

Benutzerhandbuch - RDF2WP

Gruppe: swp6-11

Inhaltsverzeichnis

1	Installation	1
1.1	WordPress Plugin RDF2WP	1
1.2	Einrichtung der Packete	1
1.2.1	Package Import	1
1.2.2	Package Widget	2
1.2.3	Package Output	2
1.2.4	Package Editor	3
2	Nutzung	3
2.1	Import	3
2.2	Export	4
2.3	Eingabe von Daten in Blogeinträge	6
2.3.1	Syntax	6
2.3.2	Präfixe	8
2.3.3	Templates	8
2.4	Widgets	10
2.5	RSS-Feed	11

1 Installation

1.1 WordPress Plugin RDF2WP

Die Installation des RDF2WP Plugins entspricht der Standard-Plugin-Installation in Wordpress.

Sollte Ihnen das Plugin in einer .zip- Datei vorliegen, so kann es über die WordPress Schnittstelle im Admin-Bereich installiert werden. Gehen Sie dafür in das Settings-Menü „Plugins“ Eintrag „Add New“ und wählen in der oberen Link-Leiste den Eintrag Upload. Hier kann der .zip Ordner von der lokalen Festplatte hochgeladen und installiert werden. Anderenfalls lässt sich das Plugin durch einfaches Kopieren des Plugin Ordners „RDF2WP“ in den Plugin Ordner Ihrer Lokalen Wordpress Installation hinzufügen. Dieser befindet sich in den häufigsten Fällen unter:

<Pfad zur WP-Installation>/wp-content/plugins.

Möchten Sie einige der Pakete nicht nutzen, so können Sie diese in der Datei RDF2WP.php in den Zeilen 11 bis 53 auskommentieren. So kommentiert man z.B. das Widget Paket aus:

```
Zeile 18: //include_once('Widget/KSCWidget.php');
```

```
Zeile 19: //$widget = new KSCWidget();
```

Welcher Teil zu einem Package gehört, ist dort ausdrücklich kommentiert.

1.2 Einrichtung der Pakete

Im Folgenden werden die Komponenten des Plugins (Packages) beschrieben, sowie einige Änderungen die sie an ihnen vornehmen können.

1.2.1 Package Import

Das Import-Package bezieht sich auf den Import des Professorenkatalogs der Universität Leipzig, ist aber durchaus in der Lage RDF Ressourcen aus anderen Endpunkten in die Wordpress Installation einzubinden. WordPress-Pfad: Die Importschnittstelle befindet sich im Admin-Bereich Menü-Eintrag „Settings“ Unterpunkt „Import Professorum Catalogum Lipsiensis“.

Plugin-Pfad:

RDF2WP/Import/

Empfehlung:

Es wird empfohlen das Collapsing Categories Plugin¹ zu installieren, um eine hierarchische Darstellung der erzeugten Kategorien zu ermöglichen.

1.2.2 Package Widget

Mit dem Widget-Package wird die Anzeige von RDF Daten in Wordpress ermöglicht. Ein frei konfigurierbares Widget zeigt immer die aktuellen Informationen in der Widget-Sidebar an.

Plugin-Pfad:

RDF2WP/Widget/

1.2.3 Package Output

Das Output-Package sorgt zum einen dafür, dass RDF Daten welche sich in einem Post befinden, besucherfreundlich angezeigt werden. Zum anderen stellt es eine Schnittstelle für den Export von RDF Daten (Linked Data) bereit. Und es bringt einen RDF Parser mit sich, welcher die Daten validiert und dem Verfasser bei Fehlern hilfreiche Hinweise auf die Ursache liefert.

Plugin-Pfad:

RDF2WP/Output/

Anpassung der Darstellung von Bildern:

Um die Darstellung von Bildern in einem Post anzupassen ist eine Änderung in der Datei *ViewSemantics.php* erforderlich. Wenn ein bestimmtes Prädikat auf Bilder verweisen soll, so muss dieses in das Array `$images` in Zeile 123 eingefügt werden, zB:

```
$images = array("http://www.klappstuhlclub.de/wp/thumbnail",  
"http://catalogus-professorum.org/cpm/picture");
```

¹<http://wordpress.org/extend/plugins/collapsing-categories>

1.2.4 Package Editor

Das Editor-Package bringt ein bequemes Template System mit sich. Für eine schnelle Eingabe von semantischen Daten in einen Post, werden zwei Buttons dem Editor hinzugefügt. Diese sind für die Klappstuhlclub Mitglieder interessant. Allerdings können die Templates auch angepasst werden.

