бизнес-процесса.
- [gs-worker-single-replay](#) — компонент, обрабатывающий исключения в процессе точечной индексации (результаты работы gs-worker-single).
- [gs-worker-single-task](#) — компонент для постановки задач компоненту gs-worker-single.
- [gs-worker-querried-single-task](#) — компонент для формирования задач компоненту gs-worker-single.

Для настройки компонентов скачайте исходные файлы. [Скачать файлы](#).

Список портов, используемых компонентами глобального поиска:

**Важно.** Если вы используете FireWall, убедитесь, что все перечисленные порты доступны и открыты.

Название компонента	Исходящий порт	Входящий порт	Примечание
gs-web-api		81	Входящий порт настраивается переменной WEB_API_PORT
gs-web-indexing-service		82	Входящий порт настраивается переменной WEB_INDEXING_SERVICE_PORT
gs-search-service	9200	83	Входящий порт настраивается переменной SEARCH_SERVICE_PORT
gs-worker	9200		Необходимо соединение с сервером, на котором расположен elasticsearch
gs-worker-single	9200		Необходимо соединение с сервером, на котором расположен elasticsearch
elasticsearch		9200	

## Последовательность действий для настройки глобального поиска

1. Установить Docker на физическую или виртуальную машину с операционной системой Linux.
2. Установить Docker-Compose.
3. Установить ElasticSearch.
4. Настроить переменные контейнеров.
5. Установить и запустить компоненты Global Search Service.
6. Подключить функциональность глобального поиска в Creatio.

### Установить Docker

Для развертывания компонентов глобального поиска необходимо установить Docker на операционную систему Linux. Для установки воспользуйтесь инструкцией в [документации Docker](#).

Для проверки установленной версии Docker запустите команду **docker --version** на linux-машине.

### Установить Docker-Compose

Для установки Docker-Compose воспользуйтесь инструкцией в [документации Docker](#).

### Установить ElasticSearch

**На заметку.** В данной инструкции описано развертывание ElasticSearch в Docker-Compose. Вы

также можете развернуть его как daemon OS, что позволит обойтись без предварительной установки Docker и Docker-Compose. Для этого воспользуйтесь инструкцией в [документации ElasticSearch](#).

Для установки ElasticSearch:

1. На сервере, выделенном для установки ElasticSearch (сервер 1), зайдите в папку /opt.
2. Распакуйте в открытую папку архив с установочными файлами. [Скачать архив](#)
3. Перейдите в папку с компонентами /opt/compose/elasticsearch и выполните команду:

```
docker-compose up -d
```

Команда может выполняться несколько минут.

4. После выполнения команды проверьте, что в логах нет ошибок. Для этого выполните команду

```
docker logs es-01
```

## Настроить переменные контейнеров

Все контейнеры компонентов глобального поиска конфигурируются из файла с переменными среды (environment variables). Переменные содержатся в файле /opt/compose/services/.env. Установите значения переменных, отредактировав этот файл.

Название переменной	Описание	Значение по умолчанию
GS_ES_URL	Внешний хост ElasticSearch, который используется для доступа из Creatio. Необходимо указать ip-адрес сервера, на котором развернут ElasticSearch.	http://elasticsearch-publicip:9200
CURRENT_SERVER_IP	Внешний ip-адрес сервера, на котором развернуты сервисы глобального поиска (сервер 2).	10.0.0.1

**На заметку.** Чтобы проверить внешний ip-адрес сервера, выполните команду

```
hostname -I | awk '{ print $1 }'
```

Дополнительные переменные, которые управляют параметрами индексации данных в ElasticSearch

Название переменной	Описание	Значение по умолчанию
GS_DB_INCREMENT_DAYS	Количество дней, которое необходимо проиндексировать за одну итерацию планировщика. Для сравнения используются данные колонки ModifiedOn записей системы.	500 дней
GS_DB_FILL_QUEUE_INTERVAL	Интервал сбора данных из БД Creatio регулярным планировщиком. Чем меньше этот параметр, тем выше нагрузка на БД Creatio, но быстрее происходит первичная индексация.	30000 (указывается в миллисекундах)

## Запустить контейнеры с компонентами Global Search Service

**Важно.** Для корректной работы контейнеров необходимо, чтобы UTC-время на linux-машине, на которой установлен Docker, соответствовало UTC-времени на сервере БД Creatio. Допустимое отклонение — до пяти минут. Иначе глобальный поиск может индексировать не все записи.

1. На сервере, выделенном для установки компонентов глобального поиска (сервер 2), зайдите в папку `opt`.
2. Распакуйте в открытую папку архив с установочными файлами. [Скачать архив](#)
3. Перейдите в папку с компонентами `/opt/compose/services` и выполните команду:

```
docker-compose up -d
```

### Проверить успешность запуска контейнеров

Для просмотра всех запущенных контейнеров глобального поиска введите в консоли команду:

```
docker ps --filter "label=service=gs" -a --format "table {{.Names}}\t{{.Ports}}\t{{.Status}}\t{{
```

Все запущенные контейнеры должны быть со статусом **Up**.

### Логирование

По умолчанию в контейнере логирование происходит в `stdout` и `stderr`.



**На заметку.** `docker logs --tail 100 gs-worker-01` выводит 100 последних строк логов из контейнера `gs-worker`.

**На заметку.** При старте возможны ситуации, когда контейнер `mysql` или `rabbitmq` временно недоступны, так как они запускаются позже остальных компонентов. В этом случае дождитесь, пока в логах не отобразится сообщение об удачном подключении и запуске контейнера, например: "Now listening on: http://[::]:80 Application started. Press Ctrl+C to shut down".

## Подключить сервис глобального поиска в Creatio

### Действия на сервере

Для подключения глобального поиска к Creatio выполните следующие действия на сервере 2, где находятся компоненты глобального поиска:

1. Для http-запросов установите утилиту `api-get` `install curl` или `yum install curl`.

```
apt-get install curl
```

2. Выполните http-запрос на регистрацию сайта в глобальном поиске, указав:
  - a. `[ DATABASE_TYPE ]` — тип базы данных Creatio (`mssql`, `postgresql` или `oracle`).
  - b. `[ DATABASE_CONNECTION_STRING ]` — строка подключения к БД Creatio.
  - c. `[ SITE_NAME ]` — название сайта Creatio, например, `my-test-site`.
  - d. `[ SERVER2_IP_ADDRESS ]` — IP-адрес Linux сервера, на котором развернуты компоненты глобального поиска.

```
curl -v -X POST -d '{"databaseType": "[DATABASE_TYPE]", "databaseConnectionString": "[DATAI
```

**Пример для MS SQL.** `curl -v -X POST -d '{"databaseType": "mssql", "databaseConnectionString": "Server=myserver\\mssql2016; Database=my-test-site; User Id=my-login; Password='my-password'; Connection Timeout=10"}' -H "Content-Type: application/json" http://[SERVER2_IP_ADDRESS]:81/sites/my-test-site`

**Пример для PostgreSQL.** `server=[SERVER_IP];port=5432;database=[DB_NAME];user id=[USER_NAME];password=[PASSWORD];timeout=10;commandtimeout=400;maxpoolsize=1024`

3. Выполните http-запрос на подключение поиска к ранее добавленному сайту, указав:
  - a. `[ SITE_NAME ]` — название сайта Creatio, например, `my-test-site`.

- b. [ *TEMPLATE\_NAME* ] — название шаблона поиска, который используется в Elasticsearch. Перечень доступных к использованию шаблонов приведен в таблице ниже.
- c. [ *SERVER2\_IP\_ADDRESS* ] — IP-адрес Linux сервера, на котором развернуты компоненты глобального поиска.

```
curl -v -X POST -d '{"templateName": "[TEMPLATE_NAME]"}' -H "Content-Type: application/json"
```

**Пример.** `curl -v -X POST -d '{"templateName": "default.json"}' -H "Content-Type: application/json" http://[SERVER2_IP_ADDRESS]:81/sites/my-test-site/search`

**На заметку.** Приведенный запрос вернет URL к созданному индексу в Elasticsearch. URL нужно сохранить и использовать в приведенном ниже sql-скрипте установки системных настроек.

**Важно.** Чтобы изменить шаблон поиска, выполните DELETE запрос на `/sites/{siteName}/search` и http-запрос на подключение поиска, описанный выше, после чего будет выполнена полная переиндексация сайта.

Все доступные шаблоны поиска и их характеристики приведены в таблице ниже:

	Старый шаблон (версия 1.6)	default.json	ngram_2.json	ngram_3.json	without_ng
Поиск по части слова	+	-	+	+	-
Поиск по слову с опечатками	+	-	+	+	-
Поиск по части номера телефона в средствах связи	+	+	+	+	-
Нечеткий поиск по средствам связи	+	+	+	+	-
Поиск по...	.	.	.	.	.

