

Arbeitsplan

Wordpress-Plugin für Ontologiebasierte
Datenevaluation

- 1. Projektvision**
- 2. Voraussetzungen**
- 3. Designübersicht und Funktionalität**
- 4. Arbeitspakete**
- 5. Vorprojekt**
- 6. Glossar**

1. Projektvision

Der Behindertenverband Leipzig hat über die Jahre eine große Datenbank mit Informationen zur Barrierefreiheit von öffentlichen Gebäuden und Plätzen in Leipzig aufgebaut. Diese Informationen sind über die Website des BVL zugänglich bzw. einsehbar. Zur einfacheren Erfassung, Verwaltung und Verarbeitung der Daten ist ein neues System notwendig.

Ziel dieses Projekts ist es, ein Wordpress Plugin zu entwickeln, das diese Informationen zu Gebäuden und anderen Plätzen über HTML Formulare erfasst, an Ontologien auswertet und in Semantic Web Graphen abspeichert. Die Formulare sollen aus den jeweiligen Ontologien, passend zur Ressource generiert werden. Der Nutzer soll die Möglichkeit haben, erfasste Daten zu verwalten, sowie Ontologien zu löschen bzw. hinzuzufügen.

Außerdem wird Wert auf die barrierefreie Nutzung des kompletten Plugins gelegt, da es primär für den Behindertenverband Leipzig entwickelt wird.

2. Voraussetzungen

Für die erfolgreiche Umsetzung des Projekts bedarf es folgender Voraussetzungen:

- Auf der internen Ebene muss sich jedes Teammitglied bei Slack, Taiga und Gitlab registrieren, sowie mit der Bedienung dieser Dienste vertraut sein, um die Kommunikation zu gewährleisten bzw. zu strukturieren. Dies erfordert unter anderem Einrichtungsaufwand auf Seiten des Projektleiters.

Zudem wird durch wöchentlich angeordnete Treffen sichergestellt, dass alle Teammitglieder auf dem aktuellem Stand des Projektfortschrittes sind und ihre Aufgaben kennen.

- Auf technischer Seite benötigt jedes Teammitglied sowie der Entwicklungsserver eine einheitliche WordPress-Installation, da die Entwicklung des Plugins direkt von WordPress abhängig ist. Selbiges gilt für eine MySQL-Datenbank, sowie eine einheitliche PHP Version.¹

Da Wordpress mindestens eine PHP Version von 5.2.4 (5.6 empfohlen) und MySQL Version 5.0 (5.6 empfohlen) voraussetzt, werden wir PHP 5.6 und MySQL 5.6 verwenden.

Der Entwicklungsserver wird dabei von David Geistert gestellt und es wird vorausgesetzt, dass genügend Speicherplatz sowie Rechenleistung zur Verfügung stehen.

Um das Einhalten der festgelegten Programmierstandards zu gewährleisten, muss jeder Entwickler PHPUnit (zur Funktionalitätsprüfung des Codes) und CodeSniffer (zur Einhaltung des PSR-2 Coding-Standards) in seine Entwicklungsumgebung einbinden.

Weiterhin werden voraussichtlich mehrere Zusatz-Bibliotheken wie Twig, FormComponent und Dependency Injection eingebunden und über die PHP Paketverwaltung Composer installiert.

¹ <https://wordpress.org/about/requirements/>

- Um Allgemein die einfache und barrierefreie Bedienung des Endproduktes realisieren zu können, müssen alle Teammitglieder über die fachgerechte Umsetzung dieser Anforderungen informiert sein. Dies wird durch den Recherchebericht gewährleistet.

Seitens des Teams wird gefordert, dass jedes Mitglied die nötige Zeit und Motivation aufbringt.

3. Designübersicht und Funktionalität

Use Case 1: Hochladen einer Ontologie und Generierung von Formularen daraus

Beschreibung: Der Administrator möchte eine neue Ontologie hochladen und aus dieser soll dann ein Formular erstellt werden, das die gewünschten Daten des Nutzers erfragt.

Funktionalität: Dem Administrator wird eine HTML-Seite angezeigt, auf der über ein Dateiauswahlfeld eine Ontologie hochgeladen werden kann. Nach dem Upload wird der Administrator auf eine Seite weitergeleitet, auf der er auswählt, welche Klassen bzw. Properties der Ontologien in dem späteren Formular verarbeitet werden. Dazu wird ein Twig Code generiert, welcher frei vom Administrator bearbeitbar ist.

