Creatio Academy

Containerized components

Global search

Version 7.17

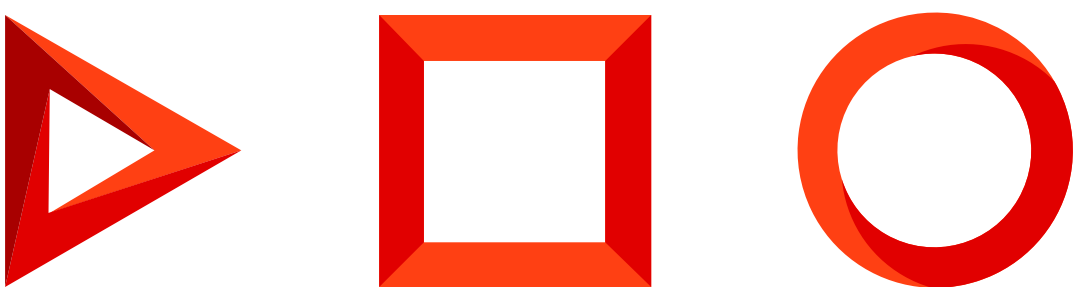

Table of Contents

Global search

PRODUCTS: ALL CREATIO PRODUCTS

The Global Search Service integrates ElasticSearch with Creatio. It performs the following functions:

- **Recording**:
  - Subscribes clients by creating an index in ElasticSearch and saves the connection between the index and the application.
  - Disconnects clients by removing their index in ElasticSearch.
- **Transporting**:
  - Participates in the indexing process by retrieving data from the database.

The sequence of actions during the global search setup depends on the global search version you are going to use. We recommend always using the latest version of the global search service with the latest version of Creatio.

You can deploy the components of global search version 3.0 using **Kubernetes orchestrator and Helm package manager** or **Docker**.

We also recommend backing up ElasticSearch daily to ensure the correct operation of the service and to enable the restoration of data after failures, e. g., power outages.

If you have any questions during the setup, we recommend consulting the Global search and deduplication FAQ.

Attention. You need repository access to set up the current version of the global search service. Contact Creatio support to verify your license and gain access to the repository.

Set up the global search service using Kubernetes

To set up the service, download the source files. Download files.

To install the service:

1. Set up the target environment:
   a. **Kubernetes cluster**. Learn more about setting up and managing the cluster in the Kubernetes documentation.
   b. **Helm package manager**. Learn more about installing the package manager in the Helm documentation.
2. Unpack the **values-onsite.yaml** file from the source file archive and save the file to the same directory as the archive.
3. Open the file. View the main parameters the file uses in the table below.
4. Specify the public service URL in the format of http://k8s-node:30332 in the `global.searchService.url` variable.

5. If you **have not installed** Redis, RabbitMQ, ElasticSearch, and PostgreSQL service database yet, proceed to the next step.

   If you **have already installed** these services, for example, when setting up other containerized components, disable the installation of the services and set up the connection to existing services in the **values-onsite.yaml** source file of the global search service.

   > **Note.** For highly loaded environments, we recommend deploying ElasticSearch in a cluster. Learn more in ElasticSearch documentation.

   - For **Redis**:
     - Disable the service installation. To do this, specify `enabled: false` in the `redis` section of the **values-onsite.yaml** file.

       ```
       redis:
         enabled: false
       ```

     - Set up the connection to the previously installed Redis. To do this, specify the connection parameters in the `global.redis` section of the **values-onsite.yaml** file.

       ```
       global:
         redis:
           host: [host]
           port: [port]
           database: [database]
       ```

       Where

       [host] is the address of the Redis server.

       [port] is the connection port of the Redis server.

       [database] is the name of the Redis database.

   - For **RabbitMQ**:
     - Disable the service installation. To do this, specify `enabled: false` in the `rabbitmq` section of the **values-onsite.yaml** file.

       ```
       rabbitmq:
         enabled: false
       ```

     - Set up the connection to the previously installed RabbitMQ. To do this, specify the connection parameters in the `global.rabbitmq` section of the **values-onsite.yaml** file.

```
global:
  rabbitmq:
    host: [host]
    vhost: [vhost]
    port: [port]
    user: [user]
    password: [password]
```

Where

[host] is the address of the RabbitMQ service.

[vhost] is the virtual host address of the RabbitMQ service.

[port] is the amqp connection port of RabbitMQ.

[user] is the RabbitMQ user.

[password] is the RabbitMQ user password.

