Plugin *hub*

Dokumentation

¨

|  |  |
| --- | --- |
| Version vom: | 8. August 2014 |
| Version: | 1.2 |
| Plugin-version | 1.1.13 und höher |
| Verfasser: | Fabian Schmid, studer + raimann ag |

|  |  |  |
| --- | --- | --- |
| Version | Autor | Änderungen |
| 1.0 | Fabian Schmid |  |
| 1.1 | Fabian Schmid | Neue Einstellungsmöglichkeiten |
| 1.2 | Fabian Schmid | Dokumentation zum ersten Release auf github.com:  https://github.com/studer-raimann/Hub |

Inhaltsverzeichnis

1 Einleitung 4

2 Standort Middleware-Tabellen 6

2.1 ILIAS 6

2.2 MySQL-Server ausserhalb 6

3 Anbindungsmöglichkeiten 7

3.1 Variante 1: Füllen der hub-Tabellen durch Drittsystem 7

3.1.1 Erstellen einer Schnittstelle 7

3.1.2 Konventionen zum Füllen einer hubTabelle 8

3.2 Variante 2: Schnittstellen zu Drittsystem 9

3.2.1 Erstellen einer Schnittstelle 9

3.2.2 hubOrigin-Plugin erstellen 9

4 Synchronisationsablauf 11

4.1 Verbindungsaufbau und hubObjekte erstellen 11

4.2 hubObjekt-Status 12

4.2.1 Allgemein 12

4.2.2 Ermitteln des Status 12

4.2.3 Statusmodell 13

5 Konfiguration 14

5.1 Grundlagen-Konfiguration 14

5.2 Synchronisation je hubObjekt-Typ 14

5.2.1 hubCategory 14

5.2.2 hubCourse 15

5.2.3 hubUser 16

5.2.4 hubMembership 17

6 Starten der Synchronisation 18

6.1 Manueller Start 18

6.2 Cron-Job 18

7 Vorlage Synchronisationsregeln 19

7.1 Benutzerkonten (hubUsers) 19

7.2 Kategorien (hubCategories) 19

7.3 Workspaces (hubCourses) 20

7.4 Kursmitgliedschaft (hubMemberships) 20

# Einleitung

Das ILIAS-Plugin *hub* dient der Synchronisation von Drittsystemen zu ILIAS-Objekten und beinhaltet eine Middleware. Das Plugin vereinfacht die Entwicklungsarbeiten zum Anbinden eines Drittsystems an ILIAS dahingehend, dass es die Vollsynchronisation übernimmt und diese nicht für jede einzelne Anbindung implementiert werden muss. Es müssen lediglich die Middelware-Tabellen durch die Drittsysteme gefüllt werden, die Synchronisation findet zwischen diesen Tabellen und ILIAS statt.

Macintosh HD:Users:fschmid:Desktop:hubDokumentation:bsp.pdf

Ziel einer Anbindung ist es, die Daten eines Drittsystem in die Form der jeweiligen hub-Middleware-Tabelle zu bringen. Alle Einträge in diesen Middleware-Tabellen, werden durch die Vollsynchronisation des Plugins mit ILIAS-Objekten synchronisiert.

*hub* enthält aktuell vier hubObjekte mit den dazugehörenden Middleware-Tabellen, welche durch die Fremdsysteme gefüllt werden:

* hubUser
* hubCourse
* hubCategory
* hubMembership

Des Weiteren enthält *hub* ein Hilfsobjekt *syncHistory*, welches zu jedem der hubObjekte (Einträge in den Middleware-Tabellen) den Synchronisationssstand speichert.

Jedes Drittsystem dient als Datenquelle für die hubObjekte-Middleware-Tabellen. Ein Drittsystem kann prinzipiell mehrere hubObjekte füllen, es können aber auch mehrere Drittsysteme ein hubObjekt füllen. Beispielsweise können mehrere LDAP-Server die Tabelle hubUser füllen:

Macintosh HD:Users:fschmid:Desktop:hub.pdf

# Standort Middleware-Tabellen

## ILIAS

Standardmässig werden die Middleware-Tabellen für die hubObjekte zusammen mit dem Plugin in ILIAS installiert. Zugegriffen wird direkt über die MySQL-Zugangsdaten, welche ILIAS bereits kennt.