Поиск с перестановкой слов	+	+	+	+	+
Поиск по точному совпадению	+	+	+	+	+
Поиск по двум символам	-	-	+	-	-
Примерная скорость поиска (чем меньше, тем лучше)		1x	13x	7x	<1x
Размер индекса на elasticsearch (чем меньше, тем лучше)		1x	4x	2,5x	<1x
Время полной первичной индексации (чем меньше, тем лучше)		1x	1,8x	1,4x	<1x

## Действия на стороне Creatio для СУБД MS SQL

1. Включите в Creatio функциональность (Feature Toggle) глобального поиска (GlobalSearch, GlobalSearch\_V2 GlobalSearchRelatedEntityIndexing), выполнив sql-скрипт:

```

DECLARE @GS_REIndexingFeature NVARCHAR(50) = 'GlobalSearchRelatedEntityIndexing';
DECLARE @GS_REIndexingFeatureId UNIQUEIDENTIFIER = (SELECT TOP 1 Id FROM Feature WHERE
Code = @GS_REIndexingFeature);

DECLARE @GlobalSearchFeature NVARCHAR(50) = 'GlobalSearch';
DECLARE @GlobalSearchFeatureId UNIQUEIDENTIFIER = (SELECT TOP 1 Id FROM Feature WHERE Code =

DECLARE @GlobalSearchV2Feature NVARCHAR(50) = 'GlobalSearch_V2';
DECLARE @GlobalSearchV2FeatureId UNIQUEIDENTIFIER = (SELECT TOP 1 Id FROM Feature WHERE Code
DECLARE @allEmployeesId UNIQUEIDENTIFIER = 'A29A3BA5-4B0D-DE11-9A51-005056C00008';

IF (@GlobalSearchFeatureId IS NOT NULL)

```

```

BEGIN
    IF EXISTS (SELECT * FROM AdminUnitFeatureState WHERE FeatureId = @GlobalSearchFeatureId)
        UPDATE AdminUnitFeatureState SET FeatureState = 1 WHERE FeatureId = @GlobalSearchFeature
    ELSE
        INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@all
    END;
ELSE
    BEGIN
        SET @GlobalSearchFeatureId = NEWID()
        INSERT INTO Feature (Id, Name, Code) VALUES (@GlobalSearchFeatureId, @GlobalSearchFeature,
        INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allEm
    END;

IF (@GlobalSearchV2FeatureId IS NOT NULL)
    BEGIN
        IF EXISTS (SELECT * FROM AdminUnitFeatureState WHERE FeatureId = @GlobalSearchV2FeatureId)
            UPDATE AdminUnitFeatureState SET FeatureState = 1 WHERE FeatureId = @GlobalSearchV2Featu
        ELSE
            INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@all
    END;
ELSE
    BEGIN
        SET @GlobalSearchV2FeatureId = NEWID()
        INSERT INTO Feature (Id, Name, Code) VALUES (@GlobalSearchV2FeatureId, @GlobalSearchV2Feat
        INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allEm
    END;

IF (@GS_REIndexingFeatureId IS NOT NULL)
    BEGIN
        IF EXISTS (SELECT * FROM AdminUnitFeatureState WHERE FeatureId = @GS_REIndexingFeatureId)
            UPDATE AdminUnitFeatureState SET FeatureState = 1 WHERE FeatureId = @GS_REIndexingFeatureId
        ELSE
            INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allEmp
    END;
ELSE
    BEGIN
        SET @GS_REIndexingFeatureId = NEWID()
        INSERT INTO Feature (Id, Name, Code) VALUES (@GS_REIndexingFeatureId, @GS_REIndexingFeature,
        INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allEmpl
    END;

```

## 2. Установите значения системных настроек:

- a. "GlobalSearchUrl" — полный путь в elasticsearch с учетом индекса. Значение настройки возвращается при выполнении запроса на web-аpi для добавления поиска для сайта. Пример строки: `http://[SERVER2_IP_ADDRESS]:83/indexname`.
- b. "GlobalSearchConfigServiceURL" — ссылка на API глобального поиска. Значение по умолчанию: `http://[SERVER2_IP_ADDRESS]:81`.

- c. "GlobalSearchIndexingApiUrl" — ссылка на сервис моментальной индексации. Значение по умолчанию: `http://[SERVER2_IP_ADDRESS]:82`.

```
UPDATE SysSettingsValue
SET TextValue = [укажите URL к индексу ElasticSearch, строка типа: http://[SERVER2_IP_ADDRESS]:82]
WHERE SysSettingsId = (SELECT TOP 1 Id FROM SysSettings WHERE Code = 'GlobalSearchUrl')

UPDATE SysSettingsValue
SET TextValue = [укажите URL к Global Search Service, строка типа: http://[SERVER2_IP_ADDRESS]:82]
WHERE SysSettingsId = (SELECT TOP 1 Id FROM SysSettings WHERE Code = 'GlobalSearchConfigServiceUrl')

UPDATE SysSettingsValue
SET TextValue = [укажите URL к Global Search Indexing Service, строка типа: http://[SERVER2_IP_ADDRESS]:82]
WHERE SysSettingsId = (SELECT TOP 1 Id FROM SysSettings WHERE Code = 'GlobalSearchIndexingApiUrl')
```

3. Перезапустите приложение Creatio, очистите redis и выполните вход в приложение.

## Действия на стороне Creatio для СУБД Oracle

1. Включите в Creatio функциональность (Feature Toggle) глобального поиска (GlobalSearch, GlobalSearch\_V2 GlobalSearchRelatedEntityIndexing), выполнив sql-скрипт:

```
CREATE OR REPLACE FUNCTION
generate_uuid return varchar2 is
    v_uuid varchar2(38);
    v_guid varchar2(32);
BEGIN
    v_guid := sys_guid();
    v_uuid := lower(
        '{' ||
        substr(v_guid, 1,8) || '-' ||
        substr(v_guid, 9,4) || '-' ||
        substr(v_guid, 13,4) || '-' ||
        substr(v_guid, 17,4) || '-' ||
        substr(v_guid, 21) ||
        '}'
    );
    RETURN v_uuid;
END;
/

DECLARE
    GS_REIndexingFeature VARCHAR(50) := 'GlobalSearchRelatedEntityIndexing';
    GS_REIndexingFeatureId VARCHAR(38) := NULL;
    GS_REIndexingFeatureId_GUID VARCHAR(38) := generate_uuid();
```

```

GlobalSearchFeature VARCHAR(50) := 'GlobalSearch';
GlobalSearchFeatureId VARCHAR(38) := NULL;
GlobalSearchFeatureId_GUID VARCHAR(38) := generate_uuid();
GlobalSearchV2Feature VARCHAR(50) := 'GlobalSearch_V2';
GlobalSearchV2FeatureId VARCHAR(38) := NULL;
GlobalSearchV2FeatureId_GUID VARCHAR(38) := generate_uuid();
allEmployeesId VARCHAR(38) := '{7F3B869F-34F3-4F20-AB4D-7480A5FDF647}';
State_GlobalSearch VARCHAR(1) := NULL;
State_GlobalSearchV2 VARCHAR(1) := NULL;
State_GS_REI VARCHAR(1) := NULL;

BEGIN
    SELECT MAX("Id") INTO GlobalSearchFeatureId FROM "Feature" WHERE "Code" = GlobalSearchFeatu
    SELECT MAX("Id") INTO GlobalSearchV2FeatureId FROM "Feature" WHERE "Code" = GlobalSearchV2F
    SELECT MAX("Id") INTO GS_REIndexingFeatureId FROM "Feature" WHERE "Code" = GS_REIndexingFeatu

    SELECT MAX("FeatureState") INTO State_GlobalSearch FROM "AdminUnitFeatureState" WHERE "Feat
    SELECT MAX("FeatureState") INTO State_GlobalSearchV2 FROM "AdminUnitFeatureState" WHERE "Fe
    SELECT MAX("FeatureState") INTO State_GS_REI FROM "AdminUnitFeatureState" WHERE FeatureId"

    IF (GlobalSearchFeatureId IS NULL) THEN
        INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (GlobalSearchFeatureId_GUID, Global
        INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VAL
    ELSE
        IF (State_GlobalSearch IS NOT NULL) THEN
            UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = GlobalSea
        ELSE
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId")
        END IF;
    END IF;

    IF (GlobalSearchV2FeatureId IS NULL) THEN
        INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (GlobalSearchV2FeatureId_GUID, Glob
        INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VAL
    ELSE
        IF (State_GlobalSearchV2 IS NOT NULL) THEN
            UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = GlobalSea
        ELSE
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId")
        END IF;
    END IF;

    IF (GS_REIndexingFeatureId IS NULL) THEN
        INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (GS_REIndexingFeatureId_GUID, GS_REIndex
        INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (
    ELSE
        IF (State_GS_REI IS NOT NULL) THEN
            UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = GS_REIndexingFeatu
        ELSE

```

```

INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState","FeatureId") VALUES (
END IF;
END IF;

END;

```

## 2. Установите значения системных настроек:

- “GlobalSearchUrl” — полный путь в elasticsearch с учетом индекса. Значение настройки возвращается при выполнении запроса на web-api для добавления поиска для сайта. Пример строки: `http://[SERVER2_IP_ADDRESS]:83/indexname`.
  - GlobalSearchConfigServiceURL” — ссылка на API глобального поиска. Значение по умолчанию: `http://[SERVER2_IP_ADDRESS]:81`.
  - “GlobalSearchIndexingApiUrl” — ссылка на сервис моментальной индексации. Значение по умолчанию: `http://[SERVER2_IP_ADDRESS]:82`.
- Для этого выполните следующий скрипт:

```

DECLARE
URL_SETTING_ID VARCHAR(38) := NULL;
CONFIG_URL_SETTING_ID VARCHAR(38) := NULL;
IND_API_SETTING_ID VARCHAR(38) := NULL;

URL_VAL_ID VARCHAR(38) := NULL;
CONFIG_URL_VAL_ID VARCHAR(38) := NULL;
IND_API_VAL_ID VARCHAR(38) := NULL;

SYS_ADMIN_UID VARCHAR(38) := '{A29A3BA5-4B0D-DE11-9A51-005056C00008}';

ES_IND VARCHAR(500) := '[укажите URL к индексу ElasticSearch, строка, сл. типа - http://[SE
CONFIG_URL VARCHAR(500) := '[укажите URL к Global Search Service, строка следующего типа: h
IND_API_URL VARCHAR(500) := '[укажите URL к Global Search Indexing Service, строка следующе
BEGIN
SELECT "Id" INTO URL_SETTING_ID FROM "SysSettings" WHERE "Code" = 'GlobalSearchUrl';
SELECT "Id" INTO CONFIG_URL_SETTING_ID FROM "SysSettings" WHERE "Code" = 'GlobalSearchConfi
SELECT "Id" INTO IND_API_SETTING_ID FROM "SysSettings" WHERE "Code" = 'GlobalSearchIndexing

SELECT MAX("Id") INTO URL_VAL_ID FROM "SysSettingsValue" WHERE "SysSettingsId" = URL_SETTI
SELECT MAX("Id") INTO CONFIG_URL_VAL_ID FROM "SysSettingsValue" WHERE "SysSettingsId" = CO
SELECT MAX("Id") INTO IND_API_VAL_ID FROM "SysSettingsValue" WHERE "SysSettingsId" = IND_A

IF (URL_VAL_ID IS NULL)
THEN
INSERT INTO "SysSettingsValue"
("SysSettingsId", "SysAdminUnitId", "IsDef", "TextValue")
VALUES
(URL_SETTING_ID, SYS_ADMIN_UID, '1', ES_IND);
ELSE

```

```

        UPDATE "SysSettingsValue" SET "TextValue" = ES_IND WHERE "SysSettingsId" = URL_SETTING_
    END IF;

    IF (CONFIG_URL_VAL_ID IS NULL)
    THEN
        INSERT INTO "SysSettingsValue"
            ("SysSettingsId", "SysAdminUnitId", "IsDef", "TextValue")
        VALUES
            (CONFIG_URL_SETTING_ID, SYS_ADMIN_UID, '1', CONFIG_URL);
    ELSE
        UPDATE "SysSettingsValue" SET "TextValue" = CONFIG_URL WHERE "SysSettingsId" = CONFIG_U
    END IF;

    IF (IND_API_VAL_ID IS NULL)
    THEN
        INSERT INTO "SysSettingsValue"
            ("SysSettingsId", "SysAdminUnitId", "IsDef", "TextValue")
        VALUES
            (IND_API_SETTING_ID, SYS_ADMIN_UID, '1', IND_API_URL);
    ELSE
        UPDATE "SysSettingsValue" SET "TextValue" = IND_API_URL WHERE "SysSettingsId" = IND_API
    END IF;
END;

```

3. Перезапустите приложение Creatio, очистите redis и выполните вход в приложение.