- For **PostgreSQL** service database:
  - Disable the PostgreSQL installation. To do this, specify `enabled: false` in the `postgresql` section of the **values-onsite.yaml** file.

```
postgresql:
  enabled: false
```

  - Set up the connection to the PostgreSQL service database. To do this, specify the connection parameters in the `global.postgresql` section of the **values-onsite.yaml** file.

```
global:
  db:
    user: [user]
    password: [password]
    database: [database]
    host: [host]
    port: [port]
```

Where

[user] is the PostgreSQL user on whose behalf to connect to the database.

[password] is the PostgreSQL user password.

[database] is the PostgreSQL service database.

[host] is the address of the PostgreSQL database.

[port] is the port to connect to the database.

- For **ElasticSearch**:
  - Disable ElasticSearch installation. To do this, specify `enabled: false` in the `elasticsearch` section of the **values-onsite.yaml** file.

    ```
    elasticsearch:
      enabled: false
    ```

  - Set up the connection to the previously installed ElasticSearch. To do this, specify the connection parameters in the `global.elasticsearch` section of the **values-onsite.yaml** file.

    ```
    global:
      elasticsearch:
        url: [url]
        user: [user]
        password: [password]
    ```

    Where

    [user] is the ElasticSearch user.

    [password] is the ElasticSearch user password.

    [url] is the ElasticSearch service URL in the format of http://elasticsearch:9200.

6. Run the `helm install gs -f values-onsite.yaml globalsearch.tgz` command. As a result, Helm will install the global search service and selected dependencies.

> **Note.** By default, Helm deploys services with NodePort type.

The main parameters of the global search service the **values.yaml** file uses.

Parameter	Parameter description
scheduler.env.fillQueueInterval	The run interval for primary indexing, in milliseconds.
worker.env.indexingCommandTimeout	The timeout of the Creatio database query upon primary indexing, in seconds.
workerSingle.env.indexingCommandTimeout	The timeout of the Creatio database query upon instant indexing, in seconds.
global.incrementDays	The number of days within which to index the changed records during a single primary indexing iteration. Affects the indexing speed and Creatio database load. The higher the value, the faster the indexing and higher the load. The lower the value, the longer the indexing and lower the load.
log4Net	Logging settings.
global.indexingContentLength	The maximum length of text fields upon indexing.
global.elasticsearch	ElasticSearch connection parameters.
global.searchService	Public URL to the search-service in the format of http://k8s-node:30332.
global.rabbitmq	RabbitMQ connection settings.
global.db	The connection settings of the global search service's internal service database.
global.redis	Redis connection settings.

Set up the global search service in Docker

The global search requires 2 dedicated physical or virtual servers (“server 1” and “server 2”) with Linux installed. Use the requirements calculator to check the server requirements.

Attention. The global search setup procedure depends on the version you intend to install.

Learn more about the OS versions supported by Docker in the Docker documentation. Depending on your company needs, you can use either Docker Community Edition (CE) or Enterprise Edition (EE). Learn more in the Docker documentation.

Note. The procedure below is relevant for global search service version 3.0. If you need to set up an

earlier version of the global search, follow the procedure described in Creatio 7.16 documentation.

To **update** global search version 2.0 to version 3.0, run the docker-compose down -v command to delete all docker volumes on servers 1 and 2, then install and set up the services once again.

Global search components

Deploy on server 1:

- elasticsearch. The search engine.

Deploy on server 2:

- postgres. The database for configuring the global search components.
- rabbitmq. The message broker.
- redis. The database used for caching and performance improvement.
- gs-web-api. The web service that configures global search components.
- gs-web-indexing-service. The web service that processes requests to perform targeted indexing of Creatio data.
- gs-search-service. An elasticsearch proxy web service for data search.
- gs-scheduler. The scheduler of Creatio data indexing in ElasticSearch.
- gs-worker. The component that indexes Creatio data in ElasticSearch as per the scheduler tasks.
- gs-worker-replay. The component that processes the indexing results (gs-worker results).
- gs-worker-single. The component that performs the targeted indexing of business process data in ElasticSearch upon a request from the business process.
- gs-worker-single-replay. The component that processes exceptions as part of the targeted indexing (gs-worker-single results).
- gs-worker-single-task. The component that schedules tasks for gs-worker-single.
- gs-worker-queried-single-task. The component that generates tasks for gs-worker-single.

To set up the components, download the source files. Download files.

The list of ports used by global search components:

Attention. If you are using a firewall, make sure the ports below are open and available.