## MySQL-Server ausserhalb

Die Middleware-Tabellen für die hubObjekte können auch auf einem von ILIAs unabhängigen MySQL-Server installiert und verknüpft werden. Damit kann mit unterschiedlichen Benutzerdaten gearbeiten werden, wenn bspw. ein Drittsystem direkt auf die Tabelle zugreifen soll.

Achtung: Aktuell hat das Plugin hub für diese Funktionalität noch keine Benutzeroberfläche. Falls Sie Interesse an dieser Funktionalität haben sollten, melden Sie sich bitte bei fs@studer-raimann.ch

# Anbindungsmöglichkeiten

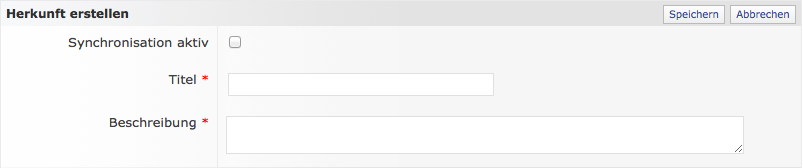
## Variante 1: Füllen der hub-Tabellen durch Drittsystem

Die Middlewaretabellen der hubObjekte können direkt mittels MySQL durch das Drittsystem gefüllt werden.

### Erstellen einer Schnittstelle

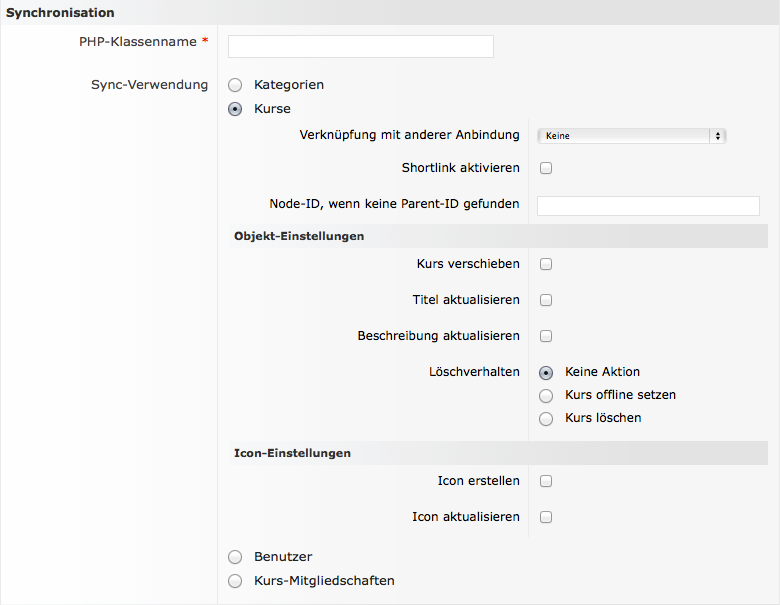
Im *hub*-Administrationbereich kann über den Link „[Anbindung hinzufügen](studer-raimann.ch)“ das Formular für eine neue Schnittstelle geöffnet werden:

#### Generelle Einstellungen



Wählen Sie einen Titel und eine Beschreibung für die Anbindung. Entscheiden Sie, ob die Synchronisation aktiviert sein soll oder nicht.

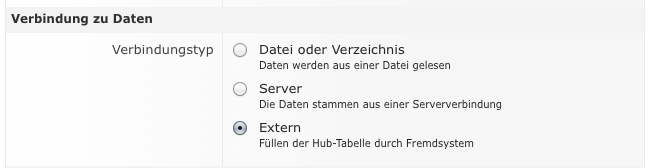
#### Synchronisationseinstellungen



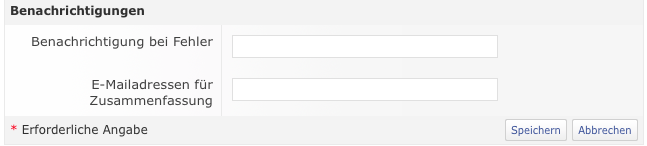
* Entscheiden Sie mittels „Sync-Verwendung“, für welches hubObjekt die Anbindung verwendet wird.
* Wählen Sie einen PHP-Klassennamen (z.B. myCourseSync).