## Действия на стороне Creatio для СУБД PostgreSQL

1. Включите в Creatio функциональность (Feature Toggle) глобального поиска (GlobalSearch, GlobalSearch\_V2 GlobalSearchRelatedEntityIndexing), выполнив sql-скрипт:

```

DO $$

DECLARE
    GlobalSearchFeature VARCHAR(50) := 'GlobalSearch';
    GlobalSearchFeatureId uuid;
    GlobalSearchV2Feature VARCHAR(50) := 'GlobalSearch_V2';
    GlobalSearchV2FeatureId uuid;
    GS_RelatedEntityIndexingFeature VARCHAR(50) := 'GlobalSearchRelatedEntityIndexing';
    GS_RelatedEntityIndexingFeatureId uuid;
    allEmployeesId uuid := 'A29A3BA5-4B0D-DE11-9A51-005056C00008';

BEGIN

    SELECT "Id" INTO GlobalSearchFeatureId FROM "Feature"
    WHERE "Code" = GlobalSearchFeature
    LIMIT 1;

```



```

IF (GlobalSearchFeatureId IS NOT NULL)
    THEN
        IF EXISTS (SELECT * FROM "AdminUnitFeatureState" WHERE "FeatureId" = GlobalSearchFe
            UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = Global
        ELSE
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "Feature
        END IF;
    ELSE
        GlobalSearchFeatureId := uuid_generate_v4();
        INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (GlobalSearchFeatureId, GlobalSear
        INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VA
    END IF;

SELECT "Id" INTO GlobalSearchV2FeatureId FROM "Feature"
WHERE "Code" = GlobalSearchV2Feature
LIMIT 1;
IF (GlobalSearchV2FeatureId IS NOT NULL)
    THEN
        IF EXISTS (SELECT * FROM "AdminUnitFeatureState" WHERE "FeatureId" = GlobalSearchV2Fe
            UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = GlobalS
        ELSE
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId"
        END IF;
    ELSE
        GlobalSearchV2FeatureId := uuid_generate_v4();
        INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (GlobalSearchV2FeatureId, GlobalSe
        INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VA
    END IF;

SELECT "Id" INTO GS_RelatedEntityIndexingFeatureId FROM "Feature" WHERE "Code" =GS_RelatedE
IF (GS_RelatedEntityIndexingFeatureId IS NOT NULL)
    THEN
        IF EXISTS (SELECT * FROM "AdminUnitFeatureState" WHERE "FeatureId" = Bulk_ES_DD_FeatureId)
        UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = GS_RelatedEntityInd
        ELSE
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (
        END IF;
        ELSE
            GS_RelatedEntityIndexingFeatureId := uuid_generate_v4();
            INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (GS_RelatedEntityIndexingFeatureId, GS_
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (
        END IF;
END $$;

```

## 2. Установите значения системных настроек:

- a. "GlobalSearchUrl" — полный путь в elasticsearch с учетом индекса. Значение настройки возвращается при выполнении запроса на web-api для добавления поиска для сайта. Пример строки: `http://[SERVER2_IP_ADDRESS]:83/indexname`.

- b. GlobalSearchConfigServiceURL — ссылка на API глобального поиска. Значение по умолчанию: `http://[SERVER2_IP_ADDRESS]:81`.
- c. "GlobalSearchIndexingApiUrl" — ссылка на сервис моментальной индексации. Значение по умолчанию: `http://[SERVER2_IP_ADDRESS]:82`.

Для этого выполните следующий скрипт:

```
UPDATE "SysSettingsValue"
SET "TextValue" = [укажите URL к индексу Elasticsearch, строка следующего типа: http://[SERVER2_IP_ADDRESS]:81]
WHERE "SysSettingsId" = (SELECT "Id" FROM "SysSettings" WHERE "Code" = 'GlobalSearchUrl' LIMIT 1)

UPDATE "SysSettingsValue"
SET "TextValue" = [укажите URL к Global Search Service, строка следующего типа: http://SERVER2_IP_ADDRESS:81]
WHERE "SysSettingsId" = (SELECT "Id" FROM "SysSettings" WHERE "Code" = 'GlobalSearchConfigServiceUrl' LIMIT 1)

UPDATE "SysSettingsValue"
SET "TextValue" = [укажите URL к Global Search Indexing Service, строка, сл. типа - http://SERVER2_IP_ADDRESS:82]
WHERE "SysSettingsId" = (SELECT "Id" FROM "SysSettings" WHERE "Code" = 'GlobalSearchIndexingApiUrl' LIMIT 1)
```

3. Перезапустите приложение Creatio, очистите redis и выполните вход в приложение.

## Настроить массовый поиск дублей

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Сервис массового поиска дублей используется для дедупликации в разделах Creatio.

**Важно.** Для работы сервиса массового поиска дублей необходимо предварительно настроить сервис глобального поиска в Elasticsearch. Информацию о настройке читайте в статье [Настроить глобальный поиск](#).

Для настройки сервиса массового поиска дублей необходимы базовые знания администрирования docker-compose и ОС Linux.

Сервис глобального поиска дублей имеет собственную версию. Данная инструкция универсальна для всех версий сервиса. Версии 1.0–1.5 совместимы с любой версией Creatio, в которой есть функциональность глобального поиска дублей. Версия сервиса 2.0 совместима с Creatio версий 7.15.4 — 7.17.0. Версия 3.0 совместима с Creatio версии 7.17.0 и выше и глобальным поиском версии 3.0.

Для расчета требований к серверам воспользуйтесь [калькулятором системных требований](#).

## Компоненты сервиса массового поиска дублей

Для настройки сервиса необходимы:

1. Компоненты глобального поиска. Список приведен в статье [Настроить глобальный поиск](#).
2. Компоненты сервиса массового поиска дублей. Перечень компонентов приведен ниже.

[Mongodb](#) — документоориентированная система управления базами данных.

[dd-web-api](#) — web-сервис для общения в Creatio.

[dd-data-service](#) — внутренний сервис коммуникаций с mongodb.

[dd-duplicates-search-worker](#) — компонент поиска дублей.

[dd-duplicates-deletion-worker](#) — компонент, выполняющий точечное удаление дублей.

[dd-duplicates-confirmation-worker](#) — компонент, выполняющий группировку и фильтрацию найденных дублей с учетом уникальности.

[dd-duplicates-cleaner](#) — компонент очистки дублей.

[dd-deduplication-task-worker](#) — компонент постановки задачи дедупликации.

[dd-deduplication-preparation-worker](#) — компонент подготовки процесса дедупликации, формирует запросы для поиска дублей согласно правилам.

[dd-deduplication-task-diagnostic-worker](#) — компонент, контролирующий выполнение задачи поиска дублей.

Для настройки компонентов скачайте исходные файлы. [Скачать файлы](#).

1. Развернуть и настроить глобальный поиск в Creatio.
2. Скачать, разархивировать и скопировать необходимые для настройки исходные файлы на машину с установленным ПО docker, docker-compose. [Скачать файлы](#).
3. Настроить переменные окружения.
4. Запустить контейнеры.
5. Проверить успешность запуска контейнеров.
6. Проверить логирование.
7. Подключить функциональность массового поиска дублей в Creatio.

## Настроить переменные окружения

Переменные окружения содержатся в файле compose/.env. Установите значения переменных, отредактировав этот файл.

Название переменной	Описание	Значение по умолчанию
ELASTICSEARCH_URI	Ip-адрес сервера, на котором был развернут Elasticsearch на шаге настройки глобального поиска в Creatio.	http://user:password@external.elasticsearch-ip:9200/

## Запустить контейнеры

Для запуска контейнеров необходимо выполнить команду:

```
cd compose # перейти в папку compose
docker-compose up -d
```

## Проверить успешность запуска контейнеров

Для просмотра списка всех запущенных контейнеров необходимо в консоли выполнить команду:

```
docker ps --filter "label=service=dd" -a --format "table {{.Names}}\t{{.Ports}}\t{{.Status}}\t{{
```

У всех запущенных контейнеров отобразится статус “Up”.

## Проверить логирование

По умолчанию логирование происходит во время выполнения команды “stdout” контейнеров. Чтобы просмотреть последние 100 записей логов из контейнера dd-data-service, выполните команду:

```
docker logs --tail 100 dd-data-service
```

## Подключить функциональность массового поиска дублей в Creatio

На стороне Creatio необходимо выполнить следующие шаги.

1. Установите значение системной настройки “Адрес сервиса дедупликации”.
2. Настройте права на системную операцию “Поиск дублей”.
3. Включите в Creatio функциональность массового поиска дублей. Обратите внимание, что данная настройка отличается для разных СУБД.
4. Перезапустите приложение Creatio.

### Установить значение системной настройки “Адрес сервиса дедупликации”

В разделе [ *Системные настройки* ] найдите настройку “Адрес сервиса дедупликации” (DeduplicationWebApiUrl) и в ней укажите URL к dd-web-api, строка следующего типа:  
http://external.deduplication-web-api:8086.

