

CLAIMING BY PLUGINS

Technisches Feinkonzept

Version 0.1 | 30.04.2014
Jörg Lützenkirchen | leifos GmbH
luetzenkirchen@leifos.com
+49.221.1207.18922

1 ZIELE

Features von ILIAS, die sich besonders für das Customizing von Installationen eignen, sollen mit Plugin-Slots versehen werden, die eine Kontrolle der GUI und damit der vorhandenen Daten ermöglichen. In einem ersten Schritt werden die erweiterten Metadaten und die benutzerdefinierten Felder mit einem eigenen Plugin-Slot versehen. Diese Slots werden jedoch auf einer klar definierten, gemeinsamen technischen Basis erstellt, um zukünftige Änderungen und Erweiterungen zu vereinfachen.

2 DIE PERMISSION-HELPER

Die GUIs, die optional unter die Kontrolle von Plugins gebracht werden sollen, müssen um Rechteabfragen erweitert werden, die über die Möglichkeiten der RBAC hinausgehen. Dabei werden für einen bzw. den aktuellen Benutzer und der aktuellen Objekt-Instanz („ref_id“) spezifische Kontexte und Aktionen definiert.

2.1 Kontext

Ein Kontext stellt eine Untereinheit einer ILIAS-Objekt-Instanz dar, z.B. einen Datensatz der erweiterten Metadaten. Diese Untereinheiten sind über das RBAC nicht zu verwalten. Ein Kontext besteht immer aus einem Typ und einer Id.

2.2 Aktion

Eine Aktion im Sinne der Permission-Helper findet immer zu einem Kontext statt. Sie entspricht prinzipiell den einzelnen Rechten (bzw. „operations“) des RBAC. Da die Permission-Helper sich vor allem auf GUI-basierte Prozesse konzentrieren, steht jeder Button und jeder Link für eine eigene Aktion – sofern die Verfügbarkeit dieser Elemente über ein Plugin kontrollierbar sein soll.

3 PLUGIN-SLOTS

Für jedes Features und dessen GUIs ist ein eigener Plugin-Slot zu definieren. Die spezifischen Plugins dieses Slots müssen gegen die für dieses Feature definierten Kontexte und Aktionen programmiert werden. Ebenso sind Hilfsmethoden, um z.B. Datenbankseinträge vorzunehmen, nicht Feature-übergreifend umzusetzen.

4 KONTROLLE DER ERWEITERTEN METADATEN

Die erweiterten Metadaten sollen per Plugin verwaltet werden können. Dies umfasst die Datenbank-Einträge der Datensätze und der einzelnen Felder. Ebenso sollen einzelne Elemente und Funktionen der GUIs kontrolliert werden können.

4.1 Plugin

Das Plugin muss in folgendem Pfad abgelegt werden:

/Customizing/global/plugins/Services/AdvancedMetaData/<Plugin-Name>

Entsprechend gestaltet sich die Klassen-Signatur:

```
include_once("../Services/AdvancedMetaData/classes/class.ilAdvancedMDClaimingPlugin.php");  
class <Plugin-Name> extends ilAdvancedMDClaimingPlugin
```

Weitere Infos zu Plugins finden sich im ILIAS Dev-Guide, Kapitel 31.

4.2 Kontexte und Aktionen

Die die Permission-Helper eine Vielzahl von Kontexten und Aktionen bieten, soll im Plugin auf ein Mapping „Methode zu (Kontext-)Aktion“ verzichtet werden. Folglich müssen die Plugins im Wesentlichen nur eine einzige Methode implementieren (`checkPermission`). Die Parameter-Werte sind durch Konstanten eindeutig definiert und bieten für beide Seiten (ILIAS und Plugin) eine größere Flexibilität, nicht zuletzt auch für zukünftige Erweiterungen.

Die User-Id als Parameter und ein Boolean als Rückgabe sollten selbsterklärend sein. Nachfolgend werden die restlichen Parameter dieser Methode und deren möglichen Werte gelistet.