Alle weiteren Einstellungen werden unter 5.2 erläutert.

#### Verbindungseinstellungen



#### Benachrichtigung



Setzen Sie eine E-Mailadresse, welche bei Fehlern benachrichtigt wird. Mit „Speichern“ wird die Anbindung erstellt.

### Konventionen zum Füllen einer hubTabelle

Eine Middleware-Tabelle enthält mind. die folgenden Felder, die auch zwingend gefüllt werden müssen:

|  |  |  |
| --- | --- | --- |
| Feldname | Wertebereich | Beschreibung |
| ext\_id | varchar 512 | Einmalige ID des Eintrages auf dem Drittsystem, z.B. numerische ID, ShibbolethUniqueId, ...  Das Feld ist PrimaryField und wird für den Objektaufruf verwendet. Sämtliche Änderungen am Eintrag muss über dieses Feld laufen. |
| delivery\_date\_micro | float 8 | Zeitpunkt der Lieferung[[1]](#footnote-1) eines Eintrages in Mikrosekunden. Kann das Drittsystem keine Mikrosekunden berechnen, kann auch die aktuelle Unix-Time eingesetzt werden. |
| sr\_hub\_origin\_id | int 8 | ID der Anbindung, siehe dazu 3.1.1 |

Operationen müssen immer als „UPDATE INTO ...“ ausgeführt werden (siehe dazu auch 4), damit neue Einträge erstellt, bereits vorhandene aber aktualisiert werden.

Bsp. einer Änderung eines Eintrages

UPDATE INTO `sr\_hub\_membership` (`container\_role`, `usr\_id`, `container\_id`, `ext\_id`, `delivery\_date\_micro`, `sr\_hub\_origin\_id`)

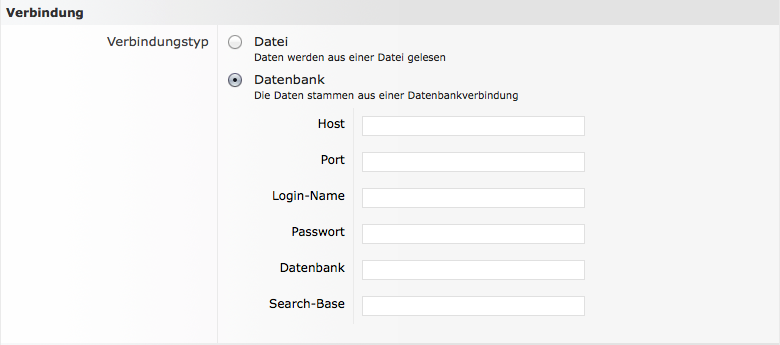
VALUES (2, 256, 44551456, 'fs@studer-raimann.ch\_468454', 1386237227.6189, 4) WHERE `ext\_id` = 'fs@studer-raimann.ch\_468454'

## Variante 2: Schnittstellen zu Drittsystem

### Erstellen einer Schnittstelle

Analog zu 3.1.1: Erstellen einer Schnittstelle.

#### Verbindungseinstellungen



Wählen Sie zusätzlich Verbindungseinstellungen, z.B. zu einer Textdatei oder zu einem Datenbank-, LDAP-Server. Das Passwort wird gesaltet in der Datenbank gespeichert.

