

Entwurfsbeschreibung

DTP-16: LD Hypstrr - Datadriven Template Publication

6. Juni 2016

INHALTSVERZEICHNIS

| | | |
|----------|--|----------|
| 1 | Allgemeines | 3 |
| 2 | Produktübersicht | 3 |
| 3 | Grundsätzliche Struktur- und Entwurfsprinzipien | 3 |
| 3.1 | Jekyll-Plugins | 3 |
| 4 | Struktur- und Entwurfsprinzipien einzelner Pakete | 3 |
| 4.1 | RdfMainGenerator | 3 |
| 4.2 | RdfPageData | 4 |
| 4.3 | RdfTemplateMapper | 4 |
| 4.4 | RdfTerm | 4 |
| 4.5 | RdfLiteral | 4 |
| 4.6 | RdfResource | 4 |
| 4.7 | RdfStatement | 5 |
| 4.8 | RdfProperty | 5 |
| 4.9 | RdfPropertyList | 5 |
| 4.10 | PropertyQueryFilter | 5 |
| 5 | Datenmodell | 5 |
| 6 | Glossar | 6 |
| 6.1 | FOAF | 6 |
| 6.2 | Graph | 6 |
| 6.3 | HTML | 6 |
| 6.4 | Linked Data | 6 |
| 6.5 | Literal | 6 |
| 6.6 | Metadaten | 6 |
| 6.7 | Namespaces bzw. Namensräume | 6 |
| 6.8 | Notation3 bzw. N3 | 6 |
| 6.9 | Ontologie | 7 |
| 6.10 | OntoWiki | 7 |
| 6.11 | SPARQL | 7 |
| 6.12 | RDF | 7 |
| 6.13 | RDFS | 7 |
| 6.14 | URI | 7 |
| 6.15 | Turtle | 8 |
| 6.16 | Triple-Store | 8 |
| 6.17 | W3C | 8 |

1 ALLGEMEINES

Siehe Handbuch

2 PRODUKTÜBERSICHT

Unser Plugin Jekyll-RDF erweitert das Static Site Generation Tool Jekyll um die Funktionalität, für jedes Subjekt eines Tripels eines RDF-Graphen eine eigene HTML-Ressource zu rendern. Auf jeder einzelnen Seite pro Subjekt werden schließlich standardmäßig alle mit diesem Subjekt in Verbindung stehenden Prädikate und Objekte tabellarisch aufgelistet. Darüber hinaus ist es möglich, über Template-Mappings in der Konfigurationsdatei spezielle, nur für diese URIs bestimmte Layouts anzugeben und diese zu rendern. Es ist möglich die RDF-Dateitypen ttl ("turtle"), nt ("n-triples"), RDF-XML (rdf bzw. xml), n3 und trig fehlerfrei zu verarbeiten und eine Einschränkung des RDF-Graphen durch eine (in der "_config.yml" hinterlegten) SPARQL-SELECT Query vorzunehmen.

3 GRUNDSÄTZLICHE STRUKTUR- UND ENTWURFSPRINZIPIEN

3.1 JEKYLL-PLUGINS

Jekyll-Plugins werden in Kategorien unterteilt, die in verschiedene Prozesse von Jekyll eingreifen. Es stehen folgende Kategorien zur Auswahl: Generators (greifen in den Generator-Prozess ein, wenn beispielsweise weitere Daten geladen werden sollen), Converters (übersetzen spezielle Informationen in Markdown-Dateien zu HTML-Content), Commands (ermöglichen die Einführung eines neuen Subcommands für Jekyll), Tags (ermöglichen die Einführung neuer Liquid-Tags) und Hooks (ermöglichen eigene Funktionen vor oder nach dem Rendering-Prozess einzelner Ressourcen wie zum Beispiel der kompletten Site, einzelner Pages oder Posts oder Dokumenten).

4 STRUKTUR- UND ENTWURFSPRINZIPIEN EINZELNER PAKETE

4.1 RDFMAINGENERATOR

Die erste Klasse MainGenerator erbt vom Jekll-Generator und definiert das Laden der RDF-Datei mithilfe der RDF.rb Library und des in der Config-Datei spezifizierten Pfades zur RDF-Datei. Der zu ladende RDF-Graph ist weiterhin durch die in der Config-Datei hinterlegte SPARQL-SELECT Query einschränkbar (durch die Methode `extract_resources`). Da die Klasse MainGenerator direkt den Graphen lädt, ist die Arbeit mit ttl, nt, n3, trig und rdf bzw. xml-Dateien möglich. Anschließend werden in einer for-Schleife für alle Ressourcen des Graphen diese Ressourcen in ein eigenes Datenmodell überführt (siehe RdfTerm) und für jede Ressource eine Page-Instanz erstellt.

