

Universität Leipzig
Softwaretechnik-Praktikum
Sommersemester 2014

Story-Planung

Projekt	Graphical SPARQL Builder
Gruppe	s14.swp.gsb
Verantwortlich	Lukas Eipert
Erstellt am	12. Mai 2014

Inhaltsverzeichnis

Releasebundle R1 zum 22.04.2014	3
/R1/0 Auftrennen der Controllerlogik	3
/R1/1 availableClassesService zum endpointService erweitern	4
/R1/2 Übersetzungsfunktion als eigener Service	4
/R1/3 Drei-Schritte-Layout & Draggable	4
Releasebundle R2 zum 28.04.2014	5
/R2/0 Endpunktanbindung	5
/R2/1 Objekteigenschaften: Inverse Beziehung	5
/R2/2 Datentypeeigenschaft: Zahlenwerte	5
Releasebundle R3 zum 05.05.2014	6
/R3/0 Optionale Eigenschaften	6
/R3/1 Sonderfall: Anonyme Klasse	6
/R3/2 Datentypeeigenschaft: Strings	6
Releasebundle R4 zum 12.05.2014	7
/R4/0 JSON speichern & Laden	7
/R4/1 Legende & Benutzerhandbuch	7
/R4/2 Aggregierte Funktionen	7
Releasebundle R5 zum 19.05.2014	8
/R5/0 Überarbeitung Dokumentation Code-Ebene	8
/R5/1 Testcases	8
/R5/2 Datentypeeigenschaft: Daten & Uhrzeiten	8

Dies ist die erste Version der Storyplanung, diese wird sich wahrscheinlich im Laufe der Zeit zumindest teilweise ändern. Intern arbeiten wir mit Github Issues, welche für das Projekt auf der Github Issues Seite¹ zu finden sind und eher den aktuellen Entwicklungsstand darstellen.

Releasebundle R1 zum 22.04.2014

Christoph steht aufgrund Urlaubs nicht für dieses Releasebündels zur Verfügung. Das erste Releasebundle steht ganz im Licht der Strukturierung des Projekts. Es geht um Umstrukturierung auf der Codeebene, welche die zukünftige Arbeit erleichtert. Des Weiteren wird es größere Layoutänderungen geben, eine Dreiteilung in einen Workspace, in dem der Benutzer seine Anfrage zusammenstellt, einem Querybereich, in dem man die erstellten JSON- & SPARQL-Repräsentationen der Anfrage ansehen kann, sowie einem Resultatbereich, in dem die Ergebnisse der Anfrage angezeigt werden.

/R1/1 Auftrennen der Controllerlogik

Verantwortlich: Nico & Lukas

Aufwandsschätzung: 4 h

Es geht beim Auftrennen der Controllerlogik darum, sich Gedanken zu machen, inwiefern ein Aufbrechen der Bestehenden Controller Sinn ergibt und wie dieses geschehen kann, um die zukünftige Arbeit zu erleichtern. Zu diesem Zeitpunkt scheinen folgende verschachtelte Controller (von Außen nach Innen) sinnvoll:

1. **App:** Kontrolliert die Anwendung, hier werden allgemeine Einstellungen vorgenommen und Dinge wie eine die Übersetzungen von GSBL zu JSON / JSON zu SPARQL in Gang gesetzt
2. **SubjectCollection:** Die Sammlung aller Subjekte, hier wird das hinzufügen und entfernen von Subjekten erledigt.
3. **SubjectInstance:** Ein Individuelles Subjekt, hier werden alle Funktionen vorgenommen, die das einzelne Subjekt betreffen
4. **PropertyCollection:** Analog zur SubjectCollection, bloß für die Eigenschaften
5. **PropertyInstance:** Eine Individuelle Eigenschaft, eventuell noch aufspaltbar (oder zusätzliche Ebene), um Datentypeeigenschaften und Objekteigenschaften zu unterscheiden

¹<https://github.com/swp14-gsb/gsb/issues>

/R1/2 availableClassesService zum endpointService erweitern

Verantwortlich: Siggi & Christoph

Aufwandsschätzung: 2 h

Einen Service Schaffen, der mehrere Funktionen hat, um Metadaten aus dem Endpoint auszulesen, wie zum Beispiel:

- `getAvailableClasses()`
- `getProperties(uri)`
- `getInverseProperties(uri)`

/R1/3 Übersetzungsfunktion als eigener Service

Verantwortlich: Nick & Felix

Aufwandsschätzung: 3 h

Analog zum endpointService sollte ein translationService geschaffen werden, der die Übersetzung zu JSON und zu SPARQL bereitstellt, mit Funktionen wie:

- `translateToJSON(gsbl)`
- `translateToSPARQL(json)`

/R1/4 Drei-Schritte-Layout & Draggable

Verantwortlich: Lukas

Aufwandsschätzung: 2 h

Mergen des bereits existierenden Draggable-Banches in den Master. Des weiteren Umsetzung eines Dreigeteilten Layouts, zum Beispiel durch Tabs, in dem jeder Tab einen logischen Schritt des Workflows des GSB entspricht:

1. Workspace: Erstellung des GSBL-Wortes
2. Queries: Übersetzte JSON & SPARQL-Anfragen
3. Results: Ergebnisse einer SPARQL-Anfrage

Releasebundle R2 zum 28.04.2014

Christoph steht aufgrund Urlaubs nicht für dieses Releasebündels zur Verfügung. Im zweiten Releasebundle wird der Endpunkt angebunden. Desweiteren können Inverse Objekteigenschaften (z.B. „Stadt ist Wohnort von“) hinzugefügt werden und Arithmetik und Vergleiche von Zahleigenschaften ist möglich.