### hubOrigin-Plugin erstellen

Durch Absenden des Formular für eine neue Anbindung wird im Verzeichnis ‚origins’ uns Unterverzechnis des jeweiligen hubObjektes eine PHP-Datei mit Ihrem gesetzten PHP-Klassennamen erstellt (bspw. class.myCoursSync.php). Diese Klasse beinhaltet bereits alle nötigen Methoden für die Anbindung:

|  |  |  |
| --- | --- | --- |
| Methode | Aufgabe | Rückgabewert |
| connect() | Verbindung zu Drittsystem aufnehmen | BOOLEAN |
| parseData() | Daten auf Drittsystem lesen | BOOLEAN |
| getChecksum() | Gibt Anzahl Datensätze zurück | INT |
| getData() | Gibt Array mit allen Datensätzen zurück | ARRAY |
| buildEntries() | hubObjekt-Einträge werden erstellt und gespeichert | BOOLEAN |

Über $this->conf können die Konfigurationswerte für die Verbindung abgerufen werden:

|  |  |
| --- | --- |
| Methode | Rückgabewert |
| getSrvDatabase() | Datenbank-Name |
| getSrvHost() | Datenbank-Host |
| getSrvPassword() | Entschlüsseltes Server-Passwort |
| getSrvPort() | Server-Port |
| getSrvUsername() | Server-Benutzername |
| getSrvInstance() | Server-Instanz |
| getSrvSearchBase() | Server-Suchbasis |
| getFilePath() | Dateipfad |
| getNotificationEmail() | E-Mailadresse zum Benachrichtigen |

# Synchronisationsablauf

## Verbindungsaufbau und hubObjekte erstellen

Die Synchronisation beginnt durch einen manuellen Start oder durch einen Cron-Job (siehe 7. Starten der Synchronisation). Dabei werden alle Anbindungen gruppiert nach hubObjekt-Typ in der folgenden Reihenfolge synchronisiert:

Beispiel:

Für jede Anbindung für den hubObjekt-Typ hubUser wird folgender Ablauf der Methoden durchgegangen:

Sind alle Anbindungen eines hubObjekt-Typs durch, werden die dazugehörenden ILIAS-Objekte aufgebaut. Dazu wird der Status jedes Eintrages in den hubObjekt-Tabellen (Middleware) ermittelt.

## hubObjekt-Status

### Allgemein

Daten werden in diese Middleware-Tabellen immer mittels „UPDATE INTO ...“ gespeichert, d.h. dass die Tabellen zu jedem Zeitpunkt die aktuelle Datenlage sowie die vergangenen (und nicht mehr gelieferten) Einträge beinhalten.

Unabhängig von den Einträgen, welche durch die Drittsysteme geliefert werden, können Synchronisierte Objekte auch in ILIAS geändert oder gelöscht werden (bspw. ein Kurs wird durch einen Administrator gelöscht oder eine Kategorie wird an einen anderen Ort im Baum verschoben.)

### Ermitteln des Status

Um den Status zu ermitteln, wird für jedes hubObjekt sein hubSyncHistory-Objekt hinzugezogen:

Macintosh HD:Users:fschmid:Desktop:SyncAblauf.pdf

### Statusmodell

Dadurch ergeben sich für jedes der synchronisierten Objekte vier Status:

|  |  |
| --- | --- |
| Statusname | Beschreibung |
| NEW | Der Eintrag ist neu in der Middleware-Tabelle und wurde noch nie synchronisiert |
| UPDATED | Der Eintrag wurde vom Drittsystem wieder geliefert |
| DELETED | Der Eintrag wurde vom Drittsystem nicht mehr geliefert |
| DELETED\_IN\_ILIAS | Der Eintrag wurde vom Drittsystem wieder geliefert, das dazugehörende ILIAS-Objekt (z.B. Kurs oder Benutzerkonto in ILIAS) existiert aber nicht mehr . |
| ALREADY\_DELETED | Eintrag wurde durch Hub bereits gelöscht und kann fortan ignoriert werden |
| NEWLY\_DELIVERED | Objekt wurde von Hub zu einem früheren Zeitpunkt gelöscht, der Eintrag wird nun aber erneut geliefert. Aktuell werden diese Einträge als UPDATED behandelt. |
| IGNORED | Wird in speziellen Fällen verwendet, um Einträge direkt zu übergehen (bspw. Kursmitgliedschaften, welche nicht mehr dem aktuellen Semester entsprechen, Beispiel siehe 7.1 getCurrentPeriod. |

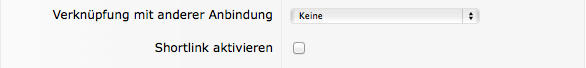
# Konfiguration

## Grundlagen-Konfiguration

Siehe 3.1.1 Erstellen einer Schnittstelle.