4.2 RDFPAGEData

Bei der bereits erwähnten zweiten Klasse `RdfPageData`, die von der Klasse `Jekll-Page` erbt, wird die Ordnerstruktur der zu erstellenden HTML-Ressource sowie das Template-Mapping bestimmt. Darüber hinaus wird der Name der aktuellen Ressource als Titel zur Page gegeben sowie die eigene RDF-Ressource in den RDF-Datenbereich der Page, sodass später über Liquid Tags und Filter auf diese Informationen zugegriffen werden können.

4.3 RDFTEMPLATEMAPPER

Der Template-Mapper wird mit den Konfigurationsinformationen für Default-Template und ressourcen- und klassenspezifische Templates initialisiert. Anschließend gibt der Mapper bei Anfrage (Methode `map`) den Namen des zu verwendenden Templates zurück - wurde die URI bzw. eine Oberklasse der URI in der Konfiguration angegeben und gemappt wird das entsprechende Template zurückgegeben, ansonsten das Default-Template. Falls bei einer Ressource mehrere Templates infrage kommen (da die Ressource beispielsweise von mehreren Klassen erbt), so wird eine Warnung im Kommandofenster ausgegeben.

4.4 RDFTERM

Die Klasse `RdfTerm` erbt von der Klasse `Liquid::Drop` und beschreibt ein eigenes Datenmodell für die `RdfStatements`. Durch dieses neue Datenmodell werden wir unabhängig von der RDF-Library und können die Daten für unsere Zwecke speziell aufbereiten. Genauer wird es ermöglicht RDF-Terme in der Liquid Templating Engine zu repräsentieren und auf diesen zu arbeiten. Es bestehen Funktionalitäten zum Erstellen (Methode `initialize`) und Ausgeben (Methode `to_s`) solcher Terme, sowie zum Einfüttern dieser in die Seitendaten (Methode `self.build_term_drop`).

4.5 RDFLITERAL

Die Klasse `RdfLiteral` erbt von `RdfTerm` und repräsentiert RDF-Literale in der Liquid Templating Engine. Sie besitzt eine einzige Methode (Methode `name`), welche die Ausgabe als String ermöglicht.

4.6 RDFRESOURCE

Von der Klasse `RdfResource` werden Methoden zur Verfügung gestellt, welche die Arbeit mit einzelnen Ressourcen erleichtern. So ist es möglich durch die Methoden `statements_as_subject` bzw. `predicate` und `object` alle Statements zu finden, in der die angegebene URI als Subjekt, Prädikat oder Objekt auftaucht. Ebenso implementiert die Klasse die Methode `types`, welche alle Oberklassen einer URI zurückgibt um so klassenspezifische Templates zu ermöglichen. Weiterhin ermöglicht die Methode `filename` automatisch Dateinamen zu generieren. `RdfResource` erbt von `RdfTerm`.

4.7 RDFSTATEMENT

Die Klasse `RdfStatement` erbt von `Liquid::Drop` und repräsentiert RDF-Statements in der Liquid Templating Engine. Mit der Methode `initialize` ist es möglich die Statements zu erstellen und in die Seitendaten einzufüttern.

4.8 RDFPROPERTY

Die Klasse `RdfProperty` erbt von `Liquid::Filter`. Der Filter kann in den vom Benutzer erstellten Templates verwendet werden. Unter Angabe einer Prädikat-URI wird ein Objekt dargestellt, welches mit dem Subjekt, für das die aktuelle Seite gerendert wird, über eben diese URI in einer Beziehung steht. Es ist ebenso möglich die gewünschte Sprache anzugeben, indem nach dem Prädikat das Sprach-Tag angegeben wird (siehe Handbuch).