Plugin-Pfad:

RDF2WP/Output/

Das Anpassen eines Templates:

Um z.B. das Meeting-Template anzupassen, müssen Sie eine kleine Veränderung der Datei *MeetingTemplateDialog.php* vornehmen. Dazu öffnen Sie bitte diese Datei in einem Texteditor Ihrer Wahl. In den Zeilen 21 bis 28 sehen Sie schon alles was Sie brauchen. Bsp. Zeile 21:

```
$template->addEntry('| nummer = "300"', true);
```

Der erste Parameter '| nummer = "300"' ist eine Textzeile, welche später beim anklicken des Buttons dem Editor hinzugefügt werden soll.

Der zweite Parameter gibt an, ob diese Zeile in der Checkbox des Templates vorab ausgewählt werden soll. true bedeutet Sie wird ausgewählt, false hingegen heißt bei dieser Zeile wird das Häkchen nicht gesetzt. Wollen Sie eine neue Zeile hinzufügen, so kopieren Sie eine der Zeilen 21 bis 28 und fügen Sie diese an.

So kann z.B. Ihre neue Zeile aussehen:

```
$template->addEntry('| essen = "altes Brötchen", "Pflaume"', false);
```

2 Nutzung

2.1 Import

Die Importfunktion dient dazu, Daten aus dem Import Catalogus Professorum Lipsiensis in das WordPress zu importieren. Sie befindet sich im Menü *Settings* unter dem Eintrag *Import Catalogus Professorum Lipsiensis* (Abb. 1).

Die Importseite gliedert sich in drei Bereiche. Im obersten Bereich werden die Einstellungen für den Import vorgenommen. In den Einstellungen kann der SPARQL Endpunkt sowie eine entsprechende Anfrage definiert werden.

Limit legt fest, wie viele Ressourcen in den Blog übernommen werden. Wird kein Limit angegeben, so werden alle Ressourcen die sich im Ergebnis befinden importiert. Beachten Sie hierbei, dass das Interface nur Anfragen die sich auf eine RDF Klasse beziehen, wie es beim Professorenkatalog die Professoren sind, ordnungsgemäß importiert. Von dieser Klasse können außerdem beliebig viele Prädikate abgefragt werden. Abfragen ähnlich, die des „Graphical Query Builders“² sind somit kein Problem.

Da jede Ressource als eigener Post im Blog erscheinen wird, kann bei den Einstellungsmöglichkeiten neben *Posts* bestimmt werden, wie WordPress diese behandeln soll. Für die angefragten Attribute werden jeweils eigene Kategorien erstellt. Alle Instanzen der angegebenen Klasse werden in eine Oberkategorie mit dem Namen dieser Klasse importiert. Alle erstellten Kategorien werden in einer Oberkategorie abgelegt. Ihr Standardname ist „Catalog“. Der Name lässt sich auf einfache Weise anpassen, indem in der Datei *CategoryCreator.php* die Konstante „mainCategory“ auf den gewünschten Namen gesetzt wird. Mit dem Button *Änderungen übernehmen* werden die Einstellungen gespeichert. Um den Import durchzuführen, muss der Haken neben *Check* gesetzt werden und der *Daten Importieren* Button betätigt werden.

2.2 Export

Die Exportfunktion dient dazu, die semantischen Daten des Blogs in verschiedenen Formaten auszugeben. Das kann unter anderem dazu dienen, die Daten in einen Triple Store einzufügen. Um die semantischen Daten eines Posts oder des gesamten WordPress als Linked Data zu exportieren, wird das *Linked Data Script* verwendet. Zum Export des gesamten Inhaltes des WordPress im NTriples-Format, wird es wie folgt aufgerufen:

```
http://<servername>/<wordpress-ordner>/wp-content/plugins/RDF2WP/Output/  
EntireBlogExporter.php?format=text/plain
```

²<http://catalogus-professorum.org/graphicalquerybuilder/display/position/3/active?m=http%3A%2F%2Fcatalogus-professorum.org%2Fflipsiensis%2F>

Abbildung 1: Import von 10 Professoren aus dem Professorenkatalog

Ein einzelner Post kann wie folgt exportiert werden:

`http://<servername>/<wordpress-ordner>/wp-content/plugins/RDF2WP/Output/linkedDataScript.php?id=381&format=text/plain`

Hier kann der Parameter “*format*” folgenden Werte annehmen:

<code>application/rdf%2Bxml</code>	für die Ausgabe im RDF/XML Format
<code>application/json%2Brd</code>	für die Ausgabe im JSON Format
<code>text/rdf%2Bn3</code>	für die Ausgabe in N3
<code>text/plain</code>	für die Ausgabe in N-Triples

Alternativ kann der Export der Daten eines Posts über die Links unterhalb der HTML Tabelle erfolgen (Abb. 2).

