ELO Integration Client

Installation and operation

Table of contents

Installation and operation

Configuration

You can configure what functions are made available in the ELO Integration Client. This configuration determines which client root entry to show and what functions are available. Functions that are not available are hidden.

Default configuration

The client starts with a default configuration that is customized during start-up.

The configuration contains the following values:

Name	Explanation	Type [default]
guid	The GUID of the root entry the client should show.	String [null]
noForm	Prevents a form from being displayed for an entry in the document view.	Boolean [true]
noFeed	Prevents the feed from being displayed for an entry in the document view.	Boolean [true]
noFulltext	Prevents the full text content from being displayed for an entry.	Boolean [true]
noImagePreview	Prevents the image preview from being displayed for an entry.	Boolean [true]
rFInit	Name of a registered function called for start-up.	String [null]
syncFlowInit	Calls a trigger for synchronous flows instead of the registered function (rFInit).	String [null]
newDocVersFct	Standard ELO function *Create new version* to upload new version of existing document entries.	Boolean [false]
saveDocFct	Standard ELO function *Save document* to download documents.	Boolean [false]
gotoFct	Go to function to open the entry in the ELO Java Client.	Boolean [false]
webClientURL	ELO Web Client URL to open the ELO Web Client when the Go to function is enabled.	String [null]

Name	Explanation	Type [default]
gotoURL	Uses any URL as the target for the goto function. The query parameter *guid* with the ELO GUID of the current entry is appended to the URL. Can be used to open an ELO Integration Client in a new window, for example. (This function is available with ELOwf versions 12.07, 21.07, and 20.00 and higher)	String [null]
useJcHAIGoto	Uses the ELO Java Client HTTP Automation Interface to execute the goto function. The ELO Web Client URL and the goto URL must not be set. The add-on module for ELO Integration Client support available on install.myelo.net is also required in the ELO Java Client. When the ELO Java Client starts, the port of the ELO HTTP Automation Interface is buffered and called for the ELO Integration Client goto. The goto only works if the add-on module is installed, the ELO Java Client is already running, and a user is logged on. In addition, the same ELO user has to be used in the ELO Integration Client and the ELO Java Client. In restricted environments, such as Microsoft Teams, abas ERP, or SAP Business One), this enables users to go from the ELO Integration Client to the ELO Java Client. (This function is available with ELOwf versions 20.14, 23.00.004, and 23.02.000 and higher)	Boolean [false]
removeEntryFct	Standard ELO function *Delete* to delete an entry.	Boolean [false]
editInOfficeFct	Standard ELO function *Edit in Office* to edit an Office document using the local Office applications.	Boolean [false]
removeEntryFct	Standard ELO function *Delete* to delete an entry.	Boolean [false]
startWorkflowFct	Standard ELO function *Start workflow* to start a workflow.	Boolean [false]
startWorkflowTemplates	List of workflow templates available for an active *startWorkflowFct*. The configuration is an array of objects with the ID and Name properties of the template.	Array [null]
externalLinkFct	Standard ELO function *Create external link* for creating a download link for external users.	Boolean [false]
folderMask	Folder form for enabling the *Create folder* function. Only folders with the configured metadata form are created.	String [null]
documentMask	Document form for enabling the *Insert file* function. Only files with this metadata form are filed.	String [null]

Name	Explanation	Type [default]
hasSearch	Enables the option to search the entire repository. The results are listed. Note: Only certain functions are available within the search. If the form, feed, or configured ELO apps are enabled, entries within the search can still be edited, provided the user has the corresponding rights.	Boolean [false]
referenceEntryFct	Enables entries from the search to be referenced.	Boolean [false]
refreshFct	Allows you to refresh the current entry.	Boolean [false]

> **Information**
>
> Functions that are not configured are hidden. This does not prohibit the function from being used. To prevent a function from being used, you have to withdraw the users' permissions to the entry.

Start-up

Opening the client requires certain parameters that help the client load its configuration. The client goes through a specific start-up procedure to enable the most flexible configuration possible.

URL parameters

The client responds to the following URL parameters (besides the standard parameters of an ELO app):

- *system=<system ID>*: This parameter specifies the system the client has been integrated into. Each client call requires this parameter. The value of this parameter is arbitrary, but an entry must be made for this value in the configuration.
- *GUID=<GUID of the root entry>*: [optional] This parameter indicates which entry should be used as the root entry. A GUID has to be entered; an object ID is not permitted.
- *key=<group name of a field in the metadata>&value=<value of the field>*: [optional] Instead of using a GUID from the parameter as a root entry, you can also perform a search for a field value (in the metadata) to find the root entry. An invoice with a specific invoice number could serve as a root entry as an example.

In addition to these parameters, you can pass other processing parameters in an RF in the URL. These are then used later on during start-up.

Default configuration

The default configuration applies when starting the client. If a GUID parameter has been entered, this GUID is applied to the configuration.

Repository configuration

The client loads its ELO app repository configuration (elo.integration.Client.json) from the repository (under *//Administration//ELO Apps//Configuration//*). It contains a configuration for all relevant system parameters. It is used depending on the value of the system parameter (*system*) and overrides the values from the default configuration (including any GUID parameter). If the repository configuration does not contain a value, the previous value is retained.

