

# Customer-Dev

Customer Documentation with integration information and technical details

- [Swiss Learning Hub integration](#)
- [Detail report](#)
- [User Import](#)
- [Supervisor Import](#)
- [SSO / SAML 2.0](#)
  - [Integrations](#)
  - [Microsoft Azure - Exchange of Metadata: Basic Guideline](#)

# Swiss Learning Hub integration

The purpose of this documentation is to describe available integrations of the Swiss Learning Hub.

Integration	Description
<u>Detail report</u>	In the Swiss Learning Hub, a detailed report can be generated per learning content if the report definition and certain SCORM attributes are provided by the learning object.
<u>File based API</u>	File based user import in XML or JSON format over SFTP.
<u>LTI</u>	Learning Tools Interoperability (LTI) is an education technology specification developed by the IMS Global Learning Consortium, which specifies a method for a LMS to securely exchange information with external systems.
<u>SAML 2.0</u>	Single sign-on (SSO) with SAML

# Detail report

In Swiss Learning Hub, a detailed report can be created per learning content if the report definition (*report.xml*) and certain SCORM keys (**cmi.objectives.n.x** and **cmi.interactions.n.x** are provided by the learning object.

## Report definition

### File name and location

The XML file must always have the file name **report.xml** and must be stored in the *root* of the respective SCORM package. The detail report can only be created if the Swiss Learning Hub can find and use this file.

## Scheme

The XSD scheme can be used to create the report file.

### Element **report**

Is the root element of the report XML

### Element **organization**

This element refers to a chapter in the learning object.

Attribute:

- id = unique ID of the chapter (e.g.: chapter\_000\_010\_150)
- name = Name of the chapter (e.g.: Schweizer Quiz)

No hierarchical chapter structures can be mapped. The chapters must be reduced to one level.

*Example:*

- Chapter 1
  - Chapter 1-1

- Chapter 1-2
- Chapter 2
  - Chapter 2-1

*report.xml*

```
<organization name="Chapter 1">
  <item name="Page A" />
</organization>
<organization name="Chapter 1-1">
  <item name="Page B" />
</organization>
<organization name="Chapter 1-2">
  <item name="Page C" />
</organization>
<organization name="Chapter 2">
  <item name="Page D" />
</organization>
<organization name="Chapter 2-1">
  <item name="Page E" />
</organization>
```

## Element **item**

This element refers to a learning object page within a chapter.

Attribute:

- id = unique ID of the learning object page (e.g.: item\_000\_010\_010\_de)
- name = Page title (z.B.: Testbeginn / Sprachen / Hauptstadt)
- type = Type of learning object page (e.g.: dynamicFrames / multipleChoice / testInteraction, dynamicFrames)

## Element **question**

This element refers to a question within a learning object (*item*).

Attribute:

- name = Question text (e.g.: Zu welchem Kontinent gehört die Schweiz? Wählen Sie die korrekte Antwort.)
- type = Question type (e.g.: multipleChoice / matchingDragAndDrop / matching / matrix)
- maxScore = Maximum number of points for this question (for example: 1)

## Element **answer**

This element refers to an answer option of a question.

Attribute:

- id = unique ID of the answer choice (e.g.: A / 1.A / usw.)
- name = Description of the answer choice (e.g.: Afrika / Tennisspieler / Schweiz / Nicht Schweiz / usw.)
- correct = Indicates whether the answer choice is correct or not (e.g.: true / false)

## SCORM keys

The learning object must send data to the following SCORM keys according to SCORM 1.2 (Specification SCORM 1.2 RunTimeEnv).

- cmi.objectives.n.id
- cmi.objectives.n.score.raw
- cmi.objectives.n.score.min
- cmi.objectives.n.score.max
- cmi.objectives.n.status
- cmi.interactions.n.id
- cmi.interactions.n.objectives.n.id
- cmi.interactions.n.objectives.n.type
- cmi.interactions.n.objectives.n.time
- cmi.interactions.n.correct\_responses.n.pattern
- cmi.interactions.n.weighting
- cmi.interactions.n.student\_response
- cmi.interactions.n.result
- cmi.interactions.n.latency

## Mapping SCORM keys and the report XML

The SCORM key **cmi.interactions.n.id** (value e.g.: item\_000\_010\_020\_en) can be used to determine which question is involved in the report XML. All further SCORM keys for this interaction can be read out using the index (n). The same logic is used for the SCORM keys **cmi.objectives.n.x**. The index (n) for cmi.objectives.n.x and cmi.interactions.n.x does not necessarily have to have the same value per learning object page, nor is the order of the indexes

necessarily the same as the page order of the learning object.