Meeting 301
by swp6 — Categories: Uncategorized — [Leave a comment](#) — [Edit](#)

nummer	301
datum	Tue, 20 Dec 2011 21:00:00 +0100
stadt	Leipzig
anwesend	gustav
anwesend	peter
besucher	Walt Disney
aktivitaeten	Filme gucken
trilogie	Filmtrilogie
temperatur_celsius	40.0

[RDF](#) | [JSON](#) | [N3](#) | [NTriples](#)

Abbildung 2: Ausgabe der semantischen Daten in den entsprechenden Formaten

2.3 Eingabe von Daten in Blogbeiträge

2.3.1 Syntax

Die semantischen Daten müssen im Post zwischen den Tags `[semantics format="infobox"] <Daten> [/semantics]` für das infobox Format bzw. `[semantics format="prof"] <Daten> [/semantics]` für Daten des Professorenkatalogs geschrieben werden. Die Daten des Professorenkatalogs müssen im Format *Turtle* sein.

Das infobox Format kann folgendermaßen beschrieben werden:

Das *Subjekt* eines Triples ist stets der Permalink des Posts. Ein *Prädikat* und das zugehörige *Objekt* wird durch eine Zeile: `|prädikat = objekt` beschrieben.

Dabei kann das Prädikat ein beliebiges Wort sein, mehrere Wörter werden durch einen Unterstrich getrennt.

Das Objekt wird, falls es ein Literal ist, in Anführungszeichen eingeschlossen:
`|praedikat = "literal"`

Wenn das Literal dem Datentyp *DateTime* oder *float* des XML Schemas entspricht, so muss dies mit angegeben werden:

```
|praedikat = "2011-06-22T21:00:00+02:00"^^xsd:datetime
```

```
|praedikat = "23.5"^^xsd:float
```

Wenn das Objekt eine *URI* ist, wird es in spitze Klammern gesetzt oder es wird ein gültiger Präfix verwendet (nächster Abschnitt):

```
|praedikat = <http://www.klappstuhlclub.de/wp/posts/308-treffen-leipzig>
```

Wenn ein gültiges Tripel erkannt wurde, wird dies in der *RDF-Parser Konsole* angezeigt. Andernfalls wird dort ein Fehler ausgegeben (Abb. 3).

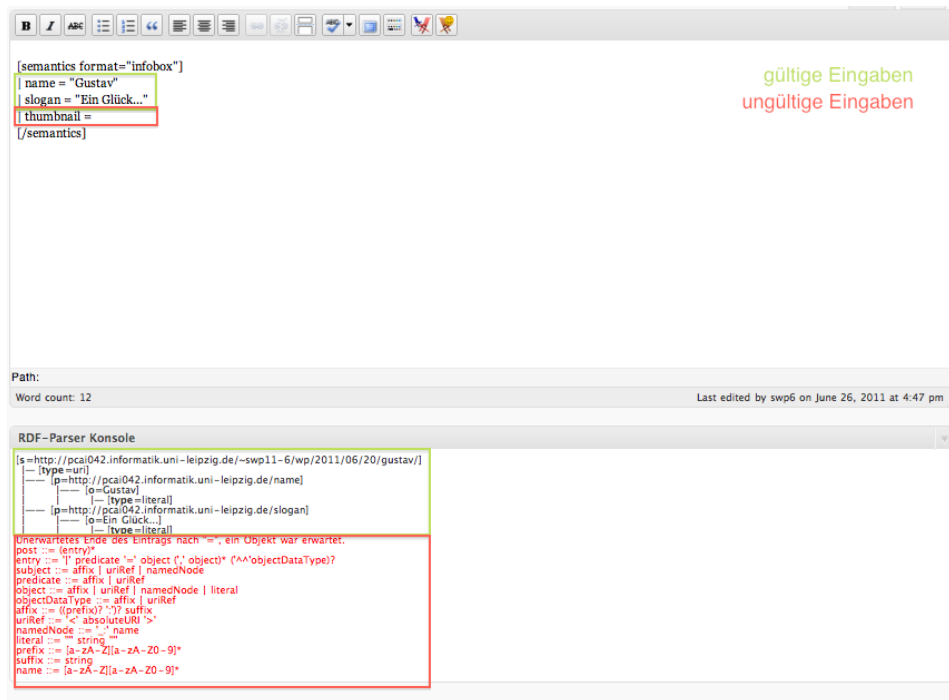


Abbildung 3: Ausgabe der RDF-Parser Konsole

2.3.2 Präfixe

Um die Darstellung von URIs zu vereinfachen, kommen Präfixe zum Einsatz. Sie können im Menüpunkt *RDF Einträge* unter *Settings* angepasst werden (Abb. 4).

