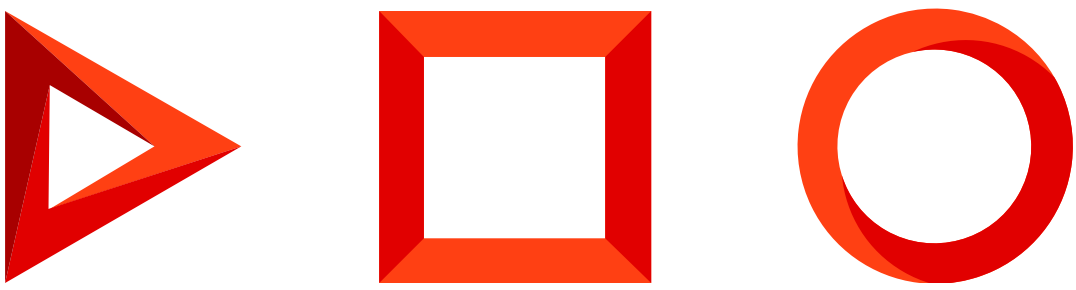
Creatio Academy

Additional setup

Version 7.17



This documentation is provided under restrictions on use and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this documentation, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

© 2023 Creatio. All rights reserved.

Table of Contents

© 2023 Creatio. All rights reserved.

Google integration

PRODUCTS: ALL CREATIO PRODUCTS

To set up Google integration on a Creatio on-site application:

1. Set up a Google account.
2. Enable access to Calendar API.
3. Generate integration keys “Client ID” and “Client Secret”.
4. Enter the keys in Creatio as system settings:

   Detailed instructions are available in the “Application registration for integration with Google” article.

Set up landing page integration

PRODUCTS: ALL CREATIO PRODUCTS

This functionality is available in all configurations containing **the Landing pages and web forms section**.

Customers who have their Creatio application deployed on-site may need to perform additional setup to have the HTML code displayed correctly on the landing page. It is required when according to URL safety rules the URL displayed in the user's browser must be different from the one used for external access to Creatio. For example, when the URL gets blocked by the firewall.

To set up landing pages:

1. Go to System Designer → [*System settings*].
2. Open the “**Landing pages data collection service URL**” system setting in the [*Landing pages section settings*] folder.
3. In the [*Default value*] field, enter the **external URL** of your Creatio application, for example, http://creatio-marketing.mydomain.com, and save your settings.

As a result, the HTML code embedded in your landing page will use the correct URL to call the web service for creating a new lead in Creatio, for example:

```
serviceUrl: "http://mysite.creatio-marketing/ServiceModel/GeneratedWebFormService.svc/SaveWebFor
```

If you use a **secure connection protocol**, enter the URL and specify https:// in it. The web service call address, in this case, will be as follows:

```
serviceUrl: "https://mysite.creatio-marketing/ServiceModel/GeneratedWebFormService.svc/SaveWebFo
```

© 2023 Creatio. All rights reserved.

Note. By default, this setting is not configured and the application URL is generated automatically.

Case resolution notifications

PRODUCTS: SERVICE CREATIO

Service Creatio enterprise edition, **Service Creatio customer center edition**, and **Financial Services Creatio, customer journey edition** products can send your customers email notifications, informing them about changes in their support case status.

To enable automatic email notifications on case resolution:

1. Go to System Designer → [*System settings*].
2. Open the “**Website URL**” system setting.
3. Specify the full URL of your Creatio website in the [*Value*] field, e.g., http://mydomain.com.
4. Click [*Save*].

Data enrichment service

PRODUCTS: ALL CREATIO PRODUCTS

Your personal cloud key and the URL to Creatio cloud services are required to use data enrichment. Use the following system settings to specify these values:

- ”Account enrichment service URL”. By default, this setting is populated for all applications.
- “Creatio cloud services API key”. This setting is populated for **cloud** applications by default but needs to be configured for **on-site** applications.

Request your personal key for your **on-site** application from the Creatio support. After receiving the key:

1. In the system designer, click the [*System settings*] link.
2. Go to the [*Creatio cloud services*] group, and select the [*Creatio cloud services API key*] system setting.
3. Specify the key in the [*Default value*] field and click [*Save*]

   Data enrichment functions can now be used.

Bulk emails

PRODUCTS: MARKETING

The functionality is available in **Creatio marketing** and CRM bundles.

Set up your email service integration with Creatio for sending bulk emails. All cloud email service settings for bulk emails are consolidated on the bulk email setup page in the [*Email*] section. You can use it to edit general settings of sending bulk emails and receiving responses, sender domains as well as to monitor the connection

© 2023 Creatio. All rights reserved.

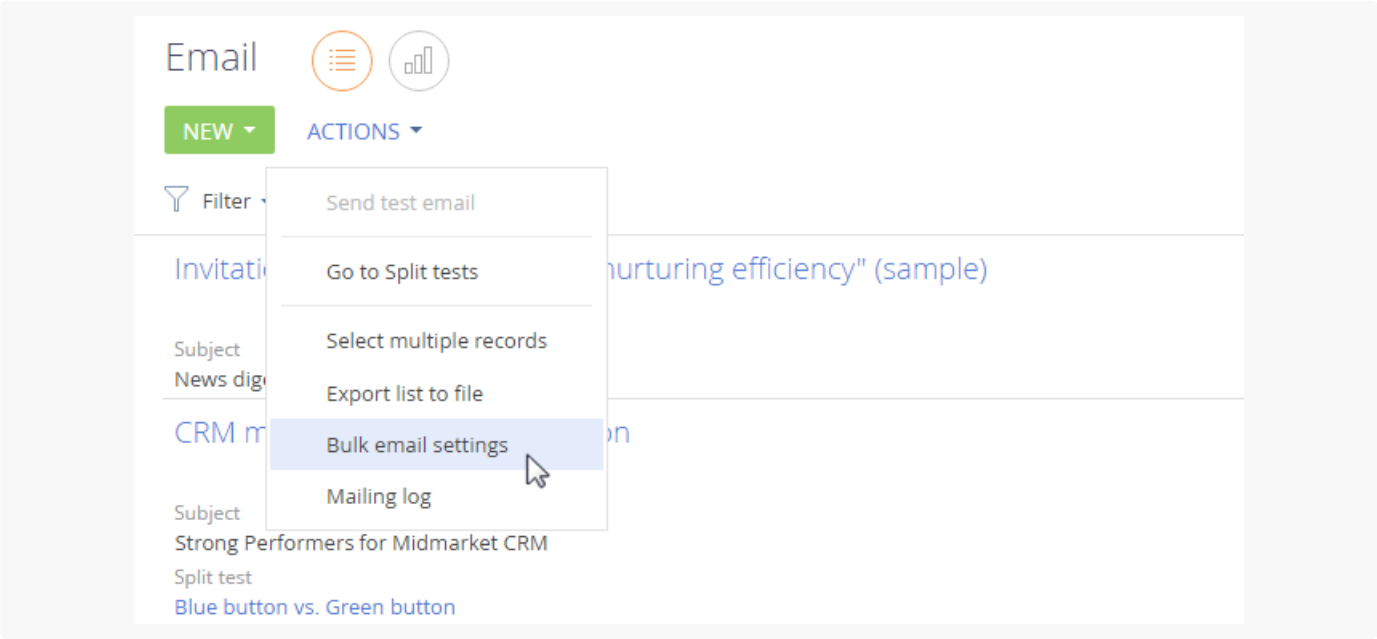
status.

Integration with Creatio cloud email service (for on-site users)

To check integration with cloud email service:

1. Go to the [*Email*] section. Open the [*Actions*] menu and select **Bulk email settings** (Fig. 1).

Fig. 1 Open the bulk email settings page



2. Fill out the [*General settings*] tab fields.

Attention. Contact Creatio support to change your bulk email service provider.

a. In the **Domain to receive responses** field, specify the domain address of your Creatio application in the following format: http://www.yourdomain.com.