*Example (report.xml)*

```
<item id="item_000_010_020_de" name="Geographie" type="multipleChoice">
  <question name="Zu welchem Kontinent gehört die Schweiz? Wählen Sie die korrekte Antwort."
type="multipleChoice" maxScore="1">
    <answer id="A" name="Afrika" correct="false"/>
    <answer id="B" name="Amerika" correct="false"/>
    <answer id="C" name="Asien" correct="false"/>
    <answer id="D" name="Europa" correct="true"/>
  </question>
</item>
```

The SCORM key **cmi.interactions.n.student\_response** (value e.g.: D) can be used to read out which response option the learner has selected.

*Example (report.xml)*

```
<question name="Zu welchem Kontinent gehört die Schweiz? Wählen Sie die korrekte Antwort."
type="multipleChoice" maxScore="1">
  <answer id="A" name="Afrika" correct="false"/>
  <answer id="B" name="Amerika" correct="false"/>
  <answer id="C" name="Asien" correct="false"/>
  <answer id="D" name="Europa" correct="true"/>
</question>
```

The SCORM key **cmi.objectives.n.score.raw** (value e.g.: 1) can be used to determine how many points the learner has achieved in this question. The SCORM key **cmi.interactions.n.result** (value: correct / incorrect) shows whether the learner has solved the question completely correctly.

# Example

## XML

File name: report.xml XML scheme: <https://cdn.swisslearninghub.com/xml/trc/v1.0/report.xsd>

```
<?xml version="1.0" encoding="utf-8"?>
<report xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xmlns="http://cdn.crealogix.com/xml/trc/v1.0/report"
xsi:schemaLocation="http://cdn.crealogix.com/xml/trc/v1.0/report
http://cdn.crealogix.com/xml/trc/v1.0/report.xsd">
<organization id="chapter_1" name="Kapitel 1">
  <item id="page_A" name="Seite A" type="dynamicFrames" />
</organization>
<organization id="chapter_1_1" name="Kapitel 1-1">
  <item id="question_1" name="Geographie" type="multipleChoice">
    <question name="Zu welchem Kontinent gehört die Schweiz? Wählen Sie die korrekte Antwort."
type="singleChoice" maxScore="1">
      <answer id="A" name="Afrika" correct="false"/>
      <answer id="B" name="Amerika" correct="false"/>
      <answer id="C" name="Asien" correct="false"/>
      <answer id="D" name="Europa" correct="true"/>
    </question>
  </item>
</organization>
<organization id="chapter_1_2" name="Kapitel 1-2">
  <item id="question_2" name="Geographie" type="multipleChoice">
    <question name="Welcher Kontinent liegt ganz auf der Südhalbkugel? Wählen Sie die korrekten
Antworten." type="multipleChoice" maxScore="1">
      <answer id="A" name="Afrika" correct="false"/>
      <answer id="B" name="Australien" correct="true"/>
      <answer id="C" name="Asien" correct="false"/>
      <answer id="D" name="Europa" correct="false"/>
      <answer id="E" name="Antarktika" correct="true"/>
    </question>
  </item>
</organization>
<organization id="chapter_2" name="Kapitel 2">
  <item id="page_B" name="Seite B" type="dynamicFrames" />
</organization>
<organization id="chapter_2_1" name="Kapitel 2-1">
  <item id="question_2" name="Geographie" type="multipleChoice">
    <question name="Welcher Kontinent liegt ganz auf der Nordhalbkugel, ganz auf der Südhalbkugel oder
auf beiden? Wählen Sie die korrekten Antworten." type="matrix" maxScore="1">
      <answer id="1.A" name="Afrika / Nordhalbkugel" correct="false"/>
      <answer id="1.B" name="Afrika / Südhalbkugel" correct="false"/>
      <answer id="1.C" name="Afrika / auf beiden" correct="true"/>
      <answer id="2.A" name="Europa / Nordhalbkugel" correct="true"/>
```

<answer id="2.B" name="Europa / Südhalbkugel" correct="false"/>

<answer id="2.C" name="Europa / auf beiden" correct="false"/>

</question>

</item>

</organization>

</report>



# User Import

This document describes the file-based interface for user import into the Swiss Learning Hub. The personal data can be transferred in XML format via SFTP.

## Data transfer

The files are transferred to the SLH server via SFTP. You will receive the SFTP access data from your SLH contact person.

## File format

The file format is XML. The files must be encoded as UTF-8.