Ziel: Der Administrator kann Ontologien hochladen, um Daten anderer Ressourcen zu erfassen.

Rollen: Administrator

Use Case 2: Löschen einer Ontologie

Beschreibung: Der Administrator möchte eine Ontologie löschen.

Funktionalität: Dem Administrator wird eine HTML-Seite angezeigt, auf der er eine oder mehrere Ontologien aus einer Liste hochgeladener Ontologien auswählen und löschen kann. Wird eine Ontologie gelöscht, dann werden ebenfalls alle verknüpften Formulare gelöscht. Die verknüpften Ressourcen bleiben jedoch erhalten.

Ziel: Der Administrator kann nicht benötigte Ontologien löschen.

Rollen: Administrator

Use Case 3: Formular einer bereits hochgeladenen Ontologie bearbeiten

Beschreibung: Der Administrator möchte das aus der Ontologie erzeugte Formular bearbeiten.

Funktionalität: Dem Administrator wird eine HTML-Seite angezeigt, auf der er eine Ontologie aus einer Liste der hochgeladenen Ontologien auswählt. Danach wird ihm ein Fenster angezeigt, in dem er den Namen bzw. den Twig Code der Ontologie bearbeiten kann.

Ziel: Ein Formular kann vom Administrator entsprechend angepasst werden.

Rollen: Administrator

Use Case 4: Ressourcen eintragen

Beschreibung: Der Benutzer möchte Daten zu einem Platz bzw. Gebäude in das System eintragen.

Funktionalität: Dem Benutzer wird eine HTML-Seite angezeigt, auf der er die Ontologie bzw. das Formular aus einer Liste auswählt, in der/dem die Eingabedaten gespeichert werden sollen. Der User wird danach auf eine Seite weitergeleitet, auf der er, in die vorgesehenen Eingabefelder, die Daten eintragen kann. Diese werden anhand der Ontologie validiert und als RDF-Daten gespeichert.

Ziel: Der Benutzer gibt valide Daten in das Formular ein und speichert diese.

Rollen: Benutzer

Use Case 5: Ressourcen bearbeiten

Beschreibung: Der Benutzer möchte die Daten einer bestehenden Ressource ändern.

Funktionalität: Der Benutzer wählt die zu ändernde Ressource aus einer Liste aus und wird auf ein

HTML-Formular weitergeleitet, auf dem die existierenden Daten bearbeitet werden können.

Ziel: Der Benutzer kann die Daten der Ressource bearbeiten.

Rollen: Benutzer

Use Case 6: Ressourcen löschen

Beschreibung: Der Benutzer möchte Ressourcen aus dem System löschen.

Funktionalität: Dem Benutzer wird eine HTML-Seite angezeigt, auf der einzelne Ressourcen ausgewählt und gelöscht werden können.

Ziel: Nicht benötigte Ressourcen können gelöscht werden

Rollen: Benutzer

4. Arbeitspakete

4.1 Mussziele

4.1.1 Grundstruktur des Plugins 10%

Das Projekt soll als Plugin für das CMS Wordpress entwickelt werden. Aufgrund fehlender Strukturvorgaben der Plugin-Entwicklung seitens Wordpress, läuft man als Entwickler schnell Gefahr, ein Plugin ohne erkennbare Struktur zu programmieren. Um dieser Gefahr entgegen zu wirken, wird sich das erste Arbeitspaket ausschließlich mit der Entwicklung eines Plugin Cores beschäftigen. Dieser Plugin Core bindet jegliche externen Frameworks wie z.B. Twig oder FormComponent ein und stellt diese u.a. in Form eines abstrakten Controllers dem Projekt zur Verfügung. Des Weiteren legt der Core die grundlegende Ordnerstruktur (Controller, Assets, Templates, ...) des Projektes fest.

4.1.2 Hinzufügen/Löschen von Ontologien 10%

Um eine Ontologie in den Datenbestand aufzunehmen, muss ein HTML Formular erstellt werden, mit dem man eine Ontologie hochladen kann. Außerdem muss eine Übersichtsseite entwickelt werden, welche die hochgeladenen Ontologien anzeigt. Nach Bedarf können Ontologien auch gelöscht werden.