Important. POST requests are always over port 443 regardless of the port on which Creatio is available. We recommend checking the connection to port 443 after you complete the setup. To do this, open the Creatio cloud services connection URL in the “https://url_address.com” format in the browser.

As a result, a blank page should open. If the page will not open, check whether the port is opened correctly.

b. In the **API key** field, specify your personal access key to the bulk email service.

c. In the **Creatio cloud services connection URL** field, specify the bulk email cloud service address in the “https://url_address.com” format.

d. In the **Auth key** field, specify the authentication key for receiving responses.

To obtain the API key and the Auth key, as well as the URL to bulk email cloud services after installing

© 2023 Creatio. All rights reserved.

product licenses, please contact our support at support@creatio.com.

e. Creatio will populate the **Email provider** field with the name of your email service provider after you fill out the [*API key*] and [*Creatio cloud services connection URL*] fields.

Sender domain list

For the sender name to be displayed correctly in the bulk email and to avoid unauthorized bulk email sent on your behalf, perform the following settings:

- Specify the list of your domains on the bulk email settings page.
- Verify each domain by using specific text SPF-, DKIM- and DMARK-records.
- Save the changes.

> **Note.** Only custom email domains can be verified. Public domains (for example, gmail.com, yahoo.com, etc.) cannot be verified. We do not recommend using public domains for bulk emails. Such emails have a high risk of being marked as spam and ruining the reputation of the sender IP address.

To do this:

1. Add the list of your domains by clicking the + button on the [*Sender domains*] tab.

   > **Note.** All added domains, including those that are no longer in use, are displayed in the list. Domains cannot be deleted from the list.

2. Select a domain from the list for verification. A DKIM/SPF setup manual for the selected domain will be displayed on the right side of the screen. The manual text will contain correct SPF and DKIM records generated for your domains.

   > **Note.** DKIM/SPF manuals are different for each domain. To view a specific manual, select the required domain from the list.

3. Set up domain verification. Learn more: Recommendations on setting up the popular DNS providers.

   As a result, the bulk email settings [*Connection status*] field will display the ● “Connection active” message.

Additional settings for integration with bulk email service

Set up one of the Creatio access options for Creatio Cloud Email Service for the correct operation of bulk email functions:

1. In the server firewall, permit receiving POST requests from the Internet to the domain where Creatio is deployed: http://www.yourdomain.com.
2. In the server firewall, permit receiving POST requests from a specific web service. For example, if the application is deployed on http://www.yourdomain.com, then the following address must be accessible: http://www.yourdomain.com/0/ServiceModel/CESWebhooksService.svc/HandleWebHooks.

© 2023 Creatio. All rights reserved.

Note. There is no need to set up processing of unsubscribe requests and to check if the Creatio application server is able to receive GET-requests. Creatio will process unsubscribe queries automatically.

Attention. If the HTTPS protocol is used to access Creatio, the application server must have an active certificate installed. In case the data transfer protocol or application address is changed, make the appropriate changes on the bulk email setup page.

It is not recommended to use IP address whitelists to limit access to open ports because the Creatio Cloud Email Service may send analytical information about responses from different IP addresses. If the whitelist does not contain the IP address that the analytical information is sent from, the data will be lost.

When using blacklists, we recommend checking that the received IP addresses are not on this list.

Bulk email monitoring on-site

We recommend that you set up monitoring of your bulk email status by the support service before you start working with bulk emails. If you do this, Creatio support will be able to resolve any potential bulk email issues faster. Support service employees will have access to aggregated bulk email metrics that do not contain personalized email message texts, email templates, etc.

Note. The procedure is different for Creatio cloud and on-site. Learn more about the setup procedure for Creatio cloud: Permit monitoring the email status by Creatio support.

The setup procedure is as follows:

1. Go to the system designer by clicking the button in the top right corner of the application window and click [*System settings*].
2. Open the [*Enable monitoring of the email troubleshooting indicators*] system setting and select the [*Default value*] checkbox. Save the changes.
3. In the application server firewall, permit access from the Internet to the web service:

   /0/ServiceModel/CESTroubleshootingService.svc/emailstate.

   For example, if the application is deployed on http://www.yourdomain.com, then the following address must be accessible:

   http://www.yourdomain.com/0/ServiceModel/CESTroubleshootingService.svc/emailstate.

   As a result, the support service employees will be able to identify and eliminate potential bulk email issues.

Application server web farm

PRODUCTS: ALL CREATIO PRODUCTS

You can enhance the performance of large-scale Creatio projects (up to several thousand users) through

© 2023 Creatio. All rights reserved.

horizontal scaling, i. e., by increasing the number of servers with deployed Creatio applications and setting up workload distribution between them.

The load balancer may be either hardware or software. To work in fault-tolerant mode, use an HTTP/HTTPS traffic balancer that supports the WebSocket protocol. Creatio has been tested on the HAProxy software load balancer. There are cases of successful implementation of other balancers, e. g., Citrix, Cisco, NginX, FortiGate, Microsoft ARR.

Note. The installation procedure of Marketplace add-ons and custom improvements for an environment that uses a balancer differs from the standard deployment process. Learn more in a separate article: Install applications from the Marketplace.

This guide covers horizontal scaling of Creatio using a free open-source load balancer (HAProxy), designed for distributing the load between several application servers.

Note. Synchronize the server time of the nodes (servers and computers) that run deployed application instances to ensure smooth operation of Creatio.

General deployment procedure

Creatio .NET Framework

To deploy Creatio using the horizontal scaling of **.NET Framework** application servers:

1. Deploy the required number of Creatio application instances in a web farm.

   Note. We recommend specifying identical names in IIS and the Application pool setting for all Creatio instances.

2. Specify identical SQL and Redis databases in the ConnectionStrings.config file for all instances.

   ```
   <add name="redis" connectionString="host=DOMAIN.COM;db=0;port=6379;maxReadPoolSize=10;maxWrit
   <add name="db" connectionString="Data Source=DOMAIN.COM;Initial Catalog=DatabaseName;Integrat
   ```

3. Add the following key in the <appSettings> block of the application's Web.config file:

   ```
   <add key="TenantId" value="1" />
   ```

   The "value" number must be identical for all Creatio instances of the web farm.

   Attention. Starting with Creatio version 7.14.1, the <add key="TenantId" value="..."/> key can only be added to the internal Web.config file (Terrasoft.WebApp\Web.config). Adding the key to an external

© 2023 Creatio. All rights reserved.

Web.config file may lead to application failures.

4. Generate a unique machineKey value for one of Creatio instances. Learn more in a separate article: Modify Web.config. Copy the resulting value and specify it in the Web.config files of each Creatio instance. You can locate the files in the root Creatio folder and the Terrasoft.WebApp folder.
5. Turn on clustering for all schedulers in the <quartzConfig> block of every node's external configuration file (Web.config):

```
<add key="quartz.jobStore.clustered" value="true" />
<add key="quartz.jobStore.acquireTriggersWithinLock" value="true" />
```

6. If the instanceId settings collide, specify unique values for each scheduler node.

   The **ways to specify** unique instanceId values are as follows:

   - Add the following string to all schedulers in the <quartzConfig> block of every node's external configuration file (Web.config):

     ```
     <add key="quartz.scheduler.instanceId" value="AUTO" />
     ```

     Attention. The "AUTO" value of the "value" attribute must be uppercase. Otherwise, Creatio will treat the value as the node name, which may lead to errors in the scheduler's operation.