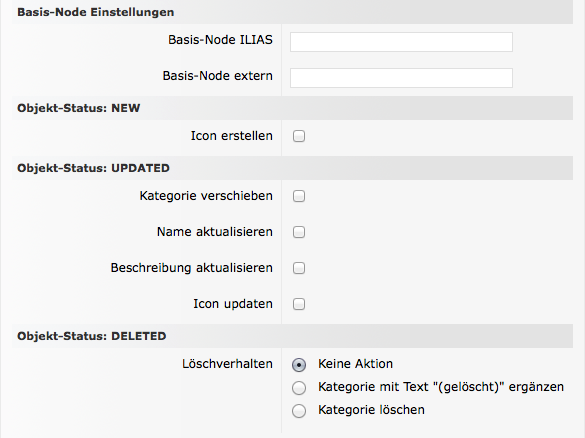
## Synchronisation je hubObjekt-Typ

Grundlegende Einstellungen:



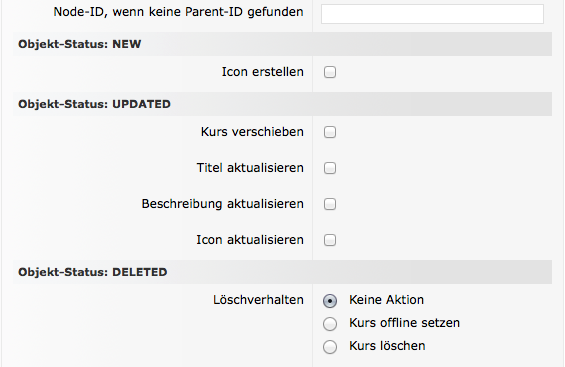
* Wenn in einer Anbindung die ID einer anderen Anbindung benötigt wird, kann diese hier ausgewählt werden.
* Shortlink aktivieren. Wird bei einem hubObjekt das Feld ‚shortlink’ gefüllt, kann direkt auf dieses über die folgende URL zugegriffen werden: /Customizing/global/plugins/Services/UIComponent/  
  UserInterfaceHook/Hub/shortlink.php?q=shortlinkwert

### hubCategory



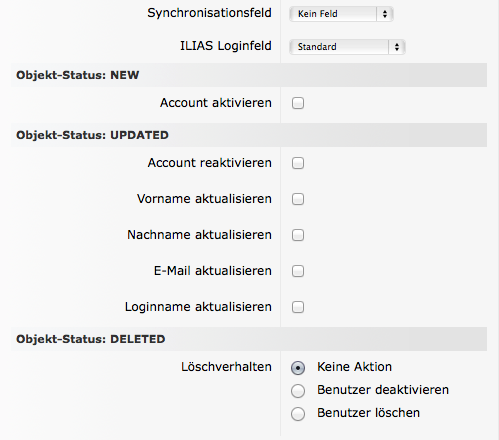
* Oberste ID im Baum: z.B. Magazin = 1
* Oberste ID im Drittsystem