Component name	Outgoing port	Incoming port	Notes
gs-web-api		81	Configure the incoming port using the WEB_API_PORT variable
gs-web-indexing-service		82	Configure the incoming port using the WEB_INDEXING_SERVICE_PORT variable
gs-search-service	9200	83	Configure the incoming port using the SEARCH_SERVICE_PORT variable
gs-worker	9200		Requires connection to the server where elasticsearch is located.
gs-worker-single	9200		Requires connection to the server where elasticsearch is located.
elasticsearch		9200	

Global search setup procedure

1. Install Docker on a physical or virtual machine running Linux OS.
2. Install Docker-Compose.
3. Install ElasticSearch.
4. Set up the container variables.
5. Install and run the Global Search Service components.
6. Enable the global search functionality in Creatio.

Install Docker

Install Docker on Linux to deploy global search components. The installation is covered in the Docker documentation.

Run the **docker --version** command on a Linux machine to verify the installed Docker version.

Install Docker-Compose

The installation of Docker-Compose is covered in the Docker documentation.

Install ElasticSearch

Note. This guide covers the procedure for deploying ElasticSearch in Docker-Compose. You can also

deploy ElasticSearch as OS daemon, which does not involve installing Docker and Docker-Compose. Learn more about the setup procedure in the ElasticSearch documentation.

To install ElasticSearch:

1. Go to the ElasticSearch installation server (server 1) and open the /opt directory.
2. Download and unpack the archive with setup files to the directory. Download the archive.
3. Open the /opt/docker-compose/elasticsearch directory (where the components are located) and run the following command:

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

   The command might take up to several minutes to complete.
4. Make sure that the log files do not contain any errors after the command is complete. To do this, run the following command:

   ```
   docker logs es-01
   ```

Set up the container variables

Configure the global search component containers via the file that contains the environment variables. The variables are stored in the /opt/compose/services/.env file. Edit this file to set the variables.

Variable name	Details	Default value
GS_ES_URL	The external ElasticSearch host required for access from Creatio. Specify the IP address of the server where the ElasticSearch is deployed.	http://elasticsearch-publicip:9200
CURRENT_SERVER_IP	The external IP address of the server where the global search components are deployed (server 2).	10.0.0.1

Note. To check the external IP address of the server, run the `hostname -I | awk '{ print $1 }'` command.

Additional variables that control the data indexing parameters in ElasticSearch

Variable name	Details	Default value
GS_DB_INCREMENT_DAYS	Number of days to index per one scheduler iteration. ModifiedOn columns of Creatio records are used for comparison.	500 days
GS_DB_FILL_QUEUE_INTERVAL	Creatio database data collection interval for the regular scheduler. The lower the variable, the higher the load on Creatio database and the faster the primary indexing.	30000 (specified in milliseconds)

Run containers that have Global Search Service components

Attention. For the correct container operation, the UTC time on the Linux machine with Docker installed must correspond to the UTC time on the Creatio database server. The permissible deviation is up to five minutes. Otherwise, the global search might not index all records.

1. Go to the server that has the global search components (server 2) and open the /opt folder.
2. Download and unpack the archive with setup files to the opened folder. Download the archive.
3. Open the /opt/compose/services folder and run the following command:

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

Verify that containers ran successfully

To see the running global search containers, use the following console command:

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

All currently running containers display the **Up** status.

Logging

By default, the container logging takes place in the stdout and stderr.

Note. The “docker logs --tail 100 gs-worker” command displays 100 last strings from the gs-worker container.

Note. The mysql or rabbitmq containers might become temporarily unavailable on start because they run after the rest of the containers. In this case, wait until a notification about the successful container connection and start appears in the log files: “Now listening on: http:/[*::*]80 Application started. Press Ctrl+C to shut down.”

Connect the global search service to Creatio

Actions on the server

To connect global search to Creatio, take the following steps on the server where the global search components are located (server 2 for the service deployed in Docker):

1. Install the api-get install curl or yum install curl utility for HTTP queries.

```
apt-get install curl
```

2. Execute the HTTP request to register the site in global search. Specify the following:
   a. [*DATABASE_TYPE*]: Creatio database type (mssql, postgresql, or oracle).
   b. [*DATABASE_CONNECTION_STRING*]: Creatio database connection string
   c. [*SITE_NAME*]: Creatio site name, e. g., my-test-site.
   d. [*SERVER2_IP_ADDRESS*] (only for Docker): the IP address of the Linux server where the global search components are deployed.

      [*GS_WEB_API_URL*] (only for Kubernetes): the IP address of the Linux server where the global search components are deployed.

      Request for Docker

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

      Request for Kubernetes

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

      Example for Microsoft SQL.