     As a result, the scheduler will generate the unique node name based on the <node name>+timestamp template.
   - Add unique quartz.scheduler.instanceId values manually.
7. Set the "value" attribute of the quartz.jobStore.clustered setting to "true."

```
<add key="quartz.jobStore.clustered" value="true" />
```

8. Grant access permissions to created application directories for the IUSR user and the user who launches the Application pool in IIS.
9. Set up a load balancer (e. g., HAProxy) to distribute the workload between the deployed application servers.
10. If necessary, set up load balancing for database and session servers.

Note. Learn more about setting up clustering in the Microsoft SQL and Oracle documentation. Learn more about setting up fault tolerance using Redis Cluster in a separate article: Redis Cluster.

Creatio .NET Core and .NET 6

To deploy Creatio using the horizontal scaling of **.NET Core and .NET 6** application servers:

© 2023 Creatio. All rights reserved.

1. Deploy the required number of Creatio application instances.
2. Specify identical SQL and Redis databases in the ConnectionStrings.config file for all instances.
3. Go to the root directory of any Creatio instance and locate the Terrasoft.WebHost.dll file.
4. Run the following command:

   ```
   dotnet Terrasoft.WebHost.dll configureWebFarmMode
   ```

   As a result, configuration files of the current application instance will be updated.
5. Enable clustering for all schedulers in the <quartzConfig> block of every node's external configuration file (Terrasoft.WebHost.dll):

   ```
   <add key="quartz.jobStore.clustered" value="true" />
   <add key="quartz.jobStore.acquireTriggersWithinLock" value="true" />
   ```
6. If the instanceId settings collide, specify unique values for each scheduler node.

   The **ways to specify** unique instanceId values are as follows:

   - Add the following string to all schedulers in the <quartzConfig> block of every node's external configuration file (Terrasoft.WebHost.dll):

     ```
     <add key="quartz.scheduler.instanceId" value="AUTO" />
     ```

     Attention. The "AUTO" value of the "value" attribute must be uppercase. Otherwise, Creatio will treat the value as the node name, which may lead to errors in the scheduler's operation.

     As a result, the scheduler will generate the unique node name based on the <node name>+timestamp template.
   - Add unique quartz.scheduler.instanceId values manually.
7. Set the "value" attribute of the quartz.jobStore.clustered setting to "true."

   ```
   <add key="quartz.jobStore.clustered" value="true" />
   ```
8. If necessary, set up load balancing for the database and session servers.
9. Copy all configuration files of the current application instance to the root folders of other application instances.
10. Set up a load balancer (e. g., HAProxy) to distribute the workload between the deployed application servers.

Note. Learn more about setting up clustering in the DBMS vendor documentation. Learn more about setting up fault tolerance using Redis Cluster in a separate article: Redis Cluster.

© 2023 Creatio. All rights reserved.

Install the HAProxy balancer

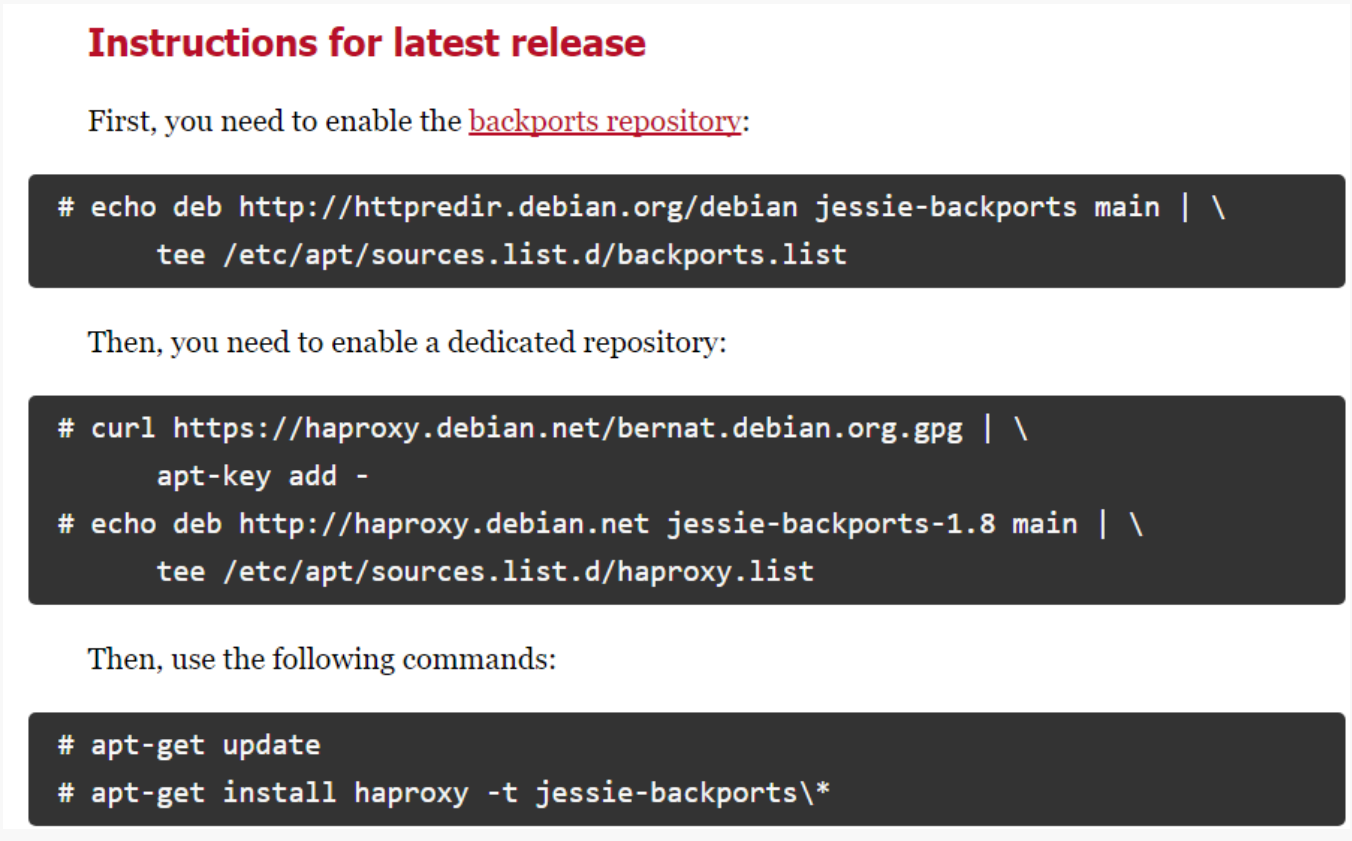
The HAProxy load balancer supports a range of free open-source OS. This guide covers one of the simpler methods of deploying HAProxy on the Debian OS via the haproxy.debian.net service.

1. Open the installation service page by clicking https://haproxy.debian.net/.
2. Select the OS and its version, as well as the HAProxy version.

> **Note.** Use the cat /etc/issue command to check the currently installed Debian version.

As a result, the service will generate a set of HAProxy installation commands to run in the Debian OS.

Fig. 1 HAProxy installation commands generated by the haproxy.debian.net service



3. Run the generated commands one after another.

Set up the HAProxy balancer

To set up HAProxy, modify the haproxy.cfg file. Follow this path to locate the file:

```
.../etc/haproxy/haproxy.cfg
```

© 2023 Creatio. All rights reserved.

Primary setup (required)

Add the sections required for HAProxy operation: **frontend** and **backend**.

The frontend section

Add the following settings to the frontend section: **bind** and **default_backend**.

- Specify the address and the port that will receive requests distributed by HAProxy in the **bind** setting.
- Specify the name that will match the name of the backend section in the **default_backend** option.

As a result, the setting will look as follows:

```
frontend front
maxconn 10000
#Using these ports for binding
bind *:80
bind *:443
#Convert cookies to be secure
rspirep ^(set-cookie:.*)  \1;\ Secure
default_backend creatio
```

The backend section

Add the following required settings to the backend section:

- Specify the type of balancing, e. g., **roundrobin**, in the **balance** parameter. Learn more about the different types of balancing in the HAProxy documentation.
- Use the **server** parameter to specify all servers (or nodes) that distribute the load.