### hubCourse



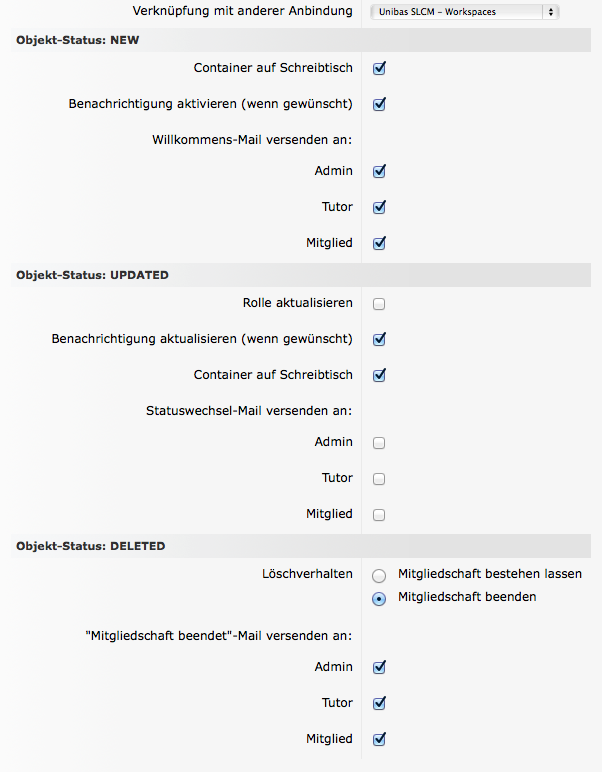
* Node-ID, wo Kurse angelegt werden, wenn keine gültige Parent-ID übergeben wurde (Fallback)

### hubUser



* Synchronisationsfeld: Sucht nach Benutzern, welche bereits im System sind anhand dieses Feldes und verknüft diese für die Synchronisation, damit keine neuen Benutzerkonten angelegt werden.
* ILIAS-Loginfeld: Form, nach welcher der Login-Name in ILIAS generiert werden soll.

### hubMembership



# Starten der Synchronisation

## Manueller Start

In der Schnittstellen-Verwaltung kann die Synchronisation für die aktiven Anbindungen mit dem Button „Synchronisierung starten“ ausgelöst werden.

## Cron-Job

Ein Cronjon kann in der folgenden Form eingerichtet werden:

30 0,12 \* \* \* www-data /usr/bin/php /var/www/ilias/ /Customizing/global/plugins/Services/UIComponent/UserInterfaceHook/Hub/cron.php benutzername passwort client > /dev/null

# Hooks

## getCurrentPeriod

Hub-Membership fragt die Origin ab, welche die aktuelle Periode (int) ist (z.b. Semester). Standardmässig wird NULL zurückgeben und damit werden alle Einträge verarbeitet. Wird aber eine Periode zurückgegeben, wird nur diese verarbeitet und die anderen erhalten den Status IGNORED.

## afterObjectInit

Nachdem ein hubObject verarbeitet wurde erhält die Origin die Möglichkeit, Änderungen am hubObject oder am dazu gehörenden ILIAS-Objekt vorzunehmen.

### appendFieldsToPropForm

Eigene Fomularelemente können so an das Einstellungsformular angefügt werden. Alle Werte sind über in der Origin $this->props()->get(„KEY“); abrufbar.

# Vorlage Synchronisationsregeln

Für die Synchronisation in ILIAS wurden folgende Aktionen je Status und Objekttyp festgelegt:

## Benutzerkonten (hubUsers)

|  |  |
| --- | --- |
| Statusname | Aktion |
| NEW |  |
| UPDATED |  |
| DELETED |  |
| DELETED\_IN\_ILIAS |  |

## Kategorien (hubCategories)

|  |  |
| --- | --- |
| Statusname | Aktion |
| NEW |  |
| UPDATED |  |
| DELETED |  |
| DELETED\_IN\_ILIAS |  |

## Workspaces (hubCourses)

|  |  |
| --- | --- |
| Statusname | Aktion |
| NEW |  |
| UPDATED |  |
| DELETED |  |
| DELETED\_IN\_ILIAS |  |

## Kursmitgliedschaft (hubMemberships)

|  |  |
| --- | --- |
| Statusname | Aktion |
| NEW |  |
| UPDATED |  |
| DELETED |  |
| DELETED\_IN\_ILIAS |  |

1. Lieferung meint, dass der Eintrag in der Tabelle zum ersten Mal geschrieben wurde oder aber auch, wenn der Eintrag wieder geliefert wurde, also beim Update eines Eintrages. Damit müssen alle Änderungen eines Eintrages auch mit dem aktuellen Zeitpunkt versehen werden. [↑](#footnote-ref-1)