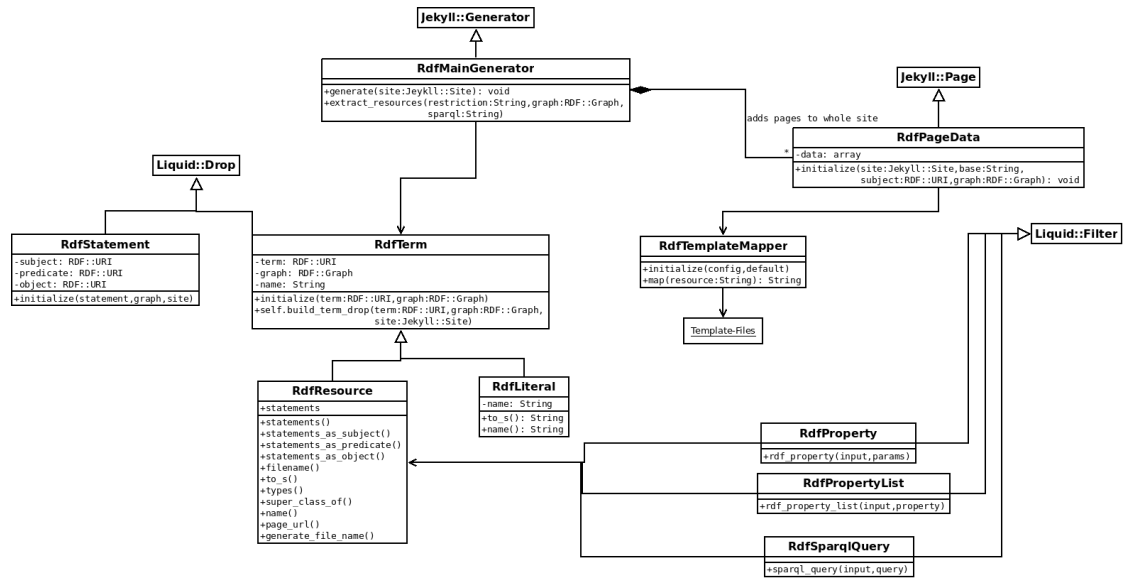
4.9 RDFPROPERTYLIST

Die Klasse `RdfPropertyList` erbt ebenso von `Liquid::Filter` und ist auch zur Verwendung in nutzeigenen Templates gedacht. Unter Angabe einer Prädikat-URI können hier alle Objekte dargestellt werden, die mit dem Subjekt, für das die aktuelle Seite gerendert wird, über diese URI in Beziehung stehen. Es ist ebenso möglich die gewünschte Sprache anzugeben, indem nach dem Prädikat das Sprach-Tag angegeben wird (siehe Handbuch).

4.10 PROPERTYQUERYFILTER

Die Klasse `PropertyQueryFilter` ist auch ein Filter, welcher allerdings unter Angabe einer Sparql-Select Query die Ergebnisse dieser darstellen kann.

5 DATENMODELL



6 GLOSSAR

6.1 FOAF

FOAF steht für "friend of a friend" und ist ein Projekt zur maschinenlesbaren Modellierung sozialer Netzwerke durch ein RDF-Schema, welches Klassen und Eigenschaften definiert. FOAF ist eines der ersten Anwendungen von Semantic-Web-Technologien.

6.2 GRAPH

Ein Graph ist eine abstrakte Struktur, die eine Menge von Objekten (Knoten) und deren Verbindungen (Kanten) darstellt. Eine Kante existiert immer nur zwischen genau zwei Knoten, ist also eine paarweise Verbindung, und kann gerichtet oder ungerichtet sein.

6.3 HTML

(kurz für: Hypertext Markup Language) ist eine vom W3C entwickelte Markup Sprache, die Texten und Daten Darstellungsformen und Eigenschaften zuweist. HTML-Daten setzen sich aus verschachtelten Tags zusammen und sind Multimedia-kompatibel.

6.4 LINKED DATA

Linked Data ist eine Methode zum Veröffentlichenden strukturierter Daten. Damit werden sie miteinander verknüpft und werden durch semantische Anfragen noch hilfreicher.

6.5 LITERAL

Ein Literal ist im informationstechnischen Sinne eine Folge von Zeichen, die zur direkten Darstellung der Werte von Datentypen zulässig ist. Dies gilt auch im RDF-Modell, wo eine Ressource nur durch eine andere Ressource oder ein Literal beschrieben werden kann.

6.6 METADATEN

Metadaten sind Daten, die Informationen über Eigenschaften anderer Dateien enthalten, jedoch nicht die Datei selbst. So gehört zum Beispiel zu den Metadaten eines Kinofilms der Name des Regisseurs, des Produktionsstudios, etc.