Add a unique “server” parameter that contains the server address, port address, and weight for each server (i. e. the deployed Creatio instance). The server weight enables the balancer to distribute the load based on the physical capabilities of the servers. The higher weight you specify for the server, the more requests it will receive. For example, if you need to distribute the load between 2 Creatio application servers, add 2 “server” parameters to backend:

```
server node_1 [server address]:[port] weight
server node_2 [server address]:[port] weight
```

As a result, the setting will look as follows:

```
backend creatio
#set balance type
balance roundrobin

server node_1 nodeserver1:80 check inter 10000 weight 2
```

© 2023 Creatio. All rights reserved.

```
server node_2 nodeserver2:80/sitename check inter 10000 weight 1
```

The new settings will be applied as soon as you restart HAProxy. Use the following command to restart HAProxy:

```
service haproxy restart
```

Check the server status

The HAProxy balancer works with the following server statuses:

Status	Description
UP	The server is operational.
UP - transitionally DOWN	The server is considered operational at the moment, but the last health check has failed. As a result, the server is currently switching to the DOWN status.
DOWN - transitionally UP	The server is not considered operational at the moment, but the last health check has succeeded. As a result, the server is currently switching to the UP status.
DOWN	The server is not operational.

Health checks initiate changes in a server's operational status. The simplest health check requires adding the "check" keyword to the server setup string. Running the health check requires the server's IP and TCP port. Health check example:

```
server node1 ... check
option httpchk GET /Login/NuiLogin.aspx
option httpchk GET /0/ping
```

Set up web stats (optional)

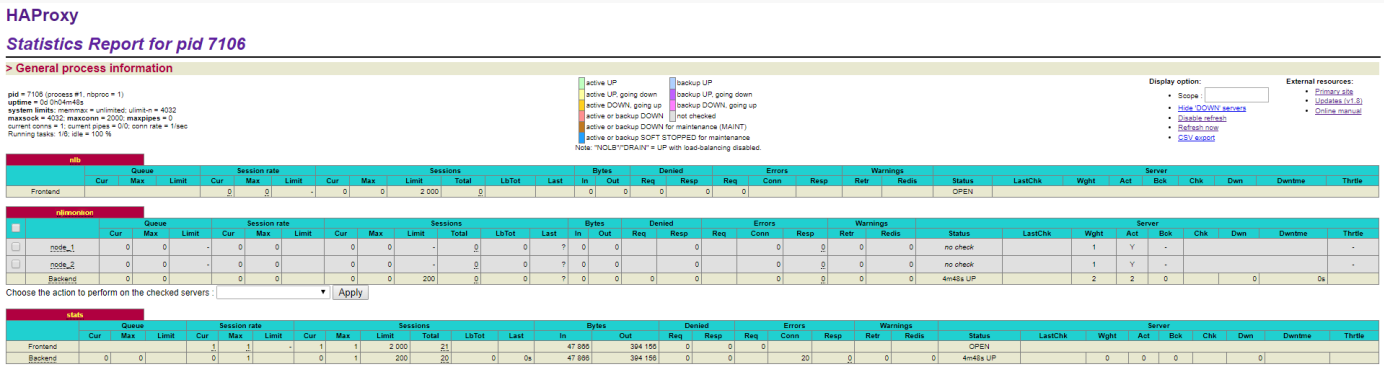
To turn on web stats, add a new listen section that contains the following parameters: **bind**, **mode http**, **stats enable**, **stats uri**. The section will look as follows:

```
listen stats # Define a listen section called "stats"
    bind :9000 # Listen on localhost:9000
    mode http
    stats enable  # Enable stats page
    stats uri /haproxy_stats  # Stats URI
```

© 2023 Creatio. All rights reserved.

As a result, you will be able to view the web stats of Creatio load balancing in the browser.

Fig. 2 The web stats of the load balancer



To view the stats, follow the path: [*balancer address*]:9000/haproxy_stats.

Set up the IP addresses in the audit log for .NET Core and .NET 6 (optional)

With a web farm, user requests reach the servers through a load balancer and/or a proxy server. As such, by default, the audit log displays the IP address of the proxy that forwarded the request last, not the actual IP address of the user.
You can configure the audit log so that it displays the actual IP address of the user. To do this:

1. Configure the balancer so that each request it forwards to one of the Creatio application instances has a header with “ForwardedForHeaderName” name and the user’s IP address value.
2. Modify the configuration files of Creatio application instances accordingly.
   a. Go to Creatio root directory and open appsettings.json.
   b. Edit the “ForwardedHeaders” section so that it reads:

```
{
    ...
    "ForwardedHeaders": {
        "Enable": true,
        "ForwardedForHeaderName": "X-Forwarded-For",
        "KnownProxiesIP": [trusted IP addresses]
    }
    ...
}
```

Where:

“**Enable**” turns on the Forwarded headers processing function in the web application.

© 2023 Creatio. All rights reserved.

"**ForwardedForHeaderName**" is the name of the header that contains the IP address.

"**KnownProxiesIP**" is the trusted IP address list. Creatio will process the "**ForwardedHeader**" value only if it receives a request from these IP addresses. They may belong to the load balancer, reverse proxy, etc. If you leave this value empty, Creatio will process the "ForwardedHeader" value received from any IP address.

c. Repeat steps a-b for all Creatio application instances in your web farm.

Example

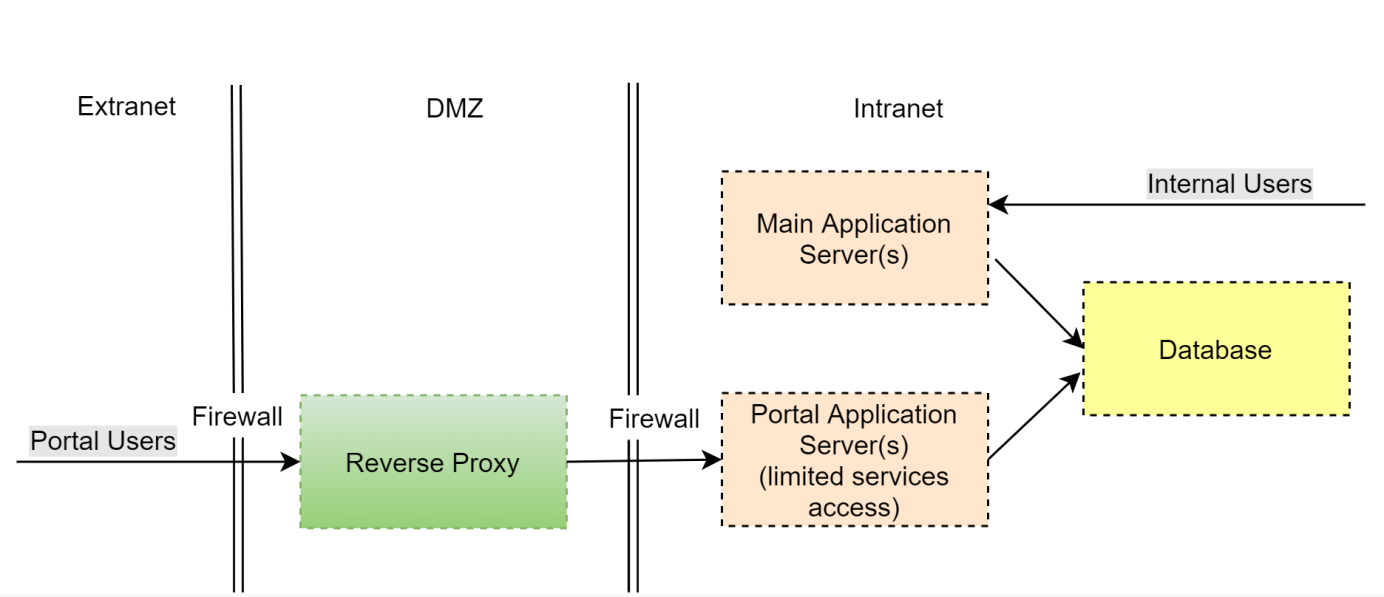
```
"KnownProxiesIP": ["127.0.0.1", "12.34.56.78", "2001:0db8:85a3:0000:0000:8a2e:0370:7334"]
```