## Process time

The process time can be determined individually (usually once a day at night), but must be coordinated with other processes by SLH.

## User import

The full functionality of a user import can only be executed if all current users are listed in the transferred file. It is possible to configure the user import for a specific organizational unit only. Thus, further organizational units can be maintained manually in the Swiss Learning Hub and are not affected by further processes.

# Processes

## Removing users

The user import is designed to automatically remove users that are not listed in the file. The import can be configured accordingly, so that these users are effectively deleted (incl. all learning data) or archived (thus access by the user to the system is prevented, but as an administrator one still has access the user incl. all his/her learning data). If a user has set the value "Deletable" = no (0) in the Swiss Learning Hub, this user will never be removed by the user import. Thus, e.g. via the interface of the Swiss Learning Hub, external users can be subscribed, which do not have to be part of the user import file.

It's also possible to protect user from deletion by adding them to a specific organizational unit. In the setup a queryparameter ('excludeorgs') with a list of units which should be excluded from deleting/archiving.

## Updating users

If a user is already integrated in the Swiss Learning Hub, it will not be newly created by the user import, but updated. However, the user must be uniquely identifiable for this purpose. In the first priority, the user is identified by the personnel number. A personal id can be, for example, a

system ID from the conversion system or effectively an internal company value. It should be noted that this is unique and will certainly never be changed (e.g., in the event of a name change due to marriage) or reassigned to another person (e.g., e-mail addresses are reassigned after a resignation). This is important in case persons are not deleted from the Swiss Learning Hub for data protection reasons, but archived and certainly thus never overwritten by a new user. In second priority, the user is also identified by user name or e-mail address.

### Updating user roles

By default the import does not update a users role. It's set on the first import (on create) of a user. After that, higher roles than “learner” have to be set manually in user administration. There is a functionality (query parameter 'changerole') in place which explicitly allows to update all user roles given in an import file.

# Data model

## Object Person

Name	Description	Key	Format	Example
Prename	first name of the person.	prename	String 255 character	Max
Name	last name of the person.	name	String 255 character	Muster
E-Mail	E-mail address of the person.	email	String 255 character	max.muster@meinefirma.ch
Username	Unique username of the person.	username	String 255 character	max.muster@meinefirma.ch
Personnel number	Unique internal id of the person.	personal_id	String 255 character	135487

Name	Description	Key	Format	Example
Status	<p>Status of the person:</p> <ul style="list-style-type: none"> <li>• activated = person can use the Swiss Learning Hub</li> <li>• deactivate d = person cannot use the Swiss Learning Hub, but the administr ator can use the person, just like an activated person</li> <li>• archived = person cannot use the Swiss Learning Hub, the admin can access the person incl. learning data.</li> </ul>	status	<p>Enum:</p> <ul style="list-style-type: none"> <li>• enabled</li> <li>• disabled</li> <li>• archived</li> </ul>	enabled
Birthday	Birthday of the person.	birthday	Date yyyy-mm-dd	1986-04-12
Deletable	Defines whether a person can be deleted by the user import (manually or automatically).	is_deletable	Boolean (0/1)	1

Name	Description	Key	Format	Example
Language	The language of the person.	language	Language (CLDR) <ul style="list-style-type: none"> <li>• de = german</li> <li>• fr = french</li> <li>• it = italian</li> <li>• en = english</li> </ul>	de
Role	The system role of the person.	role	Enum: <ul style="list-style-type: none"> <li>• learner</li> <li>• default-subadministrator</li> <li>• administrator</li> </ul>	learner
Organizational units	All organizational units of the person.	orgunits	Array of the object <i>organizational unit</i>	
Job descriptions	All job descriptions of the person.	jobdescriptions	Array of the object <i>job description</i>	

## Object Organisational unit

Name	Description	Format	Example
Name	Complete path of the organisational unit	String. Organisational units are separated by "/", per unit max 255 chars can be used	OU-1/OU-11/OU-111

## Object Jobdescription

Name	Description	Format	Example
Name	Complete path of the jobdescription	String. Jobdescriptions are separated by "/", per unit max 255 chars can be used	Developer/Frontend