6.7 NAMESPACES BZW. NAMENSRÄUME

Namespaces ermöglichen es, Elemente mit gleichem Namen eindeutig voneinander zu unterscheiden, indem die Elemente und Attribute URIs zugewiesen werden.

6.8 NOTATION3 BZW. N3

Notation3 ist eine formale Sprache zur Beschreibung von semantischen Daten. Das bekannte Turtle bildet eine Untermenge.

6.9 ONTOLOGIE

Eine Ontologie (auch: Vokabular) ist eine sprachlich strukturierte Darstellung von Wissen. Es wird genutzt, um Beziehungen zwischen einzelnen Begriffen darzustellen. Dies ermöglicht mit einem standardisierten Vokabular auch eine Form der digitalen Verwendbarkeit in verschiedenen Systemen und erleichtert das logische Verstehen der Daten.

6.10 ONTOWIKI

OntoWiki ist eine Wiki-artige Software, die das Bearbeiten und Anzeigen von semantischen Daten zulässt. Die Besonderheit besteht darin, Komplexität der Datenspeicherung zu abstrahieren. Die Bedienung erfolgt Wiki-typisch über eine Weboberfläche.

6.11 SPARQL

SPARQL ist eine Abfragesprache für RDF-Daten. Daten können von verschiedenen RDF-Datenquellen in einer Anfrage abgefragt werden. Um die Lesbarkeit zu erhöhen ist es möglich Präfixe zu definieren. Als Ergebnis wird ein RDF-Graph oder ein Ergebnisset geliefert.

6.12 RDF

RDF (kurz für: Resource Description Framework) ist ein System zur Beschreibung von Ressourcen. Es ist eine technische Herangehensweise im Internet zur Formulierung logischer Aussagen über beliebige Ressourcen. Im RDF Modell besteht jede Aussage aus 3 Einheiten: Subjekt, Prädikat und Objekt.

6.13 RDFS

RDFS (kurz für: Resource Description Framework Schema) ist eine von der W3C entwickelte Ontologie-Beschreibungssprache für RDF-Daten. Mit Hilfe von RDFS lassen sich RDF-Ressourcen semantisch durch Eigenschaften und Relationen untereinander beschreiben, wodurch man Ontologien erzeugen kann. RDFS basiert auf der Idee eines mengentheoretischen Klassenmodells. Besitzt man zum Beispiel eine RDF Instanz von Fahrzeug und eine von Auto, so kann man mit RDFS festlegen, dass Fahrzeug eine Klasse und Auto eine Subklasse von Fahrzeug ist.

6.14 URI

URI (kurz für: Uniform Resource Identifier) ist ein Identifikator und besteht aus einer Zeichenfolge, die zur Identifizierung einer abstrakten oder physischen Ressource dient. URIs werden zur Bezeichnung im Internet und dort besonders im World Wide Web genutzt. URIs können als Zeichenfolge in digitalen Dokumente eingebunden werden („Hyperlink“). Aufbau: scheme, authority, path, query, fragment.

6.15 TURTLE

Turtle steht für "Terse RDF Triple Language" und ist eine Serialisierung für RDF-Graphen. Turtle-Dateien enden auf .ttl.

6.16 TRIPLE-STORE

Ein Triple-Store (auch: RDF-Store) ist eine Datenbank zum Speichern und Verwenden von Tripeln durch semantische Anfragen. Triple-Stores speichern wie relationale Datenbanken Informationen in Tripeln, jedoch wurden Triple-Stores dafür optimiert. Tripel können durch RDF oder andere Formate importiert und exportiert werden. Wenn man dem Tripel einen Namen gibt, hat man einen Quad-Store oder einen sogenannten „named graph“.

6.17 W3C

W3C (kurz für: World Wide Web Consortium) ist ein internationales Gremium, dessen Ziel es ist, durch zukunftsorientierte Web-Standards in Form von Protokollen und Richtlinien die gesamten Kapazitäten des World Wide Webs zu erschließen und dessen langfristiges Wachstum zu gewährleisten. Es verfolgt genauer die Vorteile des Internets für jedermann zugänglich zu machen, unabhängig von Soft-, Hardware, Sprache, Kultur, etc., sowie den Aufbau eines semantischen Webs.