Secure access to the portal

PRODUCTS: ALL CREATIO PRODUCTS

To ensure the safety of data, the on-site application must be deployed as a web farm when installing the portal. You can find a use case of setting up a web farm in the "Set up a web-farm for Creatio application server" article. Portal access is set up as follows (Fig. 1):

Fig. 1 Typical infrastructure with external network portal access



- **Demilitarized zone (DMZ)**
  - A reverse proxy server must be the only publicly accessible component in your DMZ.
  - The reverse proxy is used to implement the primary network activity logging. You can also configure it to limit access to the configuration files of your application.
  - Authorized portal users only have access to those configuration web services that they are expressly allowed to access at the application level.

© 2023 Creatio. All rights reserved.

  - During the development process, the access permissions for new web services are configured. Learn more in the “Restricting access to web services for portal users” article.

- **Internal network (Intranet)**
  - A separate set of application nodes is deployed on the web farm for servicing portal users. This set does not overlap with application nodes for servicing internal users.
  - To ensure the operation of the portal application and user application, separate accounts with different access permissions are created in the database.
  - The portal application settings deny the system users the ability to log in (“AuthProviders” are disabled, except for portal users). This is required to ensure that only portal users can create sessions from an external network (Extranet).
  - Additionally, you can configure external authentication providers to add a second authorization step.
  - Portal application nodes, DBMS, and user applications are deployed in separate segments with restricted access.

Version control system for development environments

PRODUCTS: ALL CREATIO PRODUCTS

Version control is required for deploying a development environment where several developers can commit, monitor, and merge the changes made to the Creatio configuration. The purpose of the version control system in Creatio is:

- transferring of changes between configurations
- storing multiple versions of configuration schemas
- rolling back changes to return to one of the previous versions.

Creatio supports integration with the Subversion control system (SVN) of version 1.7 and higher. For more details on using SVN see Subversion control system documentation.

Note. Creatio native development tools work only with the Subversion version control system. However, you can disable version control integration and use any version control system, including Git, when developing in the “File system development mode”. Learn more about working with Git in Creatio in our SDK guide.

An SVN repository should be the only point of contact for different development environments. Otherwise, the development environment of each developer must be insulated and run on an independent application server connected to a database not used by other Creatio application instances.

More information on setting up a development environment is available in the Development Guide.

The general procedure for setting up and connecting SVN is as follows:

- Deploy SVN and create a Creatio repository

© 2023 Creatio. All rights reserved.

- Connect the repository to Creatio

Deploy SVN and create a Creatio repository

To deploy Subversion for your Creatio application:

1. Install SVN server

You can install SVN on the application server, DBMS server or on a separate dedicated server.

To install the SVN server on a Windows operating system, use one of the publicly available SVN installers:

- VisualSVN
- CollabNet

Installation instructions for other operating systems, including Debian, are available with Apache Subversion.

The SVN server can function independently or use an Apache web-server as a frontend (both the VisualSVN and CollabNet utilities can install it as a component).

If the SVN server is running independently, repositories are accessed through the **SVN** protocol. If a web server is used as a frontend, repositories are accessed through the **HTTP** and **HTTPS** protocols.

We recommend installing a web-server frontend and using the webserver protocols (**HTTP** and **HTTPS**) for integration with Creatio.

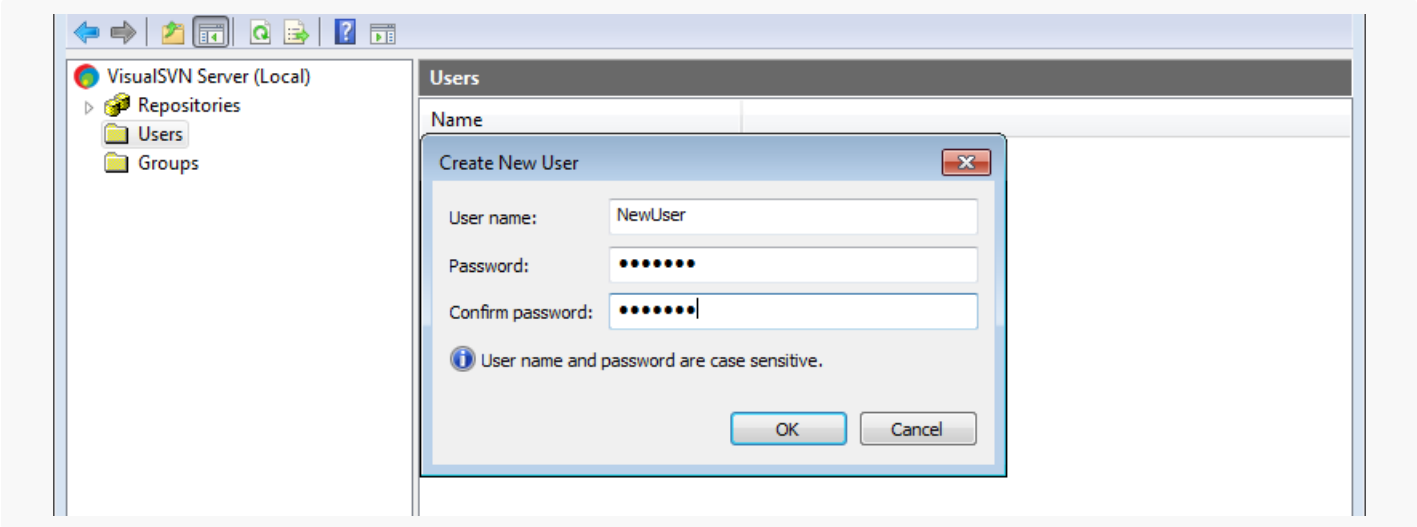
2. Create a user on the SVN server

To access the SVN server, add at least one SVN user. We recommend creating a separate user for each developer.

You can create an SVN server user with the standard tools supplied with the SVN server installation package, for example, VisualSVN (Fig. 1).

Working with the Creatio repository requires password-based authentication.

Fig. 1 Creating a new user in the SVN server (VisualSVN utility).



© 2023 Creatio. All rights reserved.

3. Create a repository on the SVN server

Create an SVN repository using the standard tools supplied with the SVN server installation package (i.e., VisualSVN and CollabNet).

Note. Creatio supports the simultaneous operation of several repositories that can be located on different SVN servers.

4. Install SVN client (optional)

You can optionally install an SVN client in the developer workplace, for example, TortoiseSVN.

Note. We recommend using TortoiseSVN client version 1.8 and up.