4.2.1 Kontexte

Die erweiterten Metadaten bieten 6 Kontexte und ihre jeweiligen Ids:

- `CONTEXT_MD`: der Administrationsbereich der AMD („ref_id“)
- `CONTEXT_RECORD`: die Datensätze („record_id“)
- `CONTEXT_FIELD`: die Felder („field_id“)
- `CONTEXT_SUBSTITUTION`: Präsentation im Magazin („obj_type“)
- `CONTEXT_SUBSTITUTION_COURSE`: Präsentation im Magazin („field_id“)¹
- `CONTEXT_SUBSTITUTION_CATEGORY`: Präsentation im Magazin („field_id“)

4.2.2 Aktionen

Jeder Kontext bietet seine eigenen, eindeutigen Aktionen:

- `CONTEXT_MD`
 - `ACTION_MD_CREATE_RECORD`
 - `ACTION_MD_IMPORT_RECORDS`
- `CONTEXT_RECORD`
 - `ACTION_RECORD_EDIT`
 - `ACTION_RECORD_DELETE`²
 - `ACTION_RECORD_EXPORT`
 - `ACTION_RECORD_TOGGLE_ACTIVATION`
 - `ACTION_RECORD_EDIT_FIELDS`
 - `ACTION_RECORD_CREATE_FIELD`
 - `ACTION_RECORD_FIELD_POSITIONS`
 - `ACTION_RECORD_EDIT_PROPERTY`³

¹ Es werden nur die Objekttypen „Kurs“ und „Kategorie“ unterstützt. (Org-Einheiten?)

² Da jeder Datensatz einzeln „verwaltet“ wird, können Multi-Aktionen in Tabellen evtl. angeboten werden, jedoch für keinen einzigen Datensatz möglich sein. Auf eine weitere, verknüpfte Aktion wie z.B. `CONTEXT_MD_DELETE_RECORDS` soll verzichtet werden.

³ Bei Aktionen, die Unteraktionen bieten, wird immer eine Unteraktion gesetzt. Also ist z.B. `ACTION_RECORD_EDIT` (keine Unteraktion) von `ACTION_RECORD_EDIT_PROPERTY` (mit Unteraktion) zu unterscheiden.

- SUBACTION_RECORD_TITLE
 - SUBACTION_RECORD_DESCRIPTION
 - SUBACTION_RECORD_OBJECT_TYPES
- CONTEXT_FIELD
 - ACTION_FIELD_EDIT
 - ACTION_FIELD_DELETE
 - ACTION_FIELD_EDIT_PROPERTY
 - SUBACTION_FIELD_TITLE
 - SUBACTION_FIELD_DESCRIPTION
 - SUBACTION_FIELD_SEARCHABLE
 - SUBACTION_FIELD_PROPERTIES
- CONTEXT_SUBSTITUTION
 - ACTION_SUBSTITUTION_SHOW_DESCRIPTION
 - ACTION_SUBSTITUTION_SHOW_FIELDNAMES
 - ACTION_SUBSTITUTION_FIELD_POSITIONS
- CONTEXT_SUBSTITUTION_COURSE
 - ACTION_SUBSTITUTION_COURSE_SHOW_FIELD
 - ACTION_SUBSTITUTION_COURSE_EDIT_FIELD_PROPERTY
 - SUBACTION_SUBSTITUTION_BOLD
 - SUBACTION_SUBSTITUTION_NEWLINE
- CONTEXT_SUBSTITUTION_CATEGORY
 - ACTION_SUBSTITUTION_CATEGORY_SHOW_FIELD
 - ACTION_SUBSTITUTION_CATEGORY_EDIT_FIELD_PROPERTY
 - SUBACTION_SUBSTITUTION_BOLD
 - SUBACTION_SUBSTITUTION_NEWLINE

4.3 Datenbank-Hilfsmethoden

Innerhalb des DB-Update-Skriptes des Plugins können folgenden Methoden verwendet werden, um Datensätze, Felder und die Präsentation zu verwalten:

- `bool hasDBRecord()`
- `int createDBRecord()`
- `bool updateDBRecord()`
- `bool deleteDBRecord()`
- `bool hasDBField()`
- `int createDBField()`⁴
- `bool updateDBField()`
- `bool deleteDBField()`
- `bool setDBSubstitution()`
- `bool setDBFieldSubstitution()`
- `bool removeDBFieldSubstitution()`

5 KONTROLLE DER BENUTZERDEFINIERTEN FELDER

⁴ Neue Datensätze werden immer an der letzten Position angehängt.