### Настроить права на системную операцию “Поиск дублей”

В разделе [ *Доступ к операциям* ] откройте системную операцию “Поиск дублей” (CanSearchDuplicates) и на детали [ *Доступ к операции* ] раздайте права необходимым пользователям/ролям, которые смогут

выполнять поиск дублей.

## Включить в Creatio функциональность массового поиска дублей

Для включения функциональности (Feature Toggle) массового поиска дублей (Deduplication, ESDeduplication, BulkESDeduplication) необходимо выполнить SQL-скрипт. Этот скрипт различается в зависимости от используемой СУБД — MS SQL, Oracle или Postgre SQL.

Для СУБД MS SQL

```
DECLARE @DeduplicationFeature NVARCHAR(50) = 'Deduplication';
DECLARE @DeduplicationFeatureId UNIQUEIDENTIFIER = (SELECT TOP 1 Id FROM Feature WHERE Code = @DeduplicationFeature);

DECLARE @ESDeduplicationFeature NVARCHAR(50) = 'ESDeduplication';
DECLARE @ESDeduplicationFeatureId UNIQUEIDENTIFIER = (SELECT TOP 1 Id FROM Feature WHERE Code = @ESDeduplicationFeature);

DECLARE @Bulk_ES_DD_Feature NVARCHAR(50) = 'BulkESDeduplication';
DECLARE @Bulk_ES_DD_FeatureId UNIQUEIDENTIFIER = (SELECT TOP 1 Id FROM Feature WHERE Code = @Bulk_ES_DD_Feature);

DECLARE @allEmployeesId UNIQUEIDENTIFIER = 'A29A3BA5-4B0D-DE11-9A51-005056C00008';
IF (@DeduplicationFeatureId IS NOT NULL)
BEGIN
    IF EXISTS (SELECT * FROM AdminUnitFeatureState WHERE FeatureId = @DeduplicationFeatureId)
        UPDATE AdminUnitFeatureState SET FeatureState = 1 WHERE FeatureId = @DeduplicationFeatureId
    ELSE
        INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allEmployeesId, 1, @DeduplicationFeatureId)
    END;
ELSE
BEGIN
    SET @DeduplicationFeatureId = NEWID()
    INSERT INTO Feature (Id, Name, Code) VALUES (@DeduplicationFeatureId, @DeduplicationFeature, @DeduplicationFeature)
    INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allEmployeesId, 1, @DeduplicationFeatureId)
    END;

IF (@ESDeduplicationFeatureId IS NOT NULL)
BEGIN
    IF EXISTS (SELECT * FROM AdminUnitFeatureState WHERE FeatureId = @ESDeduplicationFeatureId)
        UPDATE AdminUnitFeatureState SET FeatureState = 1 WHERE FeatureId = @ESDeduplicationFeatureId
    ELSE
        INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allEmployeesId, 1, @ESDeduplicationFeatureId)
    END;
ELSE
BEGIN
    SET @ESDeduplicationFeatureId = NEWID()
    INSERT INTO Feature (Id, Name, Code) VALUES (@ESDeduplicationFeatureId, @ESDeduplicationFeature, @ESDeduplicationFeature)
```

```

        INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allEn
END;

IF (@Bulk_ES_DD_FeatureId IS NOT NULL)
BEGIN
    IF EXISTS (SELECT * FROM AdminUnitFeatureState WHERE FeatureId = @Bulk_ES_DD_FeatureId)
        UPDATE AdminUnitFeatureState SET FeatureState = 1 WHERE FeatureId =@Bulk_ES_DD_FeatureId
    ELSE
        INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState,FeatureId) VALUES (@allEn
END;
ELSE
BEGIN
    SET @Bulk_ES_DD_FeatureId = NEWID()
    INSERT INTO Feature (Id, Name, Code) VALUES (@Bulk_ES_DD_FeatureId, @Bulk_ES_DD_Feature, @
    INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allEn
END;

```

#### Для СУБД Oracle

```

CREATE OR REPLACE FUNCTION
generate_uuid return varchar2 is
    v_uuid varchar2(38);
    v_guid varchar2(32);
BEGIN
    v_guid := sys_guid();
    v_uuid := lower(
'{' ||
    substr(v_guid, 1,8) || '-' ||
    substr(v_guid, 9,4) || '-' ||
    substr(v_guid, 13,4) || '-' ||
    substr(v_guid, 17,4) || '-' ||
    substr(v_guid, 21) ||
    '}');
    );
    RETURN v_uuid;
END;
/
DECLARE
    DeduplicationFeature VARCHAR(50) := 'Deduplication';
    DeduplicationFeatureId VARCHAR(38) := NULL;
    DeduplicationFeatureId_GUID VARCHAR(38) := generate_uuid();
    ESDeduplicationFeature VARCHAR(50) := 'ESDeduplication';
    ESDeduplicationFeatureId VARCHAR(38) := NULL;
    ESDeduplicationFeatureId_GUID VARCHAR(38) := generate_uuid();
    BulkESDeduplicationFeature VARCHAR(50) := 'BulkESDeduplication';
    BulkESDeduplicationFeatureId VARCHAR(38) := NULL;
    Bulk_ES_DD_GUID VARCHAR(38) := generate_uuid();

```

```

allEmployeesId VARCHAR(38) := '{7F3B869F-34F3-4F20-AB4D-7480A5FDF647}';
State_Deduplication VARCHAR(1) := NULL;
State_ESDeduplication VARCHAR(1) := NULL;
State_BulkESDeduplication VARCHAR(1) := NULL;

BEGIN
    SELECT MAX("Id") INTO DeduplicationFeatureId FROM "Feature" WHERE "Code" = DeduplicationFe
    SELECT MAX("Id") INTO ESDeduplicationFeatureId FROM "Feature" WHERE "Code" = ESDeduplicati
    SELECT MAX("Id") INTO BulkESDeduplicationFeatureId FROM "Feature" WHERE "Code" = BulkESDec
    SELECT MAX("FeatureState") INTO State_Deduplication FROM "AdminUnitFeatureState" WHERE "Fe
    SELECT MAX("FeatureState") INTO State_ESDeduplication FROM "AdminUnitFeatureState" WHERE "
    SELECT MAX("FeatureState") INTO State_BulkESDeduplication FROM "AdminUnitFeatureState" WHE
    IF (DeduplicationFeatureId IS NULL) THEN
        INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (DeduplicationFeatureId_GUID, Dedupli
        INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUE
    ELSE
        IF (State_Deduplication IS NOT NULL) THEN
            UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = DeduplicationFe
        ELSE
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUE
        END IF;
    END IF;
    IF (ESDeduplicationFeatureId IS NULL) THEN
        INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (ESDeduplicationFeatureId_GUID, ESDec
        INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUE
    ELSE
        IF (State_ESDeduplication IS NOT NULL) THEN
            UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = ESDeduplicatio
        ELSE
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUE
        END IF;
    END IF;
    IF (BulkESDeduplicationFeatureId IS NULL) THEN
        INSERT INTO "Feature" ("Id", "Name", "Code") VALUES(Bulk_ES_DD_GUID, BulkESDeduplicatio
        INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUE
    ELSE
        IF (State_BulkESDeduplication IS NOT NULL) THEN
            UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = BulkESDeduplica
        ELSE
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUE
        END IF;
    END IF;
END;

```

Для СУБД PostgreSQL

```

DO $$

DECLARE
    DeduplicationFeature VARCHAR(50) := 'Deduplication';
    DeduplicationFeatureId uuid;

    ESDeduplicationFeature VARCHAR(50) := 'ESDeduplication';
    ESDeduplicationFeatureId uuid;

    Bulk_ES_DD_Feature VARCHAR(50) := 'BulkESDeduplication';
    Bulk_ES_DD_FeatureId uuid;

    allEmployeesId uuid := 'A29A3BA5-4B0D-DE11-9A51-005056C00008';

BEGIN

    SELECT "Id" INTO DeduplicationFeatureId FROM "Feature"
    WHERE "Code" = DeduplicationFeature
    LIMIT 1;
    IF (DeduplicationFeatureId IS NOT NULL)
    THEN
        IF EXISTS (SELECT * FROM "AdminUnitFeatureState" WHERE "FeatureId" = DeduplicationFeatureId)
            UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = DeduplicationFeatureId
        ELSE
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (
            END IF;
    ELSE
        DeduplicationFeatureId := uuid_generate_v4();
        INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (DeduplicationFeatureId, DeduplicationFeature, 'Deduplication');
        INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (allEmployeesId, 1, DeduplicationFeatureId);
    END IF;

    SELECT "Id" INTO ESDeduplicationFeatureId FROM "Feature"
    WHERE "Code" = ESDeduplicationFeature
    LIMIT 1;
    IF (ESDeduplicationFeatureId IS NOT NULL)
    THEN
        IF EXISTS (SELECT * FROM "AdminUnitFeatureState" WHERE "FeatureId" = ESDeduplicationFeatureId)
            UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = ESDeduplicationFeatureId
        ELSE
            INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (
            END IF;
    ELSE
        ESDeduplicationFeatureId := uuid_generate_v4();
        INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (ESDeduplicationFeatureId, ESDeduplicationFeature, 'ESDeduplication');
        INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (allEmployeesId, 1, ESDeduplicationFeatureId);
    END IF;

```



```

SELECT "Id" INTO Bulk_ES_DD_FeatureId FROM "Feature"
WHERE "Code" = Bulk_ES_DD_Feature
LIMIT 1;
IF (Bulk_ES_DD_FeatureId IS NOT NULL)
THEN
  IF EXISTS (SELECT * FROM "AdminUnitFeatureState" WHERE "FeatureId" = Bulk_ES_DD_FeatureId) THEN
    UPDATE "AdminUnitFeatureState" SET "FeatureState" = 1 WHERE "FeatureId" = Bulk_ES_DD_FeatureId
  ELSE
    INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (
    END IF;
ELSE
  Bulk_ES_DD_FeatureId := uuid_generate_v4();
  INSERT INTO "Feature" ("Id", "Name", "Code") VALUES (Bulk_ES_DD_FeatureId, Bulk_ES_DD_Feature,
  INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (al
  END IF;
END $$;

```

## Перезапустить приложение Creatio

Очистите redis, перезапустите приложение Creatio и выполните вход в приложение.