Alle Wörter ohne Präfix, die nicht als Literal gekennzeichnet sind, werden das Präfix der ersten Zeile erhalten. Andere Präfixe müssen mit einem Doppelpunkt gekennzeichnet werden:

```
[semantics format="infobox"]
| stadt = "Leipzig"
| anwesend = :Wolf, :Claus, vip:Joerg
[/semantics]
```

In diesem Beispiel würden *stadt* und *anwesend* jeweils den Präfix `http://www.klappstuhlclub.de/wp/` erhalten, da hier kein Präfix angegeben wurde. *Leipzig* erhält keinen Präfix, weil es als Literal gekennzeichnet ist. *Wolf* und *Claus* erhalten den Präfix `http://www.klappstuhlclub.de/wp/mitglieder`, da ihnen das leere Präfix “:” zugewiesen wurde. *Joerg* erhält den Präfix `http://www.klappstuhlclub.de/wp/vips`, da er den Präfix “vip:” trägt.



Abbildung 4: Einstellung der Präfixe

2.3.3 Templates

Um die Eingabe von Daten in Blogbeiträge zu erleichtern, stehen im Editor zwei Templates zur Verfügung (Abb. 5). Diese bieten die grundlegende Syntax des Formats *infobox* für ein Treffen des Klappstuhlclubs oder ein Mitglied des Klappstuhlclubs.

Beim Aktivieren einer der Template Buttons öffnet sich ein Pop-up (Abb. 6), in dem das Template mit Hilfe von Checkboxes angepasst werden kann. Mit dem *Insert* Button wird das Template in den Editor eingefügt.

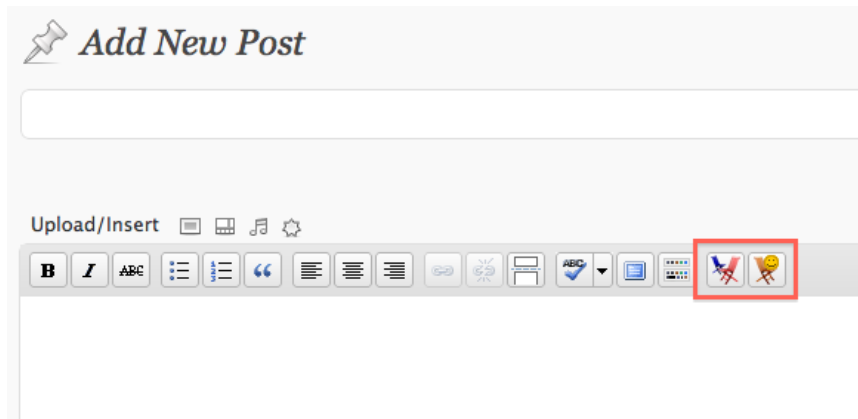


Abbildung 5: Templates des Klappstuhlclubs

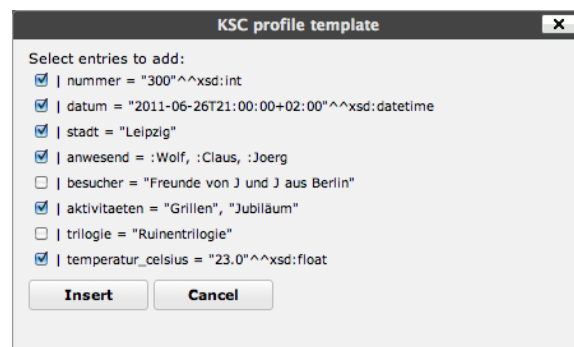


Abbildung 6: Anpassung des Templates

2.4 Widgets

Um Daten aus dem SPARQL Endpunkt abzufragen und darzustellen, wird das Widget *RDF* benutzt. Es kann im *Widgets* Menü mehrmals in das *Sidepad* gezogen werden, um mehrere Anfragen darzustellen (Abb. 7).