Registered function

If the value *rFInit* is set after reading the repository configuration, a registered function with this name is called. The registered function receives all URL parameters allowing additional parameters to be processed here, depending on the integration system. This call expects a configuration object that overrides the previous configuration value.

For example, a registered function could:

- Show/enable different functions depending on the user.
- Perform a repository search to determine the root entry as GUID.
-

Initiate a process that creates a new structure that will then be used as the root entry.

Flows trigger

You have the option to specify a flows trigger instead of a registered function. If you use a flows trigger, you cannot specify a registered function. ELO Flows must be installed and there must be a component with a corresponding trigger (synchronous trigger with result). You can store the path from the ELO Flows configuration here. It is possible to execute a trigger with multiple flows (automations) or a specific flow configured as a data query.

Examples

Automations:

```
"syncFlowInit": "api/v1/trigger/com.elo.bi.flows.component/IntegrationServiceSf/1.0/
IcCallSf"
```

Data queries:

```
"syncFlowInit": "api/v1/trigger/com.elo.bi.flows.component/IntegrationServiceSf/1.0/
IcCallSf?trigger=86696b32-a734-4aa9-9351-4d41d637b5b3"
```

Index search

If no GUID was set for the root entry during the previous start-up and if the URL parameters *key* and *value* exist, a search for the specified value (*value*) is performed in a field (from the metadata) with the group name (*key*) and the first result is used as the root entry.

Initialization

The client loads the root entry, builds the function menu, and shows the entry; the client is then operational for the user. This concludes the start-up process.

System-specific integration

You will find documentation on the system-specific installation of special modules with functions that go beyond the scope of the ELO Integration Client under:

- ELO for Microsoft
- ELO for SAP Solutions
- ELO for Salesforce
- ELO for CAS
- ELO for abas ERP

Known issues

Bootstrap Glyphicons

The integrated Internet Explorer in Navision 2015 was unable to load the font for the Bootstrap Glyphicons in some cases. The icons are not shown.

SameSite flag

If the ELO Integration Client is embedded in an external website (e.g. during the integration of Salesforce, SmartWe, or Microsoft Dynamics 365 Business Central), you will encounter the following problem:

With versions 80+ of Google Chrome and Microsoft Edge, the default handling of the SameSite cookie has changed from *none* to *lax*.

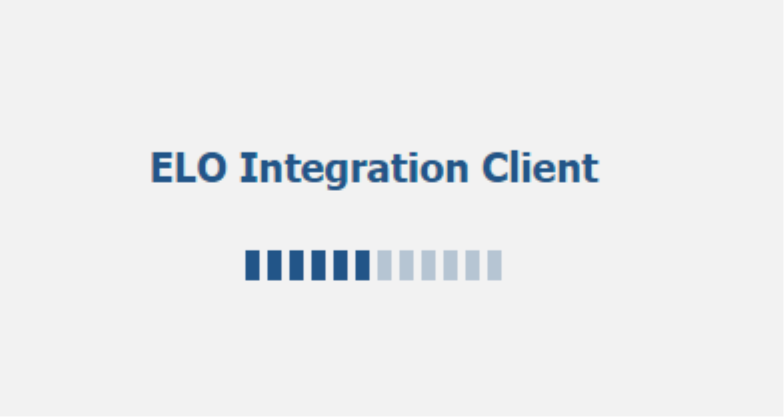
SameSite:

The SameSite flag indicates whether a cookie is attached to a request if it is integrated in an *iFrame* that is not on the same server.

The settings are as follows:

- *None*: Cookies are always attached (only works if the *secure flag* is also set)
- *Lax*: Cookies are only attached to GET requests or if the servers are the same
- *Strict*: Cookies are only attached if the servers are the same

Cookies are required for the ELOix connection (which executes post requests). We therefore encounter this problem if an ELOix connection is made in an *iFrame* that is not on the same ELO server. In this case, the ELO Integration Client will not load, or the loading indicator will stop:



This also affects ELOauth since cookies are also required here.

Information

> The ELO Salesforce integration and other integration solutions that have a browser-based third-party system meets these criteria.

Solution

Requirements:

ELO must be accessible using HTTPS, and you must either be able to operate the system with a certificate signed by a trusted authority or import a self-signed certificate in all clients. This also sets the *secure flag* on the cookies. This option is only available starting with Apache Tomcat 9.0.21 and 8.5.42.

Method:

1. Stop the ELO Application Server (Apache Tomcat).
2. In Apache Tomcat, configure the context of the web application (here: the ELO Indexserver) so that the SameSite property is set on the session cookie: The XML File to be configured is located in the following example path: *E:\ELOprofessional\servers\ELO-<Server>\conf\Catalina\localhost\ix-<Repository>.xml*

   Configuration file example:

   ```
   <?xml version="1.0" encoding="UTF-8" standalone="no"?>
   <Context docBase="E:\ELOprofessional\prog\webapps\ix.war" path="/ix-<Repository>" unpa
       <Environment name=webappconfigdir" override="false" type="java.lang.String"
       value="E:\ELOprofessional\config\ix-<Repository>\ELO-<Server>"/>
       <CookieProcessor sameSiteCookies="none" />
   </Context>
   ```

3. In the ELO installation directory, delete the deployment folder of the respective application (example: *E:\ELOprofessional\servers\ELO-<Server>\webapps\ix-<Repository>*)
4. Restart the ELO Application Server.