Mit dem Löschen einer Ontologie werden auch die dazu gehörenden Formulare gelöscht. Allerdings bleiben die Ressourcen im RDF Graphen vorhanden.

4.1.3 Verarbeitung der Ontologie 30%

Nachdem eine Ontologie hochgeladen wurde, muss der Benutzer zu einem Formular-Erstellungsassistenten geleitet werden. Dort soll auswählbar sein, welche Klassen bzw. welche Properties der einzelnen Klassen in dem Formular dargestellt werden sollen. Außerdem soll die Möglichkeit bestehen, die Grundstruktur des Formulars zu bearbeiten bzw. einzelne Felder über Twig Themes zu individualisieren.

4.1.4 Anlegen neuer Ressourcen 25%

Die generierten Formulare müssen angezeigt werden, um dem Benutzer die Eingabe von Werten zu ermöglichen. Nach dem Abschicken des Formulars müssen die Daten nach der Ontologievorschrift validiert werden. Ggf. werden Fehler angezeigt und der Benutzer wird zu Korrektur der eingetragenen Werte gebeten.

Bei erfolgreicher Validierung der Daten erfolgt die Speicherung im RDF Graphen.

4.1.5 Verwaltung eingetragener Ressourcen 15%

Um eine Bearbeitung der eingetragenen Ressourcen zu ermöglichen, muss eine Übersichtsseite entwickelt werden, welche die eingetragenen Ressourcen, nach Ontologie sortiert, anzeigt. Die angezeigten Ressourcen müssen sowohl löschar als auch editierbar sein.

4.1.6 Barrierefreiheit 10%

Das Plugin wird primär für den Behindertenverband Leipzig e.V. programmiert. Um eine bestmögliche Bedienung zu gewährleisten, muss jegliche Art der Bedienung des Plugins barrierefrei sein.

4.2 Kannziele

4.2.1 Responsive Design 10%

Es ist später angedacht, das Plugin auch über Mobile Geräte u.a. Handy und Tablet zu bedienen. Dazu müssen alle Forms per CSS Responsive gestaltet werden.

4.2.2 Barrierefreiheit WP-Admin 5%

Es ist zu evaluieren, ob die Navigation durch das Wordpress Admin-Interface zum eigentlichen Plugin barrierefrei ist. Sollte dies nicht der Fall sein, muss das Plugin eine Option bereitstellen, die eine CSS Datei einbindet, welche die Wordpress Standard Styles überschreibt und somit eine barrierefreie Navigation ermöglicht.

4.2.3 Ajax Validierung/Speicherung 10%

Um die Bedienung der Forms für den Benutzer bequemer zu gestalten, kann die Validierung und das Abschicken der Formulare per Ajax erfolgen.

4.2.4 Mehrsprachigkeit 5%

Um eine internationale Bedienung des Plugins zu ermöglichen, ist es nötig, alle Texte in Templates mehrsprachig einzubinden. Dies könnte mittels des Symfony Translator Component oder der Wordpress internen Translation API realisiert werden.

4.2.5 Suchen/Sortieren 10%

Um eine bessere Bedienung der Verwaltungsseite der Ressourcen zu gewährleisten, sollte es dem Nutzer ermöglicht werden, Ressourcen gezielt zu suchen oder nach gewissen Parametern zu sortieren.

4.2.6 RDF Dump exportieren 10%

Um die erfassten Ressourcen auch außerhalb des Plugins zu verwenden, wäre es hilfreich, eine Exportfunktion zu erstellen.

5. Vorprojekt

Im Rahmen des Vorprojektes soll die grundlegende Struktur des Plugins angelegt werden und die Ziele des ersten Arbeitspaketes umgesetzt werden. Auf der Basis der Grundstruktur des Plugins soll ein "Click Dummy" mittels Twig Templates umgesetzt werden.

Des Weiteren sollen erste Funktionalitäten wie das Hinzufügen und Löschen von Ontologien in den Click Dummy implementiert werden. Dies schließt ein System, welches Ontologien verarbeiten kann und daraus ein HTML Formular erzeugt, mit ein.

