

Inhalt

1	Einleitung	2
2	Allgemeines	2
3	Struktur der Plugin XML-Datei	2
3.1	Definition Tag	3
3.2	Menu Tag.....	3
3.2.1	Group Tag.....	4
1.1.1	Point Tag	4
1.1.2	Context Tag	4
4	Plugin Seiten.....	5
4.1	plugin_showpage.php	5
4.1.1	plugin_generateLink()	5
2	Rechtliche Hinweise / Impressum.....	6
2.1	Fragen und Anregungen	6
2.2	Impressum	6
2.3	Rechtliche Hinweise	6

1 Einleitung

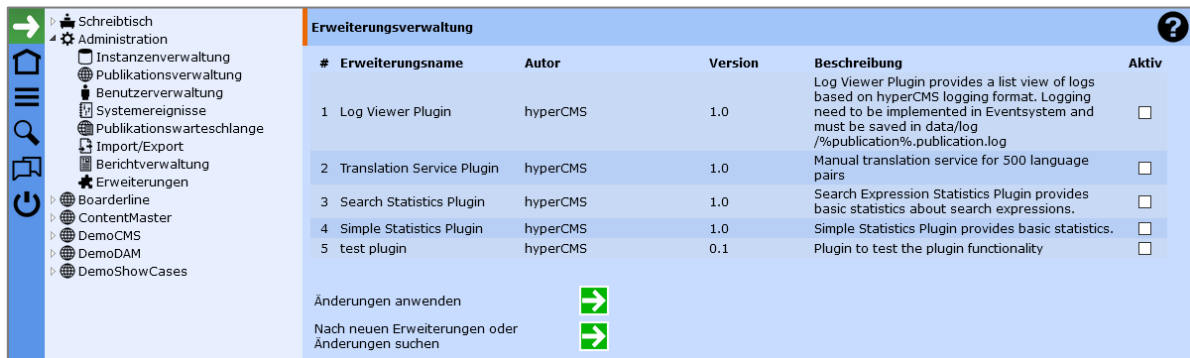
Der hyper Content & Digital Asset Management Server kann mittels Plugins (Erweiterungen) um zusätzliche Funktionen erweitert werden.

Dabei können völlig eigenständige Anwendungen mit Hilfe des hyperCMS API umgesetzt werden.


Im Installationspaket von hyperCMS ist ein Test-Plugin, das als einfache Vorlage dienen kann, enthalten (hypercms/plugin/test).


Die Erweiterungsverwaltung zeigt alle im System installierten Plugins an und erlaubt die Aktivierung und Deaktivierung einzelner Plugins.

Nur aktivierte Plugins können in den Benutzergruppen genutzt bzw. zugeordnet werden.



#	Erweiterungsname	Autor	Version	Beschreibung	Aktiv
1	Log Viewer Plugin	hyperCMS	1.0	Log Viewer Plugin provides a list view of logs based on hyperCMS logging format. Logging need to be implemented in Eventsystem and must be saved in data/log/%publication%.publication.log	<input type="checkbox"/>
2	Translation Service Plugin	hyperCMS	1.0	Manual translation service for 500 language pairs	<input type="checkbox"/>
3	Search Statistics Plugin	hyperCMS	1.0	Search Expression Statistics Plugin provides basic statistics about search expressions.	<input type="checkbox"/>
4	Simple Statistics Plugin	hyperCMS	1.0	Simple Statistics Plugin provides basic statistics.	<input type="checkbox"/>
5	test plugin	hyperCMS	0.1	Plugin to test the plugin functionality	<input type="checkbox"/>

Änderungen anwenden 

Nach neuen Erweiterungen oder Änderungen suchen 

2 Allgemeines

Für jedes Plugin muss ein Ordner im „hypercms/plugins/“ Ordner angelegt werden. Innerhalb dieses Ordners muss sich eine „plugin.xml“ Datei befinden damit das System das Plugin erkennt. Diese XML-Datei enthält die Grunddaten eines Plugins sowie die Menüpunkte über die man auf das Plugin zugreifen kann.

3 Struktur der Plugin XML-Datei

Die plugin.xml Datei muss folgendem XML-Format entsprechen:

```
<?xml version="1.0" encoding="UTF-8" ?>
<plugin>
  <definition>
    <name>My Plugin</name>
    <author>hyperCMS</author>
    <version>1.0</version>
    <description>Just a test Plugin</description>
  </definition>
  <menus>
    <main>
    </main>
    <publication>
    </publication>
  </menus>
</plugin>
```

3.1 Definition Tag

Hierbei handelt es sich um die Basisdaten eines jeden Plugins. Folgende Felder müssen einen Wert besitzen: Name, Author, Version, Description

```
<definition>
  <name>My Plugin</name>
  <author>hyperCMS</author>
  <version>1.0</version>
  <description>Just a test Plugin</description>
</definition>
```