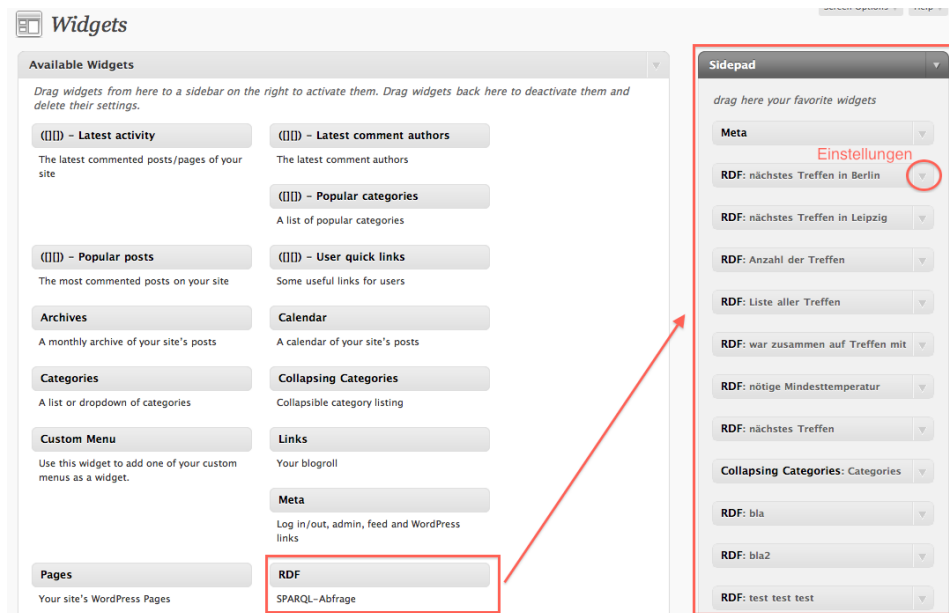
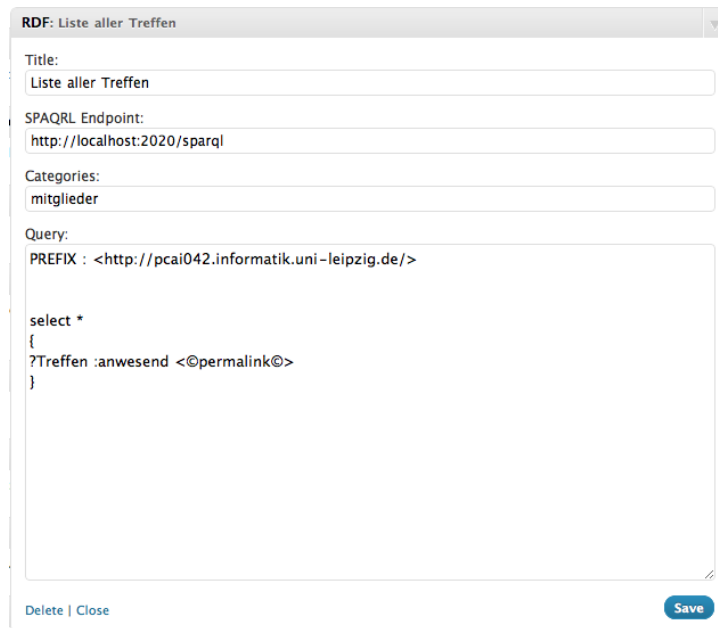


Abbildung 7: Widgets in das Sidepad einfügen

In den Einstellungen des Widgets können Anpassungen zur Erscheinung und Ausgabe des Widgets vorgenommen werden (Abb. 8). Mit *Title* wird die Überschrift der Ausgabetable angepasst. *SPARQL Endpoint* muss die Adresse des SPARQL Endpunkts beinhalten, dem die Query gestellt werden soll. In *Categories* wird festgelegt, in welchen Top-Level Kategorien des Blogs das Widget erscheinen soll. Die einzelnen Kategorien werden durch ein Komma getrennt. Falls Sie dieses Feld frei lassen, so ist das Widget uneingeschränkt sichtbar. Schließlich wird unter *Query* die SPARQL Anfrage eingegeben, die von dem Widget dargestellt werden soll. Um die Widgets dynamisch zu konfigurieren, kann der Platzhalter `<©permalink©>` in der SPARQL Query verwendet werden, der jeweils durch den Permalink des aktuellen Posts ersetzt wird.



RDF: Liste aller Treffen

Title:
Liste aller Treffen

SPARQL Endpoint:
http://localhost:2020/sparql

Categories:
mitglieder

Query:
PREFIX : <http://pcai042.informatik.uni-leipzig.de/>

select *
{
?Treffen :anwesend <@permalink@>
}

Delete | Close Save

Abbildung 8: Widget konfigurieren, hier: Die Liste aller Treffen die ein Mitglied besucht hat, sichtbar in der Kategorie *Mitglieder*

2.5 RSS-Feed

Der RSS-Feed ist standardmäßig im WordPress enthalten und über:
`http://<servername>/<wordpress-ordner>/feed` erreichbar.