PubMan Import - Technical Documentation

Version: 1.0
Date: 16.08.13
State: Final
Index

1 INTRODUCTION AND DEPENDENCIES ........................................................................................................3
2 STRUCTURE ..................................................................................................................................................4
3 OVERVIEW ....................................................................................................................................................5
   3.1 PLUGIN - BACKEND ..................................................................................................................................5
   3.2 PLUGIN – ADMI GUI ..................................................................................................................................6
   3.3 HELPER CLASSES AND FUNCTIONS ......................................................................................................7
4 INTERACTION DIAGRAM FORCE UPDATE ..................................................................................................8
1 Introduction and Dependencies

The PubMan Import plugin is used to import documents from PubMan and create Wordpress blog posts. PubMan provides a REST interface which is used for the communication between the plugin and PubMan.

The plugin PubMan Import is written in PHP 5.3 for Wordpress 3.5.2
For the admin GUI the plugin uses jQuery which comes with Wordpress.
No external libraries are installed with the plugin.
# 2 Structure

The plugin consists of various files in different directories.

<table>
<thead>
<tr>
<th>Directory</th>
<th>Filename</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>/css</td>
<td>admin.css</td>
<td>CSS for admin GUI</td>
</tr>
<tr>
<td>/includes</td>
<td>pmi_html.php</td>
<td>Helper class to generate HTML forms</td>
</tr>
<tr>
<td></td>
<td>pmi_log.php</td>
<td>Logging class of the plugin</td>
</tr>
<tr>
<td></td>
<td>pmi_rest.php</td>
<td>Communication with REST interface and XML parsing</td>
</tr>
<tr>
<td></td>
<td>pmi_wp.php</td>
<td>Creating and modifying of Wordpress posts</td>
</tr>
<tr>
<td>/js</td>
<td>admin.js</td>
<td>Javascript for admin GUI</td>
</tr>
<tr>
<td>/lang</td>
<td>plugin.po</td>
<td>Language file for plugin (not used in the moment)</td>
</tr>
<tr>
<td>/views</td>
<td>admin.php</td>
<td>Definition of admin GUI</td>
</tr>
<tr>
<td>/</td>
<td>pubman.php</td>
<td>Plugin main class</td>
</tr>
</tbody>
</table>
3 Overview

3.1 Plugin - Backend

The PubMan Import plugin consists of five classes.
The class PubManImport is the main class and has the following functions:

- Implements the required functions and hooks of the Wordpress API a plugin must have
- Functions for the main features of the plugin like cron_trigger(), forceUpdate() and manualUpdate()
- Logic to get all data from PubMan even if there are more than 5000 results: getAllDataFromPubMan()
- Delegates calls to PMIRest for communication with the PubMan REST interface
- Delegates calls to PMIWp for creating Wordpress posts/drafts

The class PMIRest is responsible for communicating with the PubMan REST interface and for parsing the XML from PubMan.
The class PMIWp is doing all Wordpress specific operations. It contains the logic to create a single Wordpress post depending on the configured settings.

Furthermore the class has two methods deletePMIPosts() and deleteEmptyCategoriesAndTags() which are used by the Force Update feature to delete all existing posts and all empty tags and categories.
The plugin must know which posts it created in order to make updates or delete them again.
Therefore a custom field with the name PMI_ESCIDOC_ID is added to each post. The value of the attribute is the ESCIDOC_ID of the item in PubMan.
The custom field and its value can be seen when editing a post in Wordpress.
3.2 Plugin – Admi GUI

What’s called Admin GUI is the settings page of the PubMan Import plugin in Wordpress.

The files css/admin.css, js/admin.js and views/admin.php are responsible to render the settings page and add the required functions.

The Javascript library jQuery which ships with Wordpress is used in the Admin GUI for the navigation and for hiding/showing certain fields like Link Default Text.

Furthermore it is used to call the required actions for Force Update, Manual Update and Delete Log.

It is also used to schedule the automatic update when the settings are saved. Therefor an AJAX call is sent to the sever.

The following table shows how the functions, action parameter in the query string and the methods in the class PubManImport are connected.

<table>
<thead>
<tr>
<th>Function</th>
<th>Action Parameter</th>
<th>Method in class PubManImport</th>
</tr>
</thead>
<tbody>
<tr>
<td>Force Update</td>
<td>pmi_force_update</td>
<td>forceUpdate()</td>
</tr>
<tr>
<td>Manual Update</td>
<td>pmi_manual_update</td>
<td>manualUpdate()</td>
</tr>
<tr>
<td>Delete Log</td>
<td>pmi_delete_log</td>
<td>deleteLog()</td>
</tr>
<tr>
<td>Schedule Automatic Update</td>
<td>pmi_change_update_interval</td>
<td>reset_schedule()</td>
</tr>
</tbody>
</table>
3.3 Helper Classes and Functions

There are two helper classes used in the plugin. The class `PMIHtml` is used to create HTML input elements in the Admin GUI. It makes the definition of the Admin GUI more readable and understandable.

The second class is `PMILog` which is responsible for the logging of the plugin. The name of the log file is defined as `log.txt` in the class itself and is saved in the PubMan Import plugin folder.

There are two constants defined in the `PubManImport` class which configure the logging:

- `PMI_DEBUG` can be `TRUE` or `FALSE` and defines if debug logging should be activated.
- `PMI_MAX_LOG_SIZE` defines the maximum size of the log file. If the log file gets bigger, it will be deleted and re-created.
4 Interaction Diagram Force Update

This interaction diagram shows the important calls for a Force Update.

```
[foreach ($posts as $post)]:processSinglePost($post)
                              [auto_update == true]:getWPPostsForEscidocID(escidoc_id) :array
                              [foreach ($postsID as $postID)]:createOrUpdatePost($postID)
                              [if ($fulltextLink != "")]:createPubManLink(fullTextView Parameters) :String
                              [if ($post['license'] <> "")] :createPubManLink(CCLicense Parameters) :String
                              [if ($postID != "")]:wp_update_post()
                              [else]:wp_insert_post()
                              wp_set_post_tags()
                              wp_set_post_terms()
                              add_post_meta(escidoc_id)
```

```
getAllFromPubMan() :posts array
getDataFromPubMan() :XML data
parseXML(XML data) :posts array
processPosts(posts array)
```
The program flow for a Manual Update and Automatic Update is quite the same, except that the messages `deletePMIPosts()` and `deleteEmptyCategoriesAndTags()` are not sent to the PMIWp class.