      For Docker:

      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

For Kubernetes:

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:// [GS_WEB_API_URL] /sites/my-test-site

Example for PostgreSQL. server=[SERVER_IP];port=5432;database=[DB_NAME];user id=[USER_NAME];password=[PASSWORD];timeout=10;commandtimeout=400;maxpoolsize=1024

3. Execute the HTTP request to connect the search to the site. Specify the following:

   a. [*SITE_NAME*]: Creatio site name, e. g., my-test-site.

   b. [*TEMPLATE_NAME*]: the name of the search template used in ElasticSearch. View the available templates in the table below.

   c. [*SERVER2_IP_ADDRESS*] (only for Docker): the IP address of the Linux server where the global search components are deployed.

   [*GS_WEB_API_URL*] (only for Kubernetes): the IP address of the Linux server where the global search components are deployed.

Request for Docker

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

Request for Kubernetes

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

Example.

For Docker:

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

For Kubernetes:

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

Note. This request returns the URL of the index created in ElasticSearch. Save the URL and use it in the system setting installation SQL script below.

Attention. To change the search template, run the DELETE query at /sites/{siteName}/search, as well as the search connection HTTP request described above. This reindexes the entire site.

View the available search templates and their features in the table below:

	Old template (version 1.6)	default.json	ngram_2.json	ngram_3.json	without_ng
Search by partial match	+	–	+	+	–
Search by misspelled words	+	–	+	+	–
Search communication options by phone number (partial match)	+	+	+	+	–
Search communication options (partial match)	+	+	+	+	–
Search by word swapping	+	+	+	+	+
Search by exact match	+	+	+	+	+
Search by two characters	–	–	+	–	–
Average search speed (lower is better)		1x	13x	7x	<1x
Index size at elasticsearch (lower is better)		1x	4x	2.5x	<1x
Primary indexing time (lower is better)		1x	1.8x	1.4x	<1x

Settings in Creatio for Microsoft SQL DBMS

1. Toggle the global search (GlobalSearch, GlobalSearch_V2, GlobalSearchRelatedEntityIndexing) feature in Creatio by running the following SQL script:

```
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. Set the values of the system settings:

   a. “GlobalSearchUrl:” the full path to ElasticSearch, including the index. The web-api returns this value if you request it to add site search.

      Example string for Docker: http://[SERVER2_IP_ADDRESS]:83/indexname.

      Example string for Kubernetes: http://[GS-SEARCH-SERVICE_URL] /indexname.

   b. “GlobalSearchConfigServiceURL:” the global search API URL.

      Default value for Docker: http://[SERVER2_IP_ADDRESS]:81.

      Default value for Kubernetes: http:// [GS_WEB_API_URL].

   c. “GlobalSearchIndexingApiUrl:” the instant indexing service URL.

      Default value for Docker: http://[SERVER2_IP_ADDRESS]:82.

      Default value for Kubernetes: http://[GS-WEB-INDEXING-SERVICE_URL].

      For Docker

```
UPDATE SysSettingsValue
SET TextValue = [specify the URL to the ElasticSearch index, use a string of the following
WHERE SysSettingsId = (SELECT TOP 1 Id FROM SysSettings WHERE Code ='GlobalSearchUrl')

UPDATE SysSettingsValue
SET TextValue = [specify the URL to the Global Search Service, use a string of the followi
WHERE SysSettingsId = (SELECT TOP 1 Id FROM SysSettings WHERE Code ='GlobalSearchConfigServ

UPDATE SysSettingsValue
SET TextValue = [specify the URL to the Global Serch Indexing Service, use a string of the
WHERE SysSettingsId = (SELECT TOP 1 Id FROM SysSettings WHERE Code ='GlobalSearchIndexingAp
```

      For Kubernetes

```
UPDATE SysSettingsValue

SET TextValue = [specify the URL to the ElasticSearch index, use a string of the following
```

```
WHERE SysSettingsId = (SELECT TOP 1 Id FROM SysSettings WHERE Code ='GlobalSearchUrl')



UPDATE SysSettingsValue

SET TextValue = [specify the URL to the Global Search Service, use a string of the followi

WHERE SysSettingsId = (SELECT TOP 1 Id FROM SysSettings WHERE Code ='GlobalSearchConfigSer



UPDATE SysSettingsValue

SET TextValue = [specify the URL to the Global Search Indexing Service, use a string of the

WHERE SysSettingsId = (SELECT TOP 1 Id FROM SysSettings WHERE Code ='GlobalSearchIndexingA
```

3. Restart Creatio, flush Redis, and log in to the application.

Settings in Creatio for Oracle DBMS

1. Toggle the global search (GlobalSearch, GlobalSearch_V2, GlobalSearchRelatedEntityIndexing) feature in Creatio by running the following SQL script:

```
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. Set the values of the system settings:

   To do this, run the following script:

   a. “GlobalSearchUrl:” the full path to elasticsearch, including the index. The web-api returns this value if you request it to add site search.