## Обновить версию массового поиска дублей

Для обновления сервиса массового поиска дублей с версии 2.0 на версию 3.0 выполните следующие шаги.

1. Сделайте резервную копию данных о дублях в приложении. Для этого через swagger web-api `http://[ip-адрес сервера]:8086/api` выполните команду `/api/snapshot/backup/gzip/{indexName}`, где `{indexName}` — название индекса (последние 64 символа) из системной настройки “Адрес сервиса поиска” (код “GlobalSearchUrl”).
2. Удалите все docker volume версии 2.0. Для этого перейдите в папку docker-compose файлов версии 2.0 и выполните команду [docker-compose down -v](#).
3. Установите сервис массового поиска дублей версии 3.0.
4. Загрузите в обновленный сервис данные о дублях, полученные на шаге 1. Для этого на новой версии сервиса выполните через swagger команду `/api/snapshot/restore/gzip`.

## Рекомендуемые работы по обслуживанию сервиса

Для поддержания работоспособности сервиса и возможности быстрого восстановления данных, например, в случае перебоев с электроэнергией, рекомендуется раз в сутки выполнять резервное копирование mongodb.

## Часто задаваемые вопросы по

# Часто задаваемые вопросы по глобальному поиску и дедупликации

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

## Как узнать, какая версия глобального поиска используется?

**Если вы используете версию Creatio cloud**, то у вас по умолчанию используется последняя версия глобального поиска.

**Если вы используете версию Creatio on-site**, то запустите в консоли команду:

```
docker ps
```

В результате вы увидите список всех запущенных контейнеров. Номер версии глобального поиска можно узнать в колонке [ *image* ].

## Как запустить переиндексацию сайта?

**Для версии глобального поиска 2.0:**

Выполните PUT запрос вида :

```
http://[ GS-WEB-API ]:81/indexation/{siteName}/reindex/full
```

**Для версии глобального поиска ниже 2.0:**

1. Откройте контейнер gs-mysql, выполнив команду:

```
docker exec -it gs-mysql bash
```

2. В контейнере gs-mysql выполните следующие команды:

```
mysql -p1665017 use gs; UPDATE GlobalSearchIndexingEntity SET LastIndexedOn = NULL, InProcess
```

## Как включить логирование работы сервиса глобального поиска?

По умолчанию сервис сохраняет только ошибки, возникающие в ходе работы. Чтобы включить логирование всех событий сервиса глобального поиска, в файле docker-compose.yaml замените строку:

```
-Log4NetPath=${LOG$NET_CONFIG_FILE:-log4net.production.config}
```

на строку вида:

```
-Log4NetPath=${LOG$NET_CONFIG_FILE:-log4net.debug.config}
```

**На заметку.** При включении логирования всех событий существенно увеличивается количество файлов журналирования.

## Какие метрики или системы слежения можно использовать для мониторинга работы глобального поиска?

Вы можете выполнять запрос на:

```
http://[ GS-WEB-API ]:81/sites/[ SITE_NAME ]/search/state
```

В данном случае [ GS-WEB-API ] — это адрес сервера, где находятся компоненты глобального поиска, а [ SITE\_NAME ] — название вашего сайта.

## Как настроить доступ к Elasticsearch по паролю?

Вы можете закрыть доступ к Elasticsearch с помощью Nginx, поддерживающего base64- авторизацию. Для настройки доступа к Elasticsearch по логину и паролю используйте плагин x-pack.

## Как добавить в индекс новый объект или изменить настройки индексируемых полей существующих объектов?

Для **версий Creatio 7.18.4 и ниже** пользовательскими средствами можно включать и отключать поиск для конкретных разделов. По умолчанию индексируются только разделы вне зависимости от их автора, а также текстовые и справочные колонки, кроме ряда исключений. Откройте [файл](#) в формате \*.pdf с актуальным перечнем исключений.

Начиная с **версии Creatio 7.18.5** вы можете оптимизировать индексируемые для глобального поиска колонки. Чтобы увеличить быстродействие системы и снизить нагрузку на сервера, вы можете явно указать те колонки, которые не должны индексироваться для глобального поиска. Например, колонки, которые содержат служебную информацию.

Чтобы изменить перечень индексируемых колонок:


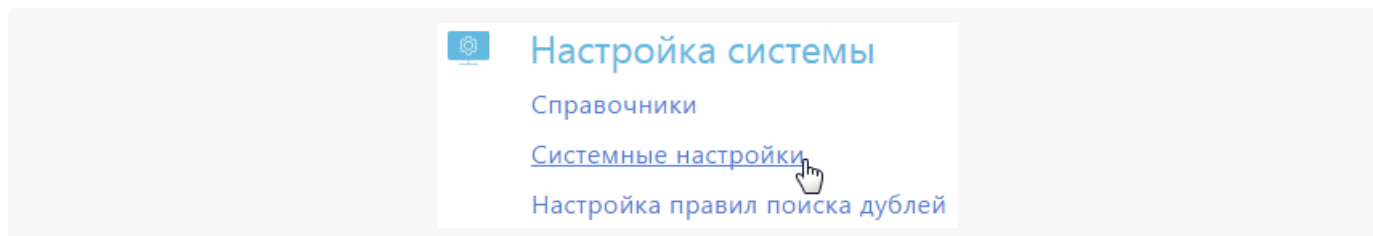
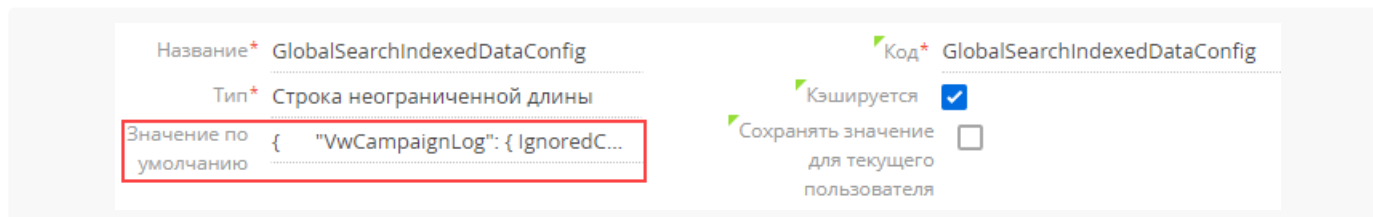
1. Откройте дизайнер системы, например по кнопке .
2. В блоке [ *Настройка системы* ] перейдите по ссылке [ *Системные настройки* ] (Рис. 1).

Рис. 1 — Раздел [ *Системные настройки* ]

3. В реестре системных настроек выберите настройку "GlobalSearchIndexedDataConfig" (код "GlobalSearchIndexedDataConfig") нажмите кнопку [ *Открыть* ].
4. На странице системной настройки в поле [ *Значение по умолчанию* ] указан перечень неиндексируемых колонок в формате JSON (Рис. 2).

Рис. 2 — Поле с перечнем неиндексируемых колонок



5. Скопируйте и отредактируйте локально список неиндексируемых колонок. Для этого вы можете использовать любой онлайн-редактор файлов JSON.

Список неиндексируемых колонок состоит из блоков, каждый из которых содержит уникальный код объекта, который вы можете найти в поле [ *Код* ] [раздела \[ Управление конфигурацией \]](#) и не подлежат индексации. Пример структуры блока представлен ниже:

```
"Contact":{ // код объекта

"IgnoredColumns":[ //перечень колонок, которые не индексируются

"dear", //код колонки, которая не должна индексироваться

"salutationtype",

"gender",...] },
```

**На заметку.** Если вместо названия колонки (объекта) указано "\*", то все колонки (объекты) не будут индексироваться. Если в названии колонки используется нижнее подчеркивание, то исключены будут все колонки, названия которых завершаются таким же фрагментом, как отделенный нижним подчеркиванием в названии. Например, название колонки "\_headerproperties" исключит все колонки, заканчивающиеся на "headerproperties".

6. Проверьте готовый JSON на валидность. Например, с помощью онлайн-редактора.
7. Вставьте обновленный список в поле [ Значение по умолчанию ] и сохраните изменения.
8. Перезапустите индексацию для гарантированного обновления списка колонок в Elasticsearch.

## Как развернуть Elasticsearch на нескольких серверах с указанием одного URL? Как настроить кластеризацию?

Подробные ответы на эти вопросы содержатся в [документации сервиса Elastic](#) на английском языке.

## Что делать, если не отображается настройка “Правила поиска дублей”?

Проверьте, заполнена ли системная настройка “DeduplicationWebApiUrl” и включена ли функциональность:

- “BulkESDeduplication”;
- “ESDeduplication”;
- “Deduplication”.

Подробнее о включении дополнительной функциональности читайте в статье [Механизм отключения функциональности Feature Toggle](#) (документация по разработке).

## Можно ли использовать сервисы глобального поиска и массового поиска дублей одновременно с двумя приложениями Creatio?

Если у вас используются два приложения Creatio, например, продуктовая и тестовая среда, то вы можете подключить сервисы глобального поиска и массового поиска дублей к каждому из приложений. Для установки воспользуйтесь инструкциями:

- [Настроить глобальный поиск.](#)
- [Настроить массовый поиск дублей.](#)

## Как сортируются результаты поиска?

Порядок отображения результатов поиска в основном зависит от релевантности поискового запроса. На релевантность влияют следующие факторы:

- Количество текста в документе.
- Количество вхождений поисковой фразы в документе.
- Количество вхождений поисковой фразы в индексе, а также ряд других, менее значимых, параметров.

Кроме того, порядок сортировки результатов поиска определяют следующие системные настройки:

- “Вес объекта по умолчанию для глобального поиска” (код “GlobalSearchDefaultEntityWeight”) — позволяет повысить приоритет отображения в списке результатов поиска записей, содержащихся в разделе, в котором выполняется поиск. Например, если вы введете поисковый запрос, находясь в разделе [ *Контакты* ], то в начале списка результатов отобразятся записи из данного раздела.
- “Вес первичной колонки по умолчанию для глобального поиска” (код “GlobalSearchDefaultPrimaryColumnWeight”) — позволяет повысить приоритет отображения в списке результатов поиска. Применяется, если совпал поисковый запрос и значение, указанное в первичной колонке такой записи (например, для контакта первичной колонкой является поле [ *ФИО* ], для контрагента — поле [ *Название* ]). Таким образом, если поисковый запрос совпадет со значением в первичной колонке записи, то такая запись будет отображена в начале списка результатов поиска.