3.2 Menu Tag

Hierbei handelt es sich um Menüpunkte die zum Navigator von hyperCMS hinzugefügt werden. Die Tags main und publication bestimmen, ob es sich dabei um einen allgemeinen oder einen publikationsspezifischen Menüpunkt handelt. Des weiteren können auch Einträge des Kontextmenüs im Objekt-Explorer definiert werden.

```
<menus>
  <main>
    <group>
      <name>Navigation Group 1</name>
      <icon>img/group.gif</icon>
      <subpoints>
        <point>
          <name> Navigation Item 1.1</name>
          <icon>img/item.gif</icon>
          <page>pages/page.php</page>
          <control>pages/control.php</control>
        </point>
        <point>
          <name> Navigation Item 1.2</name>
          <icon>img/item.gif</icon>
          <page>pages/page.php</page>
        </point>
      </subpoints>
    </group>
  </main>
  <publication>
    <point>
      <name>Publication Navigation Item 1</name>
      <icon>img/item.gif</icon>
      <page>pages/page.php</page>
    </point>
  </publication>
  <context>
    <point>
      <name>Send to translation</name>
      <icon>img/language.gif</icon>
      <page>pages/page.php</page>
    </point>
  </context>
</menus>
```

3.2.1 Group Tag

Dieser Tag fasst mehrere Menüpunkte zu einer Gruppe zusammen. Der Aufbau ist folgender:

```
<group>
  <name></name>
  <icon></icon>
  <subpoints></subpoints>
</group>
```

Name, Icon und Subpoints müssen vorhanden sein, damit diese Gruppe im Navigator des Systems angezeigt wird.

<name>

Name des Menüpunktes im Navigator.

<icon>

Der Icon Tag sollte den Link zu einem Bild (16 x 16 Pixel) relativ zum Ordner des Plugins enthalten. Das Bild sollte sich innerhalb des Plugin-Ordners befinden.

<subpoints>

In den Subpoints befinden sich einzelne Point Tags. Weitere Group Tags sind jedoch nicht möglich.

3.2.2 Point Tag

Dieser Tag beschreibt einen Menüpunkt und ist folgendermaßen aufgebaut:

```
<point>
  <name></name>
  <icon></icon>
  <page></page>
  <control></control>
</point>
```

Name, Icon und Page müssen vorhanden sein, damit das System diesen Menüpunkt in den Navigator aufnimmt.

<name>

Name des Menüpunktes im Navigator.

<icon>

Für den Icon Tag gelten die gleichen Regeln wie beim Icon in der Group.

<page>

Dieser Tag enthält den Link zur Hauptseite des Arbeitsplatzes relativ zum Ordner des Plugins. Diese Seite wird als Hauptseite via `plugin_showpage` angezeigt.

<control>

Dieser Tag enthält den Link zu der Arbeitsplatzsteuerung (optional) relativ zum Ordner des Plugins.

3.2.3 Context Tag

Der Kontext Tag erweitert das Kontextmenü des Objekt-Explorers. Alle zur Verfügung stehenden Daten einer Objektauswahl werden dabei an das Plugin weitergeleitet (POST Request).

Die verfügbaren Variablen können wie folgt aus dem POST Request ermittelt werden:

```
$action = getrequest ("action", "objectname");  
$multiobject = getrequest ("multiobject");  
$location = getrequest_esc ("location", "locationname");  
$page = getrequest_esc ("page", "objectname");  
$contexttype = getrequest_esc ("contexttype"); // contextmenu context-types (folder,  
object, media)  
$token = getrequest ("token");
```

Der übermittelte Token kann wie folgt geprüft werden:

```
$valid = checktoken ($token, $user); // true/false
```

Wurden mehrere Objekte ausgewählt, so enthält die Variable \$multiobject eine Auflistung aller Objektpfade als String mit | als Trennzeichen.

Die folgende Anweisung gibt ein Array aller Objektpfade zurück:

```
$multiobject_array = link_db_getobject ($multiobject);
```

4 Plugin Seiten

4.1 plugin_showpage.php

Diese Seite wird dazu verwendet die Plugin Seiten anzuzeigen, inklusive der Arbeitsplatzsteuerung, falls gewünscht.

4.1.1 plugin_generateLink()

Diese Funktion kann dazu verwendet werden einen Link zu einer Seite zu generieren dabei sollten die Parameter in folgender Reihenfolge übergeben werden:

\$plugin:	Identifikator/Name des Plugins.
\$page:	Link zur Hauptseite relativ zum Ordner des Plugins
\$control:	Link zur Arbeitsplatzsteuerung relativ zum Ordner des Plugins
\$getParameters:	Zusätzliche Get Parameter die an die Seite weitergesendet werden sollen (z.B. "site=myPublication&action=show")

5 Rechtliche Hinweise / Impressum

5.1 Fragen und Anregungen

Sollten Sie weitergehende Fragen oder Anregungen zum Produkt haben, so wenden Sie sich bitte an den Support.

hyperCMS Support:

support@hypercms.com

<http://www.hypercms.com>

5.2 Impressum

Verantwortlich für den Inhalt:

hyperCMS
Content Management Solutions GmbH
Rembrandtstr. 35/6
A-1020 Wien - Austria

info@hypercms.com
<http://www.hypercms.com>

5.3 Rechtliche Hinweise

Vorliegende Installationsanleitung basiert auf der zum Zeitpunkt der Verfassung des Dokumentes verfügbaren Programmversion.

Der Hersteller behält sich Programmänderungen und –Verbesserungen vor.

Fehler und Irrtümer vorbehalten.

© 2026 by hyperCMS Content Management Solutions