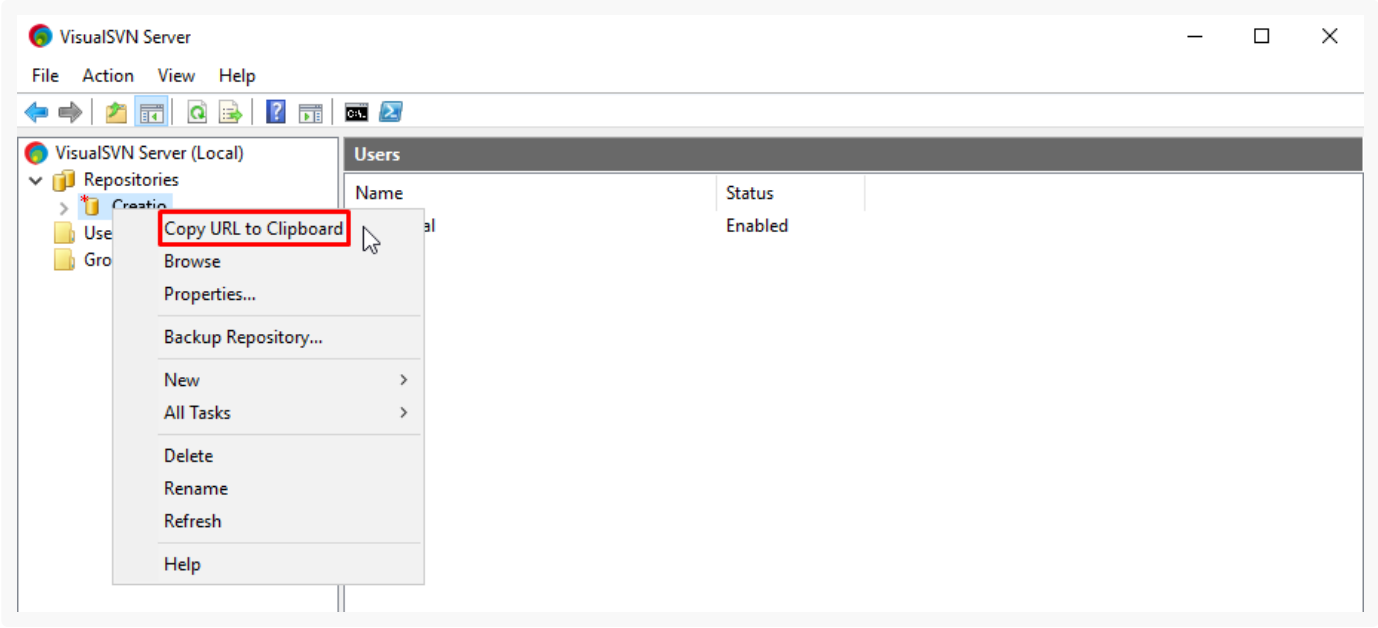
Installing an SVN client is not required since it does not affect the Creatio operation. Using an SVN client is convenient for viewing the local working copy, history, revert operations, review, etc.

Connect the repository to Creatio

To connect an SVN repository to Creatio:

1. Copy the URL of your repository. For example, in VisualSVN, right-click the repository → [*Copy URL to clipboard*] (Fig. 1).

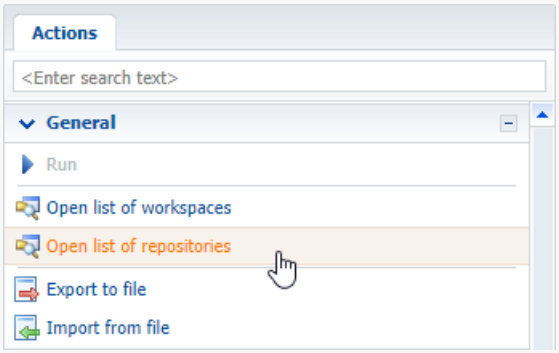
Fig. 1 Copy the URL of the repository



2. Click in the main Creatio application. The System Designer will open.
3. Click [*Advanced settings*] in the [*Admin area*] to open the [*Configuration*] section.
4. Click [*Open list of repositories*] on the [*Actions*] tab (Fig. 2).

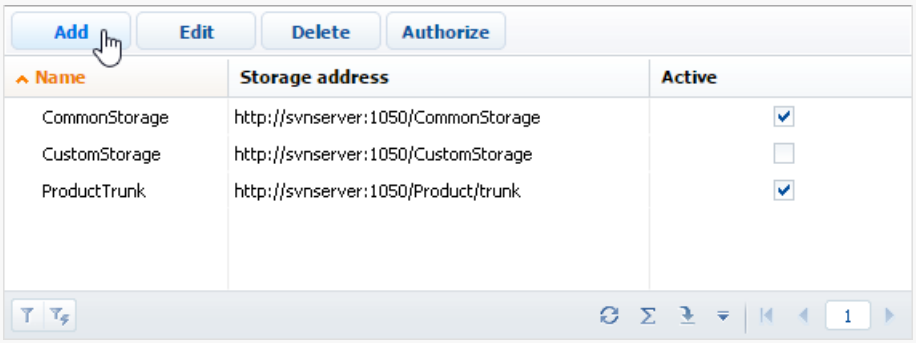
© 2023 Creatio. All rights reserved.

Fig. 2 Opening the SVN repository list



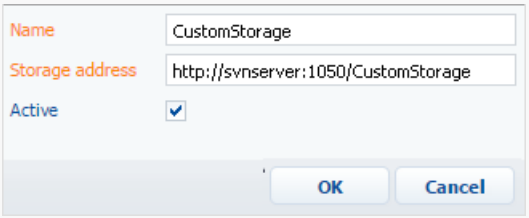
5. Click [*Add*] on the list toolbar (Fig. 3). A page for the new repository will open.

Fig. 3 Adding a new repository to the list of version control system repositories



6. In the new repository page, specify the repository data (Fig. 4).

Fig. 4 Entering the repository data in the repository page



[*Name*] – repository name.

[*Storage address*] – the network address of an existing SVN repository. Insert the URL that you copied on step 1 of this instruction.

The HTTP protocol (standard network protocol), HTTPS protocol (standard network protocol secured with SSL encryption), and SVN protocol (own network protocol of the Subversion system) are all supported in repository addressing.

[*Active*] – select this checkbox to enable using the repository in the system operation. Each new repository is marked as active by default.

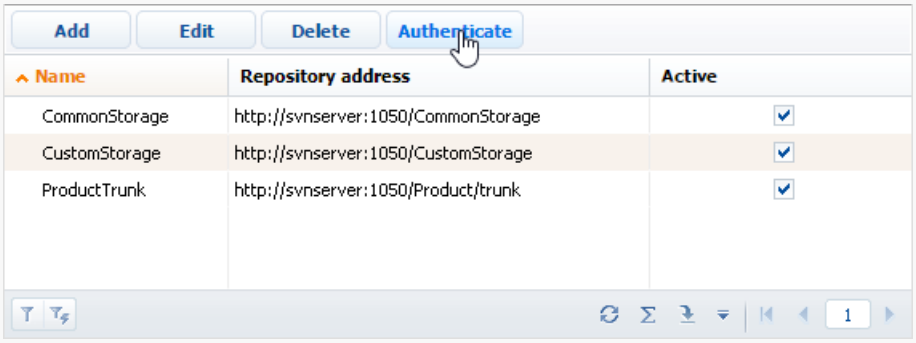
Note. You can work with active repositories only. Moreover, all repositories, from which the packages are updated, must be active. These include the repository from which the initial package is updated and

© 2023 Creatio. All rights reserved.

the repositories from which all packages-dependencies of the initial package are updated.

7. Click the repository in the repository list → [*Authenticate*] (Fig. 5).

Fig. 5 Authenticating a repository



8. Authenticate to your SVN repository using one of the users you have created on your SVN server (Fig. 6).

Fig. 6 Providing SVN credentials

As a result, your SVN repository will be connected to Creatio. Use the new repository to create custom packages and install the created packages in the workspace.

Learn more about working with packages using SVN, transferring changes using SVN, and working with SVN in general in our SDK guide.

Chat access

PRODUCTS: ALL CREATIO PRODUCTS

To manage Facebook Messenger and WhatsApp chat channels in Creatio on-site:

- Switch Creatio from HTTP to HTTPS. Learn more in a separate article: Switch a Creatio website from HTTP to HTTPS.
- Set up access to the chat service at https://sm-account.creatio.com/ on the application server.
- Set up an incoming connection to HTTPS protocol and protection by a valid certificate for the sm-account.creatio.com chat service on the application server.