      Example string for Docker: http://[SERVER2_IP_ADDRESS]:83/indexname.

      Example string for Kubernetes: http://[GS-SEARCH-SERVICE_URL] /indexname

   b. “GlobalSearchConfigServiceURL:” the global search API URL.

      Default value for Docker: http://[SERVER2_IP_ADDRESS]:81.

      Default value for Kubernetes: http:// [GS_WEB_API_URL].

   c. “GlobalSearchIndexingApiUrl:” the instant indexing service URL.

      Default value for Docker: http://[SERVER2_IP_ADDRESS]:82.

      Default value for Kubernetes: http://[GS-WEB-INDEXING-SERVICE_URL].

```
For Docker

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) := '[specify the URL to the ElasticSearch index, use a string of the fo
  CONFIG_URL VARCHAR(500) := '[specify the URL to the Global Search Service, use a string of
  IND_API_URL VARCHAR(500) := '[specify the URL to the Global Search Indexing Service, use a
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;
```

For Kubernetes

```
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) := '[specify the URL to the ElasticSearch index, use a string of the fol
 CONFIG_URL VARCHAR(500) := '[specify the URL to the Global Search Service, use a string of t
 IND_API_URL VARCHAR(500) := '[specify the URL to the Global Search Indexing Service, use a s

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. Restart Creatio, flush Redis, and log in to the application.

Settings in Creatio for PostgreSQL DBMS

1. Toggle the global search (GlobalSearch, GlobalSearch_V2, GlobalSearchRelatedEntityIndexing) feature in Creatio by running the following SQL script:

```
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" = GS_RelatedEntityInde
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. Set the values of the system settings:

   a. “GlobalSearchUrl:” the full path to ElasticSearch, including the index. The web-api returns this value if you request it to add site search.

      Example string for Docker: http://[SERVER2_IP_ADDRESS]:83/indexname.

      Example string for Kubernetes: http://[GS-SEARCH-SERVICE_URL] /indexname.

   b. “GlobalSearchConfigServiceURL:” the global search API URL.

      Default value for Docker: http://[SERVER2_IP_ADDRESS]:81.

      Default value for Kubernetes: http:// [GS_WEB_API_URL].

   c. “GlobalSearchIndexingApiUrl:” the instant indexing service URL.

      Default value for Docker: http://[SERVER2_IP_ADDRESS]:82.

      Default value for Kubernetes: http://[GS-WEB-INDEXING-SERVICE_URL].

   To do this, run the following script:

```
For Docker

UPDATE "SysSettingsValue"
SET "TextValue" = [specify the URL to the ElasticSearch index, use a string of the following
WHERE "SysSettingsId" = (SELECT "Id" FROM "SysSettings" WHERE "Code" = 'GlobalSearchUrl' LIMI

UPDATE "SysSettingsValue"
SET "TextValue" = [specify the URL to the Global Search Service, use a string of the followin
WHERE "SysSettingsId" = (SELECT "Id" FROM "SysSettings" WHERE "Code" = 'GlobalSearchConfigSer

UPDATE "SysSettingsValue"
SET "TextValue" = [specify the URL to the Global Search Indexing Service, use a string of the
WHERE "SysSettingsId" = (SELECT "Id" FROM "SysSettings" WHERE "Code" = 'GlobalSearchIndexingA
```

For Kubernetes

```
UPDATE "SysSettingsValue"

SET "TextValue" = [specify the URL to the ElasticSearch index, use a string of the following

WHERE "SysSettingsId" = (SELECT "Id" FROM "SysSettings" WHERE "Code" = 'GlobalSearchUrl' LIMI



UPDATE "SysSettingsValue"

SET "TextValue" = [specify the URL to the Global Search Service, use a string of the followin

WHERE "SysSettingsId" = (SELECT "Id" FROM "SysSettings" WHERE "Code" = 'GlobalSearchConfigSer



UPDATE "SysSettingsValue"

SET "TextValue" = [specify the URL to the Global Search Indexing Service, use a string of the

WHERE "SysSettingsId" = (SELECT "Id" FROM "SysSettings" WHERE "Code" = 'GlobalSearchIndexingA
```

3. Restart Creatio, flush Redis, and log in to the application.