Außerdem müssen Grundfunktionalitäten zum Auflisten der Formulare bzw. Ontologien entwickelt werden.

6. Glossar

6.1 HTML

HTML wird genutzt, um digitale Dokumente zu strukturieren und Seiten im Webbrowser darzustellen. HTML Dokumente bestehen aus einem Doctype, welcher den verwendeten Dokumentationstyp angibt, einem Head, in dem technische oder dokumentarische Informationen gespeichert werden und einem Body, der die angezeigten Informationen enthält.²

6.2 WordPress

Mithilfe von Wordpress lassen sich Inhalte von Websites dynamisch verwalten. Mittels Plugins kann die Funktionalität von Wordpress erweitert werden. WordPress basiert auf der Skriptsprache PHP und benötigt eine MySQL-Datenbank.³

6.3 PHP

PHP ist eine serverseitige Skriptsprache, die hauptsächlich zur Erstellung dynamischer Websites oder Webanwendungen verwendet wird. PHP zeichnet sich durch eine große Anzahl an Funktionsbibliotheken aus, wie z.B. Saft, Composer, Twig oder Form Component.⁴

6.4 Plugin

Ein Modul, welches eine Software erweitert oder verändert. Plugins werden während der Laufzeit von der Anwendung, welche sie erweitern, eingebunden und können nicht ohne selbige ausgeführt werden.⁵

6.5 Ontologien

In Ontologien wird eine Menge von Begrifflichkeiten eines gemeinsamen Gegenstandsbereiches sowie deren Beziehungen in einer sprachlichen und formalen Darstellung abgebildet.

² <https://www.w3.org/TR/2014/REC-html5-20141028/>

³ <https://de.wordpress.org/>

⁴ <http://php.net/manual/de/index.php>

⁵ https://codex.wordpress.org/Writing_a_Plugin

Zudem enthalten sie Regeln zur Schlussfolgerung und zur Gewährleistung ihrer Gültigkeit. Durch Ontologien wird ein Netzwerk von Informationen mit logischen Relationen dargestellt.⁶

6.6 CSS

CSS ist eine Stylesheet-Sprache mit der Aufgabe, die Gestaltung von HTML-Dokumenten anzupassen, wodurch der Inhalt und das Layout getrennt werden.⁷

6.7 Ajax

Ajax ist ein Verfahren, welches mithilfe von JavaScript, Daten über den Server schickt und auf dessen Antwort wartet. Mithilfe dieser Funktion, können wir die Eingabedaten, anhand des Servers und des so verbundenen SAFT-Framework sofort evaluieren.⁸

6.8 WP-Admin

Der WP-Admin beschreibt den Administrationsbereich in einem WordPress-System, über welchen alle Aktionen auf einer WordPress Website durchgeführt werden können; wie z.B. das Erweitern oder Löschen von Daten.

So kann jemand mit Zugang, Nutzer hinzufügen und löschen. Dies ist zu beachten, wenn jemand Zugang zum WP-Admin bekommt.⁹

6.9 Benutzer

Der Benutzer besitzt nur eingeschränkte Rechte, welche ihm zwar Zugang zum WP-Admin ermöglichen, jedoch nur Grundoperationen wie das Anlegen, Löschen und Bearbeiten von Ressourcen erlauben.

6.10 Administrator

Der Administrator agiert als Superuser und hat umfassende Rechte, welche es ihm ermöglichen, Ontologien und Formulare hochzuladen, zu bearbeiten und zu löschen.

Des Weiteren besitzt er auch alle Rechte eines *Benutzers*.

6.11 Ressource

Eine Ressource ist ein Objekt mit einer Identität. Ressourcen können untereinander über Prädikate verbunden sein, wodurch Eigenschaften bzw. Zusammenhänge beschrieben werden.

In unserem Fall sind dies Plätze und Gebäude, sowie deren Eigenschaften. Insbesondere in Hinsicht auf deren Barrierefreiheit.

⁶ <https://www.w3.org/standards/semanticweb/ontology>

⁷ <http://www.w3.org/TR/2000/WD-css3-roadmap-20000414>

⁸ <http://searchwindevelopment.techtarget.com/definition/Ajax>

⁹ <http://www.wpbeginner.com/glossary/administrator/>