To manage Telegram chat channels, make sure the application server has internet access.

If your Creatio application uses two-factor authentication, grant FacebookOmnichannelMessagingService, TelegramOmnichannelMessagingService, WhatsappOmnichannelMessagingService services access to incoming

© 2023 Creatio. All rights reserved.

requests.

Set up OAuth 2.0 authorization for integrated applications

PRODUCTS: ALL CREATIO PRODUCTS

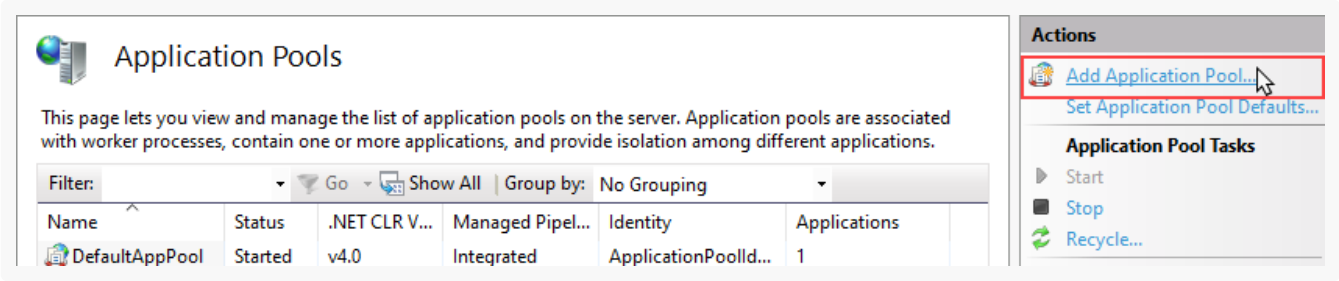
Use OAuth 2.0 protocol to securely authorize third-party applications and web services you integrate with Creatio. This technology does not pass Creatio logins and passwords to third-party applications. OAuth 2.0 also lets you restrict Creatio permissions for the integrated applications.

Install and set up the Identity Service

Deploy the database and Creatio application servers before installing and configuring the Identity Service. To install the Identity Service:

1. Access the Creatio application server.
2. Install the .NET Core runtime 2.2. Download the install file
3. Install the .NET Core Hosting Bundle. Download the install file
4. Restart the IIS.
5. Navigate to the Creatio install file folder, find the **IdentityService.zip** archive, and unzip it.
6. Add the Identity Service application **pool** to the IIS.
   a. Go to the [*Application Pools*] section in the [*Connections*] area of the IIS management window.
   b. Select [*Add Application Pool...*] in the [*Actions*] area.

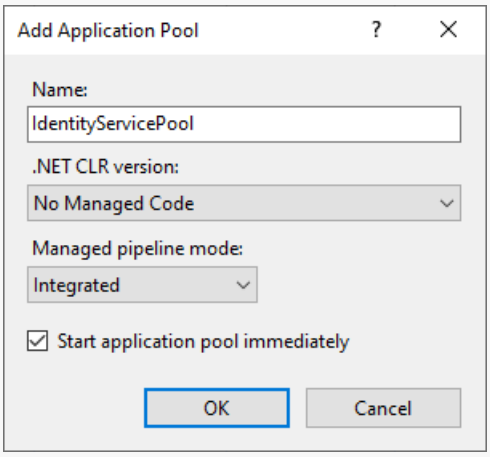
Fig. 1 Add the pool to the IIS



   c. Specify the pool name in the pool settings window, for example, “IdentityServicePool.” Set the [*.NET CLR Version*] field to “No Managed Code.”

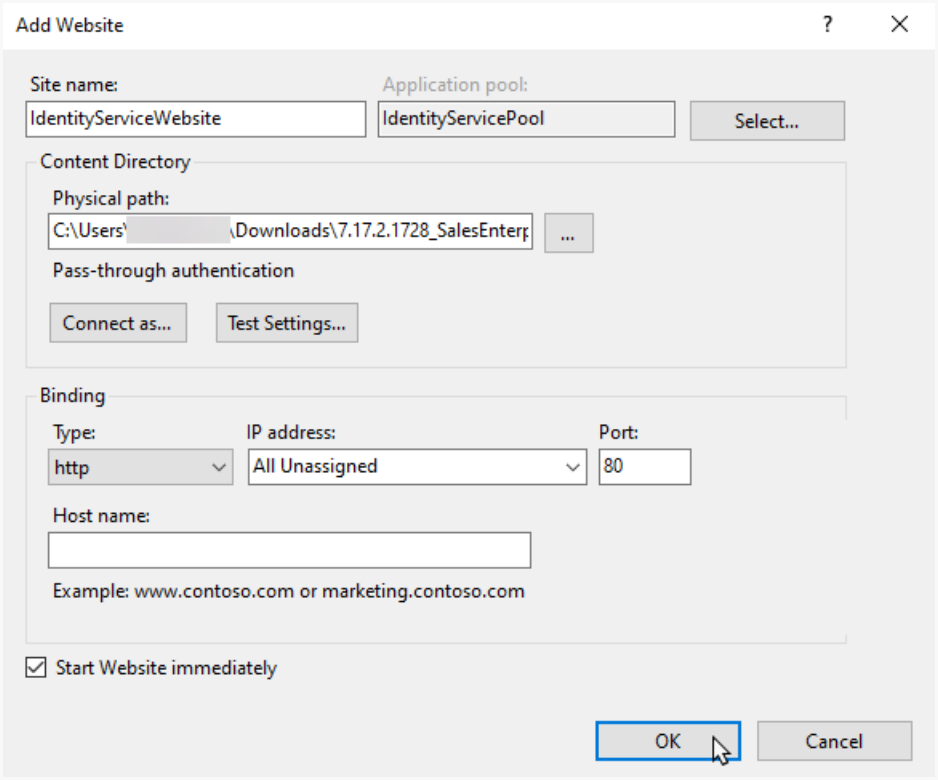
Fig. 2 Set up the Identity Service pool

© 2023 Creatio. All rights reserved.



7. Set up **access** to the application pool:
   a. Right-click the newly-created pool. Select [*Advanced Settings...*] in the context menu.
   b. Specify the user with Identity Service directory access permissions in the [*Identity*] field of the newly-opened window.
8. Create a new Identity Service **site** in the IIS.
   a. Click [*Sites*] on the IIS control panel and select [*Add Website*] from the context menu.
   b. Specify the site name, the pool and the path to the Identity Service root directory.

Fig. 3 Set up the site in the IIS



9. Connect the site to your Creatio **DBMS**. To do so, edit the **appSettings.json** configuration file in the Identity Service root directory:

© 2023 Creatio. All rights reserved.

a. Set the “DbProvider” parameter to “MsSqlServer” or “Postgres.”

b. Specify the connection string in the “MsSqlConnection” or “PostgresConnection” setting. We recommend using the same connection string you specified in Creatio. The user that connects to the database must have permissions to create and update the tables.

> **Note.** To connect the Identity Service to Creatio with Oracle DBMS, deploy an additional PostgreSQL or Microsoft SQL database instance.

10. Set up the Identity Service **system user**. To do so, specify the unique ClientId, ClientName, and ClientSecret values in the “Clients” block of the **appSettings.json** configuration file. The file is located in the Identity Service root catalog. Creatio and the Identity Service will use these values to interact with each other. All parameters support uppercase and lowercase letters, numbers, and special characters, for example, brackets or punctuation marks.

    Recommended parameters:

    ClientId — 16 characters.

    ClientSecret — 32 characters.

    ClientName — any number of characters.