Данные системные настройки влияют на релевантность, но не гарантируют отображение результатов поисковой выдачи в точно указанном порядке, поскольку сортировка зависит и от других факторов, упомянутых выше.

## Поддерживается ли аутентификация Windows в микросервисах?

Поскольку сервис глобального поиска и дедупликации разворачивается средствами Docker, аутентификация Windows в микросервисах не поддерживается.

# Сервис машинного обучения

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Сервис машинного обучения предназначен для прогнозирования значений на основании больших объемов исторических данных и текущих фактов. Подробнее: [Предиктивный анализ данных](#).

**Важно.** Для настройки сервиса машинного обучения необходимы базовые знания администрирования Docker, Linux или Windows.

## Подготовка к установке сервиса машинного обучения

**На заметку.** Для корректной работы сервиса в приложении on-site требуется [лицензия](#) “creatio predictive service on-site”. По вопросам ее приобретения необходимо обратиться к ответственному менеджеру.

Для настройки сервиса необходим сервер (физический или виртуальный компьютер) с установленной ОС Linux или Windows.

**Важно.** Для промышленной среды рекомендуется использовать сервер с ОС Linux. Сервер на базе Windows можно использовать только для среды разработки. Для получения Docker-контейнеров,

предназначенных для Windows, обратитесь в службу поддержки.

Установка компонентов сервиса выполняется с помощью ПО Docker. Для этого скачайте архив с конфигурационными файлами и скриптами установки. [Скачать архив](#). В зависимости от потребностей вашей компании можно использовать Docker Community Edition (CE) или Enterprise Edition (EE). Подробную информацию вы найдете в [документации Docker](#).

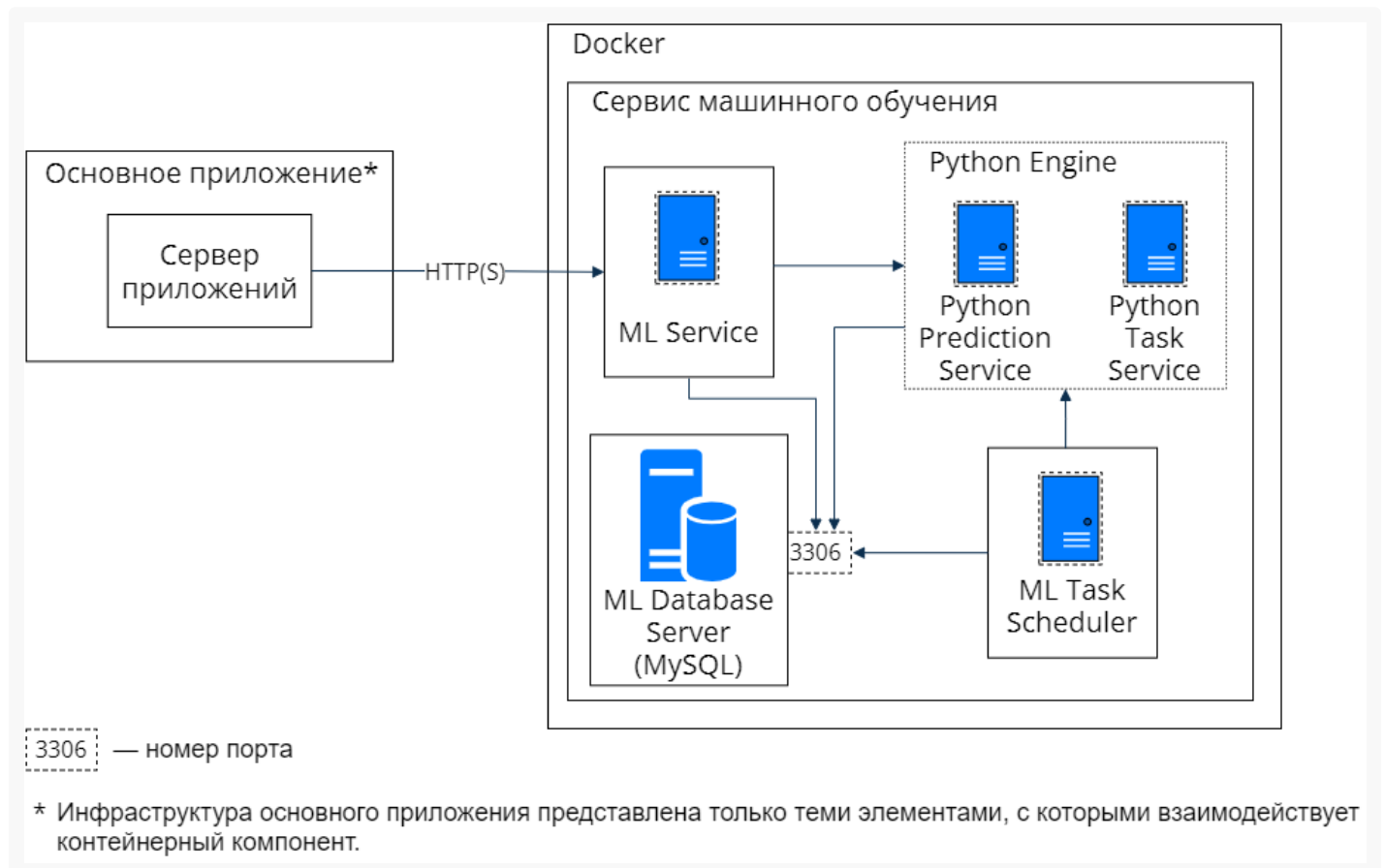
Для расчета требований к серверам воспользуйтесь [калькулятором системных требований](#).

## Компоненты сервиса машинного обучения

Сервис машинного обучения использует следующие компоненты (Рис. 1):

- **ML Service** — web-сервис машинного обучения. Единственный компонент, доступный извне.
- **Python Engine** — движок машинного обучения, представляющий собой сервисную оболочку библиотек машинного обучения с открытым исходным кодом.
- **ML Task Scheduler** — планировщик задач.
- **MySQL** — база данных MySQL. Доступна по стандартному порту 3306.

Рис. 1 — Компоненты сервиса машинного обучения



Для удобства установки сервиса on-site все компоненты упакованы в Docker-образы.

# Установить сервис машинного обучения

Последовательность действий для установки сервиса машинного обучения:

1. Установить Docker. [Подробнее >>>](#)
2. Установить Docker-Compose. [Подробнее >>>](#)
3. Установить и настроить компоненты сервиса. [Подробнее >>>](#)
4. Выполнить проверку установки. [Подробнее >>>](#)

## Установить Docker

Для установки Docker на операционную систему Linux воспользуйтесь инструкцией в [документации Docker](#).

Для проверки установленной версии Docker выполните команду `docker --version` в терминале (консоли) Linux-машины.

## Установить Docker-Compose

Для установки Docker на операционную систему Linux воспользуйтесь инструкцией в [документации Docker](#).

Для проверки установленной версии Docker Compose выполните команду `docker-compose --version` в терминале (консоли) Linux-машины.

## Настроить компоненты сервиса машинного обучения

Все контейнеры компонентов сервиса машинного обучения развертываются с помощью утилиты Docker Compose. Для этого скачайте конфигурационные файлы и скрипты, необходимые для развертывания и настройки компонентов сервиса. [Скачать архив](#)

**На заметку.** Конфигурационные файлы содержат все необходимые настройки по умолчанию для сервера на базе Linux.

### Структура архива с конфигурационными файлами и скриптами:

`/etc/`

`...\ml-service\appsettings.json` — конфигурация web-сервиса.

`...\ml-service\log4net.config` — настройка уровня логирования web-сервиса.

`...\task-scheduler\appsettings.json` — конфигурация утилиты ML task scheduler.

`...\task-scheduler\log4net.config` — настройка уровня логирования ML Task Scheduler.

`docker-compose.yml` — конфигурация утилиты Docker Compose.

`.env` — файл с переменными окружения для запуска компонентов. Например, содержит пароль для MySQL.

**Важно.** Если необходимо изменить пароль к базе данных MySQL, то его нужно обновить не только



в файле .env, но и в других конфигурационных файлах, содержащих секции настройки доступа к базе данных.

## Установить компоненты сервиса машинного обучения

1. Скачайте и разархивируйте содержимое архива с конфигурационными файлами и скриптами в произвольный каталог, например, /opt/ml.
2. Используя терминал Linux, перейдите в каталог docker-compose распакованного архива, например, /opt/ml/docker-compose.
3. Выполните команду `sudo docker-compose pull` в терминале. Дождитесь завершения загрузки необходимых образов с компонентами сервиса с [Docker Hub](#).

**Важно.** Если на сервере запрещен доступ в интернет, скачайте на машине с открытым доступом все необходимые образы вручную (см.конфигурационный файл docker-compose.yml). Затем воспользуйтесь командами [sudo docker export](#) и [sudo docker import](#) для переноса образов в виде файлов на целевую машину.

4. Выполните команду **`sudo docker-compose run dbmigration`** для инициализации структуры базы данных. Дождитесь завершения выполнения команды.
5. Выполните команду **`sudo docker-compose up -d`** для запуска сервисов. Также в текущем каталоге будет создана папка logs.

## Проверить установку компонентов сервиса машинного обучения

1. Для проверки установки веб-сервиса ML service выполните в терминале Linux команду:

**`curl -X GET localhost:5005/readiness`**

Сервис должен вернуть ответ:

**Healthy**

2. Для проверки запуска планировщика ML Task Scheduler выполните в терминале Linux команду:

**`curl -X GET localhost:5004/readiness`**

Сервис должен вернуть ответ:

**Healthy**

3. Для проверки запуска R Engine выполните в терминале Linux команду:

**`curl -X GET localhost:8081/readiness`**

Сервис должен вернуть ответ:

**R Service is ready**

4. Для проверки создания таблиц выполните в терминале команду:

**`docker exec -it DB Container Id mysql -u root --password=Supervisor ml -e "show tables;"`**

Где [ *DB Container Id* ] — идентификатор контейнера с компонентом базы данных. Узнать идентификатор контейнера можно с помощью команды **`sudo docker ps`**.

