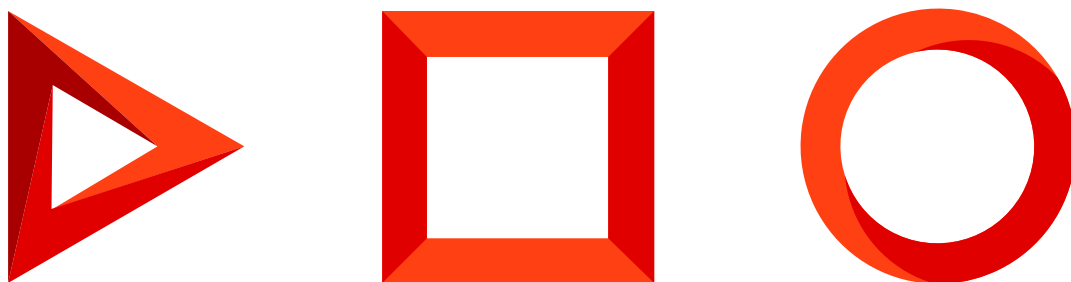


Массовый поиск дублей

Настроить массовый поиск дублей

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

Настроить массовый поиск дублей	4
Компоненты сервиса массового поиска дублей	4
Настроить переменные окружения	5
Запустить контейнеры	5
Проверить успешность запуска контейнеров	5
Проверить логирование	5
Подключить функциональность массового поиска дублей в Creatio	6
Обновить версию массового поиска дублей	11
Рекомендуемые работы по обслуживанию сервиса	11

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, @DeduplicationFeatureName, @DeduplicationFeatureCode)
    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, @ESDeduplicationFeatureName, @ESDeduplicationFeatureCode)
    INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allEmployeesId, 1, @ESDeduplicationFeatureId)
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 (@allEmployeesId, 1, @Bulk_ES_DD_FeatureId)
END;
ELSE
BEGIN
    SET @Bulk_ES_DD_FeatureId = NEWID()
    INSERT INTO Feature (Id, Name, Code) VALUES (@Bulk_ES_DD_FeatureId, @Bulk_ES_DD_FeatureName, @Bulk_ES_DD_FeatureCode)
    INSERT INTO AdminUnitFeatureState (SysAdminUnitId, FeatureState, FeatureId) VALUES (@allEmployeesId, 1, @Bulk_ES_DD_FeatureId)
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;

```



```

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" = ESDeduplication
    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, BulkESDeduplicationF
    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, DeduplicationFeatureName, DeduplicationFeatureCode);
  INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (
  END IF;

SELECT "Id" INTO ESDeduplicationFeatureId FROM "Feature"
WHERE "Code" = ESDeduplicationFeatureCode
LIMIT 1;
IF (ESDeduplicationFeatureId IS NOT NULL)
  THEN
    IF EXISTS (SELECT * FROM "AdminUnitFeatureState" WHERE "FeatureId" = ESDeduplicationFeatureId) THEN
      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, ESDeduplicationFeatureName, ESDeduplicationFeatureCode);
    INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (
  END IF;

SELECT "Id" INTO Bulk_ES_DD_FeatureId FROM "Feature"
WHERE "Code" = Bulk_ES_DD_FeatureCode
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_FeatureName, Bulk_ES_DD_FeatureCode);
    INSERT INTO "AdminUnitFeatureState" ("SysAdminUnitId", "FeatureState", "FeatureId") VALUES (
  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.