## Example

```
<?xml version="1.0" encoding="UTF-8"?>
<persons schemaVersion="1.0"
```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="https://cdn.swisslearninghub.com/xml/trc/v2.0/import_person"
xsi:schemaLocation="https://cdn.swisslearninghub.com/xml/trc/v2.0/import_person
https://cdn.swisslearninghub.com/xml/trc/v2.0/import_person.xsd">
  <person>
    <firstname>Max</firstname>
    <name>Muster</name>
    <email>max.muster@meinefirma.ch</email>
    <username>max.muster</username>
    <personal_id>135487</personal_id>
    <status>enabled</status>
    <birthday>1986-04-12</birthday>
    <is_deletable>1</is_deletable>
    <language>de</language>
    <role>learner</role>
    <orgunits>
      <orgunit>OU-1/OU-11</orgunit>
      <orgunit>Entwicklung/Team Frontend</orgunit>
    </orgunits>
    <jobdescriptions>
      <jobdescription>Meine Tätigkeit</jobdescription>
      <jobdescription>Frontend Entwickler</jobdescription>
    </jobdescriptions>
  </person>
  <!-- more persons -->
</persons>

```

Validate XML file using `xmllint` and schema file:

```

# Download schema file
wget -O import_person.xsd https://cdn.swisslearninghub.com/xml/trc/v2.0/import_person.xsd

# Validate/Lint XML file
xmllint --schema import_person.xsd import_persons.xml --noout

```

# Supervisor Import

Goal is to import/synchronize bigger sets of supervisor-to-users relations. `Content-Type` header defines the requested import format:

Supported content types:

- `application/json`
- `application/xml`
- `text/csv`

## Configuration Possibilities

Different importing-modes can be applied which define what will happen to entries that are available in the database, but are not available via the given import file anymore (deprecation).

- **None**

In mode none - nothing will happen to the existing entries in our database if it's missing in the current file

- **Delete**

Before the import we will remove all existing supervisor-user relations and recreate this from the provided file. This is the default setting

The other way round, if there are entries in the provided file that do not exist in our database, the entries are simply ignored, the import is not stopped and runs normally.

## Supported content-types / hierarchy

JSON, XML and CSV are supported filetypes. They all contain a simple list of **supervisor<->user** relation - the matching happens based on **USERNAME**.

A supervisor can have multiple users whereas a user only has one supervisor.

### application/json

#### Example

```
[
  {
    "supervisor": "chefa",
    "user": "usera"
  },
```

```
{
  "supervisor": "chefb",
  "user": ""
}
]
```

## application/xml

### Example

```
<?xml version="1.0" encoding="utf-8"?>
<supervisors>
  <supervisor>
    <supervisor>chefa</supervisor>
    <user>usera</user>
  </supervisor>
  <supervisor>
    <supervisor>chefb</supervisor>
    <user/>
  </supervisor>
</supervisors>
```

## text/csv

### Example

```
supervisor,user
chefa,usera
chefb,
```

## Fields

While the field `supervisor` is mandatory, the field `user` can be omitted (see examples above).

Field	Mandatory	Info
<code>supervisor</code>	yes	Marks user as supervisor and gives permissions
<code>user</code>	no	Attaches user as member of supervisor

## Process time

The process time can be determined individually (usually once a day at night), but must be coordinated with other processes by SLH.



# SSO / SAML 2.0

# Integrations

Security Assertion Markup Language (SAML) is an open standard that allows identity providers (IdP) to pass authorization credentials to service providers (SP).

SAML transactions use Extensible Markup Language (XML) for standardized communications between the identity provider and service providers. SAML is the link between the authentication of a user's identity and the authorization to use a service.

Integration	Description
<a href="#"><u>Microsoft Azure</u></a>	SAML 2.0 integration over Azure Active Directory admin portal

# Microsoft Azure - Exchange of Metadata: Basic Guideline

## Preparation

1. Go to Azure Active Directory admin portal
2. Go to *Enterprise Applications*
3. create a *new application (create your own application)*
  - Name: "Swiss Learning Hub" or something that helps you recognize this configuration again
  - Select: **Integrate any other application you don't find in the gallery (Non-gallery)**
4. Go to *Single sign-on* and choose SAML
  1. On step 2 (Attributes & Claims): change the *Attributes & Claims* if necessary. As *Unique User Identifier (Name ID)* the same variable must be used, which contains the value that is identical to the Swiss Learning Hub username.
  2. On step 3 (SAML Certificates): Copy the *App Federation Metadata URL*. **Send this link (or Metadata XML behind this link) to your Swiss Learning Hub contact.**
5. Go to *Users and groups* and add users or groups to be authorized for authentication.

## Receive the metadata of Swiss Learning Hub

1. Go to Azure Active Directory admin portal
2. Go to *Enterprise Applications*
3. Choose the Swiss Learning Hub application
4. Go to *Single sign-on* and choose SAML
5. Upload the received metadata file.