**Пример.** Проверка создания таблиц:

```
docker exec -it 7c9de07b53e8 mysql -u root --password=Supervisor ml -e "show tables;
```

В результате должны отобразиться названия основных таблиц сервиса: “modelinstance”, “traindata”, “trainsession” и т. д.

## Заполнить настройки Creatio

Для работы с сервисом прогнозирования в Creatio необходимо заполнить:

1. Системную настройку “API ключ облачных сервисов Creatio” (код “CloudServicesAPIKey”) — отвечает за аутентификацию экземпляра Creatio в облачных сервисах.
2. Системную настройку “Периодичность запуска задач машинного обучения” (код “MLModelTrainingPeriodMinutes”) — определяет периодичность запуска синхронизации моделей.
3. Адрес развернутого сервиса прогнозирования в поле [ *ServiceUrl* ] для всех записей в справочнике [ *Задачи машинного обучения* ].

## Обновить компоненты сервиса машинного обучения

**Важно.** Перед обновлением сервисов рекомендуется сохранить резервную копию базы данных MySQL. Подробную информацию вы найдете в [документации MySQL](#).

1. Используя терминал Linux, перейдите в каталог docker-compose с настроечными файлами, например, /opt/ml/docker-compose.
2. Выполните команду **sudo docker-compose stop** для остановки контейнеров компонентов сервиса.
3. Выполните команду **sudo docker-compose pull** в терминале. Дождитесь завершения загрузки необходимых образов с компонентами сервиса с [Docker Hub](#).
4. Выполните команду **sudo docker-compose run dbmigration** для инициализации структуры базы данных. Дождитесь завершения выполнения команды.
5. Выполните команду **sudo docker-compose up -d** для запуска сервисов.

**Важно.** Если у вас в приложении уже есть настроенные и обученные модели, после обновления сервиса рекомендуем их переобучить.

# Настроить сервис синхронизации Exchange Listener

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Сервис синхронизации Exchange Listener реализует синхронизацию Creatio с почтовыми сервисами [MS](#)

[Exchange](#) и [IMAP/SMTP](#), используя механизм подписки.

В данной статье описаны системные требования, а также схемы и процесс развертывания сервиса синхронизации для Creatio on-site.

Сервис состоит из следующих компонентов:

- **Основной модуль Exchange Listener (EL API)** — используется для создания исходящего подключения к EWS API. При этом используются учетные данные почтового ящика и создается подписка (subscription) для получения событий при поступлении новых писем. Открытая подписка остается в памяти компонента для обеспечения максимально оперативной реакции на получение нового письма. При получении соответствующего события выполняется загрузка экземпляра письма. Для развертывания сервиса **достаточным условием** является использование in-memory хранилища. Обязательный компонент сервиса.
- **NoSQL СУБД Redis** — используется для создания масштабируемой и отказоустойчивой системы узлов-обработчиков. Хранилище Redis содержит информацию об обслуживаемых почтовых ящиках. Это позволяет любому контейнеру обработать запросы Creatio на создание новой подписки или проверить статус конкретной подписки, независимо от того, на каком узле она открыта. **Необходимым требованием** к Redis является выделение отдельной базы данных для работы сервиса Exchange Listener. Обязательный компонент сервиса.
- **Дополнительный модуль Exchange Listener (EL Worker)** — используется для поддержания масштабируемой и отказоустойчивой работы основного модуля Exchange Listener. Дополнительный модуль выполняет скачивание писем с почтового сервера и доставку их в приложение Creatio. Таким образом сглаживается обработка пиковых потоков писем для высоконагруженных сервисов. В этом случае API компоненты менее загружены и не занимаются скачиванием писем, а доступны для управления подпиской и отправкой исходящих писем. Опциональный компонент сервиса. При его отсутствии эту роль выполняет основной модуль.
- **RabbitMQ** — используется для поддержания масштабируемой и отказоустойчивой работы сервиса. Брокер очередей выполняет распределение задач между компонентами в высоконагруженных средах. Опциональный компонент сервиса.

Существуют две возможные конфигурации сервиса:

- **Однокомпонентная** — для слабонагруженных сервисов. Данная конфигурация включает только обязательные компоненты сервиса.
- **Многокомпонентная** — для средненагруженных и высоконагруженных сервисов. Данная конфигурация включает все обязательные и опциональные компоненты.

## Определить конфигурацию сервиса Exchange Listener

Конфигурация сервиса Exchange Listener для вашего приложения определяется на основании среднего потока писем в секунду (исходящих и входящих), который проходит через почтовые ящики компании.

Например, если в компании используется один почтовый ящик службы поддержки с потоком 4 письма в секунду, то рекомендуемая конфигурация будет включать 15 реплик EL Worker, 4 реплики EL API и сервис RabbitMQ.

Количество почтовых ящиков	Средний поток писем в секунду	Количество реплик EL Worker	Количество реплик EL API	Наличие RabbitMQ
1 – 150	< 1	0	4	Опционально
	1 – 5	15		Обязательно
	>5	25		
>150	< 1	2	1 реплика на каждые 30 почтовых ящиков	
	1 – 5	15		
	>5	25		

**На заметку.** Количество активных реплик дополнительного компонента сервиса (EL Worker) напрямую влияет на скорость обработки писем. В продуктивном использовании письма приходят не равномерным потоком, а волнообразно, поэтому часть времени некоторые реплики EL Worker могут простаивать. В статье приведены рекомендуемые параметры конфигурации, но вы можете использовать меньше реплик, чем указано в таблице. В этом случае в период высокой нагрузки письма будут обрабатываться дольше. Оптимальный баланс между скоростью обработки писем и утилизацией ресурсов следует подбирать согласно требованиям бизнеса.

## Системные требования для реплик компонентов

Компонент	vCPU	Объем памяти
EL Worker	0,1	1,1 Гб
EL API	0.150 (с RabbitMQ), 0.7 (без RabbitMQ)	850 Мб (с RabbitMQ) 1.1 Гб (без RabbitMQ)
Redis	0,5	3 Гб
Rabbit MQ	0,5	4 Гб

**На заметку.** Значения в таблице являются рекомендуемыми, реальное потребление может отличаться в зависимости от сценариев использования сервиса. Рекомендуется выполнять мониторинг потребления CPU и памяти на развернутых сервисах для оптимизации доступных лимитов.

## Способы развертывания сервиса Exchange Listener

Для развертывания сервиса предпочтительным способом является **использование оркестратора Kubernetes и пакетного менеджера Helm**. [Подробнее >>>](#)

Для более быстрого развертывания в среде разработки можно **использовать Docker**. [Подробнее >>>](#)

## Развернуть сервис синхронизации с использованием Kubernetes

Для развертывания сервиса выполните следующие шаги:

1. Предварительно настройте целевое окружение:

- a. Кластер Kubernetes. Подробно о том, как настроить и администрировать кластер, читайте [на сайте документации Kubernetes](#).
- b. Пакетный менеджер Helm. Установка пакетного менеджера подробно описана [на сайте документации Helm](#).

2. Установите Redis. Установка Redis с использованием Helm детально описана [на сайте GitHub](#).

Пример команды для установки Redis:

```
helm install --namespace default --set usePassword=true --set password=<password> --set=slave.
```

В этом примере:

**default** — наименование namespace, куда будет установлен Redis;

**redis** — произвольное имя для экземпляра Redis.

3. Установите модуль Exchange Listener. Для установки модуля [скачайте helm-пакет](#). Доступные параметры helm-пакета описаны в таблице ниже.

**Важно.** Для более новых версий Kubernetes укажите версию API, добавив параметр:

```
--set apiVersion=apps/v1
```

Пример команды для установки Exchange Listener с использованием адреса и относительного пути сервиса:

```
helm install --set ApiUrl=<kubernetes_url> --set ingress.enabled=true --set ingress.path=<lis
```

В этом примере:

**<redis\_host>** — адрес Redis-сервера;

**<kubernetes\_url>** — URL или IP-адрес Kubernetes.

Адрес сервиса Exchange Listener: **<kubernetes\_url>/<listener\_path>**.

Для настройки подключения с использованием HTTPS необходимо установить сервис с [Ingress](#) и действительным SSL сертификатом, а в адресе сервиса Exchange Listener **<kubernetes\_url>** указать HTTPS.

Чтобы проверить доступность, сделайте запрос по адресу:

**<kubernetes\_url>/<listener\_path>/api/listeners/status** (Рис. 1).

Пример команды для установки Exchange Listener с использованием Node IP и адреса порта:

```
helm install --set env.host=<redis_host> --set service.type=<node_IP> --set service.nodePort=
```

Адрес сервиса Exchange Listener — **<node\_IP:node\_port>**.

Чтобы проверить доступность, сделайте запрос по адресу: <node\_IP:node\_port>/api/listeners/status (Рис. 1).

Рис. 1 — Пример ответа сервиса Exchange Listener

```
{
  "ServiceStatus": "Started",
  "version": "0.5.0",
  "connections": {
    "657b3ea8-477f-419c-a07a-4d4cc2158fc5": {
      "SenderEmailAddress": " ",
      "BpmUser": " ",
      "BpmEndpoint": "https:// /0/ServiceModel/ExchangeListenerService.svc/NewEmail",
      "State": "exists",
      "Id": "657b3ea8-477f-419c-a07a-4d4cc2158fc5",
      "RedisKey": "Subscription_657b3ea8-477f-419c-a07a-4d4cc2158fc5_exchangelistener-api-2",
      "UseFullEmail": true
    },
    "332bcae2-0530-4636-bc09-50a594389f53": {
      "SenderEmailAddress": " ",
      "BpmUser": " ",
      "BpmEndpoint": "https:// /0/ServiceModel/ExchangeListenerService.svc/NewEmail",
      "State": "exists",
      "Id": "332bcae2-0530-4636-bc09-50a594389f53",
      "RedisKey": "Subscription_332bcae2-0530-4636-bc09-50a594389f53_exchangelistener-api-1",
      "UseFullEmail": true
    }
  }
}
```

**На заметку.** Если вам нужно отключить маркировку загружаемых писем категорией “Creatio”, то при установке приложения в команде `helm install` нужно указать параметр `--set FeaturesMarkEmailsAsSynchronized=false`.