    “Clients” block setup example:

    ```
    "[{\"ClientId\":\"{generate ClientId}\",\"ClientName\":\"{generate name}\",\"Secrets\":[\"{ge
    ```

    > **Note.** To avoid errors on Identity Service launch, specify the full path to openssl.pfx in the “**X509CertificatePath**” setting of the appsettings.json file. openssl.pfx is located in the root of the Identity Service directory.

11. Switch the Identity Service to **HTTPS**. The setup process is similar to switching Creatio to HTTPS. Read more: Switch Creatio website from HTTP to HTTPS.

12. Enable the Identity Service **logging**.

    a. Navigate to the Identity Service directory, open the web.config file, and set the “stdoutLogEnabled” parameter to “true.”

    b. Specify where you would like to store the Identity Service logs in the file's “stdoutLogFile” parameter.

    c. Open the appsettings.json file in the Identity Service root directory and configure the log level:

    ```
    "Logging": {
      "LogLevel": {
        "Default": "Error"
      }
    ```

© 2023 Creatio. All rights reserved.

```
}
```

Set up the Identity Service integration on Creatio's end

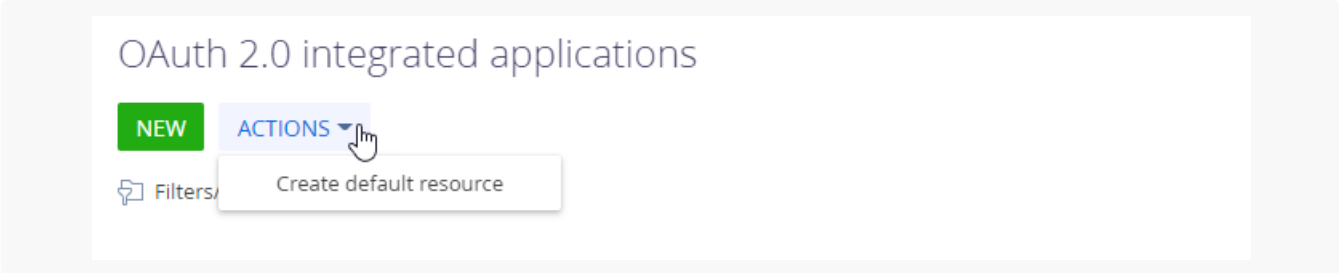
1. **Enable** the OAuth 2.0 integration in Creatio. To do so, execute this script in your Creatio database. Use it for both Microsoft SQL and PostgreSql.

```
UPDATE "AdminUnitFeatureState"

    SET "FeatureState" = 1

WHERE "FeatureId" = (

    SELECT

        "Id"

    FROM "Feature"

    WHERE "Code" = 'OAuth20Integration')
```

2. Fill in the system settings in the “OAuth 2.0” group:
   a. “**Authorization server Url for OAuth 2.0 integrations**” (“OAuth20IdentityServerUrl” code) — the IdentityServer URL, for instance, http://isEnpointExample.
   b. “**Client id for OAuth 2.0 integrations**” (“OAuth20IdentityServerClientId” code) — the Identity Service user ID you specified in the “ClientId” parameter of the appSettings.json file when setting up the IdentityServer.
   c. “**Client secret for OAuth 2.0 integrations**” (“OAuth20IdentityServerClientSecret” code) — the Identity Service user's secret key you specified in the “ClientSecret” parameter of the appSettings.json file when setting up the IdentityServer.
3. Create a default resource. You only need to perform this action once when setting up the Identity Service integration.
   a. Click to open the System Designer.
   b. Open the [*OAuth 2.0 integrated applications*] section.
   c. Select [*Create default resource*] in the [*Actions*] menu.

Fig. 4 Add a default resource

© 2023 Creatio. All rights reserved.



This will create a default resource record with your Identity Service account details.

Set up the OAuth 2.0 authorization

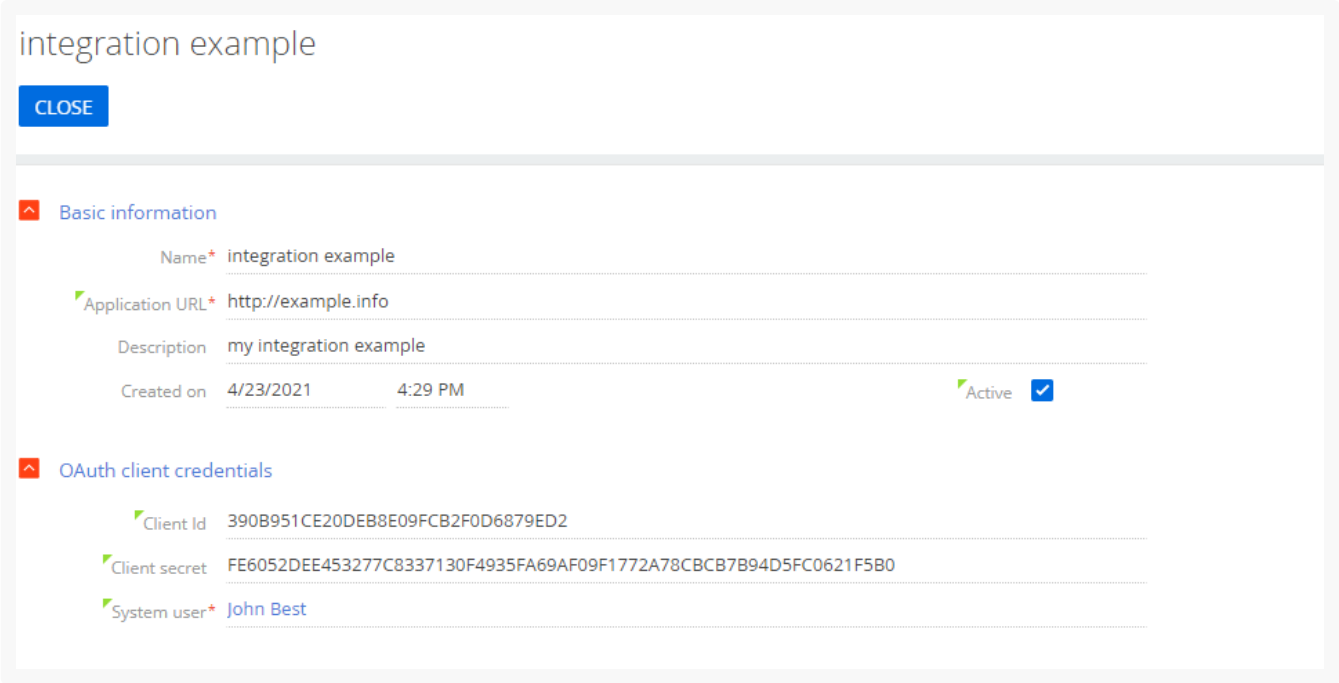
Once you install the Identity Service and connect it to Creatio, add a client record for each application you are going to authorize with OAuth 2.0. To do so:

1. Click to open the System Designer.
2. Open the [*OAuth 2.0 integrated applications*] section.
3. Click [*New*].
4. Fill in the client parameters for the relevant application on the newly-opened page.
   a. [*Name*] — the title that the integration list and the logs will use.
   b. [*Application URL*] — the URL of the integrated application or the web service.
   c. [*Description*] — the purpose of the integration.
   d. [*Active*] — enables and disables the integration.
   e. [*System user*] — the Creatio user with sufficient permissions for this integration. We recommend permitting this user to only read and edit the fields the integrated application or the web service need to change. For example, if you are integrating a web service that passes the currency exchange rates to Creatio, grant permissions to only read and edit the [*Rate*] and [*Start*] fields of the [*Currency*] lookup.

      Creatio automatically populates the **client account details** (the ID and the secret).

      Fig. 5 Set the client

© 2023 Creatio. All rights reserved.



5. Save the record.
6. Repeat steps 3-6 for all applications you need to authorize with OAuth 2.0.

© 2023 Creatio. All rights reserved.