Доступные параметры helm-пакета Exchange Listener

Параметр	Описание параметра	Значение по умолчанию
replicaCount	Количество StatefulSet-обработчиков.	2
service.type	Тип сервиса. Детально типы сервисов Kubernetes описаны в <a href="#">документации Kubernetes</a> .	ClusterIP
service.nodePort	Если параметр service.type равен NodePort, то в этом параметре указывается внешний порт сервиса. Детально тип NodePort описан в <a href="#">документации Kubernetes</a> .	
env.host	Адрес хоста Redis.	
env.port	Порт хоста Redis.	6379
env.base	Номер базы данных Redis.	0
ingress.enabled	Использование переопределения адресов при помощи ingress.	false
ApiUrl	Адрес сервиса при ingress.enabled=true.	
ingress.path	Относительный путь сервиса.	
log4Net.level	Уровень логирования по умолчанию.	Info

Для расчета требований к серверам воспользуйтесь [калькулятором системных требований](#).

## Развернуть сервис синхронизации в Docker

Для настройки сервиса необходим сервер (физический или виртуальный компьютер) с установленной ОС Linux или Windows.

**Важно.** Развертывание сервиса синхронизации в Docker рекомендуется к установке только для dev-среды. Данный способ отличается высокой скоростью развертывания, но он не обеспечивает выполнение требований, актуальных для продуктивной среды: отказоустойчивости функции, масштабирования для обработки большого количества запросов и единого подхода к управлению компонентами с использованием систем оркестрации контейнеров. Для продуктивных сред настоятельно рекомендуется установка с использованием оркестратора Kubernetes и пакетного менеджера Helm.

Для развертывания сервиса выполните следующие шаги:

1. Предварительно настройте целевое окружение:

- a. Платформа контейнеризации Docker. Подробно о том, как установить и настроить платформу, описано [на сайте документации Docker](#). Для установки Docker Desktop на Windows Server необходимо использовать [отдельную инструкцию](#) на сайте Microsoft.
  - b. СУБД Redis. Redis Server устанавливается при помощи [установочного файла для ОС Windows](#) или при помощи [инструкции для ОС Linux](#). Подробнее о разворачивании СУБД Redis в Docker читайте на сайте сообщества [Docker Hub](#).
2. Установите и запустите модуль Exchange Listener. Для этого необходимо скачать и развернуть образ Docker-контейнера.

Рассмотрим пример команды для скачивания и запуска образа с использованием командной строки и установленного Docker.

```
docker run \
-d \
# Проброс портов
-p <localhost_port>:80 \
--restart unless-stopped \
# Подключение к Redis
--env ExchangeListenerRedisHost=<redis_host>:<redis_port> \
--env ExchangeListenerRedisDatabase=<redis_database_number> \
--env PodName=ExchangeListener \
--name ExchangeListener \
# Актуальный образ ExchangeListener в Docker Hub
bpmonline/exchangelistener:<listener_version>
```

В этом примере:

- <localhost\_port>** — порт локального сервера;
- <redis\_host>** — адрес redis-сервера;
- <redis\_database\_number>** — номер БД redis-сервера;
- <service\_name>** — название сервиса (задается вручную).
- <listener\_version>** — версия микросервиса Listener.

**На заметку.** Актуальную версию Exchange Listener можно узнать в сообществе Docker Hub по [ссылке](#).

3. Чтобы проверить доступность развернутого Docker-контейнера, выполните команду:

```
docker ps -a --filter "name=<service_name>"
```

Адрес сервиса Exchange Listener — **localhost:<localhost\_port>**.

Чтобы проверить доступность, сделайте запрос по адресу: <Localhost:<localhost\_port>/api/listeners/status (Рис. 2).



# Настроить сервис Exchange Listener на стороне Creatio

1. Убедитесь, что анонимный сервис ExchangeListenerService доступен по адресу [ Адрес приложения Creatio ]/0/ServiceModel/ExchangeListenerService.svc (Рис. 2).

Рис. 2 — Пример ответа сервиса ExchangeListenerService

**Service**

This is a Windows® Communication Foundation service.

**Metadata publishing for this service is currently disabled.**

If you have access to the service, you can enable metadata publishing by completing the following steps to modify your web or application configuration file:

1. Create the following service behavior configuration, or add the <serviceMetadata> element to an existing service behavior configuration:

```
<behaviors>
  <serviceBehaviors>
    <behavior name="MyServiceTypeBehaviors" >
      <serviceMetadata httpGetEnabled="true" />
    </behavior>
  </serviceBehaviors>
</behaviors>
```

2. Add the behavior configuration to the service:

```
<service name="MyNamespace.MyServiceType" behaviorConfiguration="MyServiceTypeBehaviors" >
```

Note: the service name must match the configuration name for the service implementation.

3. Add the following endpoint to your service configuration:

```
<endpoint contract="IMetadataExchange" binding="mexHttpBinding" address="mex" />
```

Note: your service must have an http base address to add this endpoint.

The following is an example service configuration file with metadata publishing enabled:


```
<configuration>
  <system.serviceModel>

    <services>
      <!-- Note: the service name must match the configuration name for the service implementation. -->
      <service name="MyNamespace.MyServiceType" behaviorConfiguration="MyServiceTypeBehaviors" >
        <!-- Add the following endpoint. -->
        <!-- Note: your service must have an http base address to add this endpoint. -->
        <endpoint contract="IMetadataExchange" binding="mexHttpBinding" address="mex" />
      </service>
    </services>

    <behaviors>
      <serviceBehaviors>
        <behavior name="MyServiceTypeBehaviors" >
          <!-- Add the following element to your service behavior configuration. -->
          <serviceMetadata httpGetEnabled="true" />
        </behavior>
      </serviceBehaviors>
    </behaviors>

  </system.serviceModel>
</configuration>
```

For more information on publishing metadata please see the following documentation: <http://go.microsoft.com/fwlink/?LinkId=65455>.

2. Установите нужные значения системных настроек. Для этого:
  - a. Перейдите в дизайнер системы, например, по кнопке .
  - b. В блоке “Настройка системы” перейдите по ссылке “Системные настройки”.
  - c. Укажите значения системных настроек:
 

“**ExchangeListenerServiceUri**”(код “ExchangeListenerServiceUri”). Формат значения настройки:

[ *адрес сервиса, используемый при установке* ]/api/listeners.

**“URL сервиса обработки событий Exchange в Creatio”** (код “BpmonlineExchangeEventsEndpointUrl”). Формат значения настройки: [ *адрес анонимного сервиса ExchangeListenerService* ]/NewEmail. Например, <https://mycreatio.com/0/ServiceModel/ExchangeListenerService.svc/NewEmail>.

## Диагностика настроек Exchange Listener

На странице диагностики настроек сервиса Exchange Listener представлены инструменты контроля и проверки работы сервиса.

Здесь вы можете:

- проверить подключение необходимой функциональности;
- убедиться в доступности сервиса;
- получить информацию о подписках;
- проверить правильность заполнения системной настройки “ExchangeListenerServiceUri”;
- проверить работоспособность почтового ящика.

Чтобы перейти на страницу диагностики настроек Exchange Listener, в адресной строке браузера добавьте “/0/ClientApp/#/IntegrationDiagnostics/ExchangeListener” к URL-адресу вашего сайта Creatio и нажмите Enter. Например:

```
http://mycreatio.com/0/ClientApp/#/IntegrationDiagnostics/ExchangeListener
```

Страница диагностики состоит из нескольких блоков с параметрами синхронизации, по которым можно запустить диагностику (Рис. 3). Чтобы отобразилась информация о параметрах подключения, необходимо в блоке с интересующим вас параметром нажать [ *Запустить диагностику* ].

Рис. 3 — Диагностика настроек Exchange Listener

## Диагностика сервиса Exchange Listener

[ЗАПУСТИТЬ ПОЛНУЮ ДИАГНОСТИКУ](#)

## Состояние доступности функциональности

- ✓ Синхронизация почты Exchange (ExchangeListenerEnabled)
- ✓ Синхронизация почты Exchange и IMAP (EmailIntegrationV2)
- ✗ Отправка писем (SendEmailsV2)

## Проверка доступности сервиса

[ЗАПУСТИТЬ ДИАГНОСТИКУ](#)

## Ошибка

Сервис ExchangeListener в данный момент не доступен для текущего сайта  
Unable to connect to the remote server

## Получение информации по подпискам

[ЗАПУСТИТЬ ДИАГНОСТИКУ](#)

## Ошибка

Http failure response for  
https://[redacted]/rest/MailDiagnosticToolsService/GetSubscribersInfo: 500 OK

## Проверка правильности заполнения системной настройки ExchangeListenerServiceUri

[ЗАПУСТИТЬ ДИАГНОСТИКУ](#)

## Успешная операция

URL сервиса обработки событий Exchange в Creatio  
https://[redacted]/ServiceModel/ExchangeListenerService.svc/ProcessFullEmail валиден

## Проверка работоспособности почтового ящика

Почтовый ящик \*



Отправить тестовое письмо

[ЗАПУСТИТЬ ДИАГНОСТИКУ](#)

## Ошибка

Unable to connect to the remote server

Состояние доступности функциональности	<p>Для данного блока диагностика запускается автоматически при открытии страницы.</p> <p>Выполняется проверка подключения в Creatio функциональности, необходимой для работы Exchange Listener:</p> <ul style="list-style-type: none"> <li>• ExchangeListenerEnabled;</li> <li>• EmailIntegrationV2;</li> <li>• SendEmailsV2.</li> </ul> <p>Подробнее о подключении функциональности читайте в статье <a href="#">Механизм отключения функциональности Feature Toggle</a> документации по разработке.</p>
Проверка доступности сервиса	Выполняется проверка доступности сервиса Exchange Listener из вашего приложения Creatio.
Получение информации по подпискам	Выполняется проверка соединения с удаленным сервером.
Проверка правильности заполнения системной настройки ExchangeListenerServiceUri	Выполняется проверка валидности адреса, указанного в системной настройке.
Проверка работоспособности почтового ящика	<p>Выполняется проверка работы почтовых ящиков MS Exchange.</p> <p>Установите признак [ <i>Отправить тестовое письмо</i> ], чтобы при запуске диагностики на указанный адрес было отправлено тестовое email-сообщение.</p>