/R2/1 Endpunktanbindung

Verantwortlich: Nico & Felix

Aufwandsschätzung: 4 h

Anbinden eines Endpunkts in den endpointService (durch erstellen passender Konfigurationen) durch geeignete SPARQL-Queries, sodass auf echten Daten eines Endpunkts gesucht werden kann. Vorzugsweise sollte dies an einem ERM Endpunkt² geschehen. Falls dies nicht möglich ist, muss auf die dbpedia³ zurückgegriffen werden.

Achtung: Es sollten auch die Eigenschaften der Überklassen ausgelesen werden (Stichwort: Hierarchien in Subjektklassen).

/R2/2 Objekteigenschaften: Inverse Beziehung

Verantwortlich: Christoph & Siggi

Aufwandsschätzung: 4 h

Umsetzung der Inversen Beziehung (*is of*).

Also zum Beispiel: „Universität *is* Studienort *of* Person“

/R2/3 Datentypeeigenschaft: Zahlenwerte

Verantwortlich: Lukas & Nick

Aufwandsschätzung: 6 h

Arithmetik, Vergleiche und Übersetzung zu SPARQL von *xsd:integer*, *xsd:decimal*, *xsd:float* und *xsd:double*.

²<http://aksw.org/Projects/ERM.html>

³<http://dbpedia.org>

Releasebundle R3 zum 05.05.2014

Im dritten Releasebundle wird die Möglichkeit geschaffen, Eigenschaften als optional zu markieren. Des weiteren wird man Vergleiche auf String-Eigenschaften ausführen können und es wird ein „Anonymes Subjekt“ hinzugefügt, mit dem man nach Dingen Suchen kann, deren Klasse man nicht kennt, die eine Bestimmte Eigenschaft haben.

/R3/1 Optionale Eigenschaften

Verantwortlich: Felix & Sigg

Aufwandsschätzung: 3 h

Möglichkeit schaffen eine Eigenschaft als optional zu deklarieren.

/R3/2 Sonderfall: Anonyme Klasse

Verantwortlich: Nico & Nick

Aufwandsschätzung: 3 h

Eine Subjektklasse, die *owl:Thing* repräsentiert und mit der alle Eigenschaften der Ontologie durchsucht werden können.

/R3/3 Datentypeeigenschaft: Strings

Verantwortlich: Lukas & Christoph

Aufwandsschätzung: 6 h

Vergleiche (equals, contains, starts|endsWith, Regex) und Langparameter von *xsd:string*, sowie Übersetzung zu SPARQL

Releasebundle R4 zum 12.05.2014

Im vierten Releasebundle wird dem Nutzer das Speichern und Laden einer Abfrage ermöglicht. Es wird ein Benutzerhandbuch für die Administratoren und Endnutzer gefertigt, des Weiteren der Website eine Legende hinzugefügt, welche kurze Erklärungen liefert. Es wird auch möglich sein aggregierte Funktionen wie „COUNT, SUM, MIN, MAX, AVG, GROUP _ CONCAT“ auf Eigenschaften auszuführen.

/R4/1 JSON speichern & Laden

Verantwortlich: Nico & Christoph

Aufwandsschätzung: 5 h

Speichern und Laden eines GSBL-Wortes im JSON-Formates.

/R4/2 Legende & Benutzerhandbuch

Verantwortlich: Siggi & Nick

Aufwandsschätzung: 4 h

Erstellung einer Legende in der Oberfläche, Erstellung eines Benutzerhandbuchs für den Endnutzer in HTML. Handbuch für Administratoren im Markdown oder PDF-Format.

/R4/3 Aggregierte Funktionen

Verantwortlich: Lukas & Felix

Aufwandsschätzung: 4 h

Die Aggregierenden SPARQL-Funktionen „COUNT, SUM, MIN, MAX, AVG, GROUP _ CONCAT“ sollen auf Eigenschaften anwendbar sein.

Releasebundle R5 zum 19.05.2014

Beim Finalen Release werden kleinere Pakete in Angriff genommen, um eventuell entstandene Verzögerungen und Bugs zu beheben zu können. Es werden UNIT-Testcases für alle Controller geschaffen, die Dokumentation auf Code-Ebene überarbeitet und die Möglichkeit Daten- & Uhrzeiteigenschaften einzugrenzen.

/R5/1 Überarbeitung Dokumentation Code-Ebene

Verantwortlich: Nick & Christoph

Aufwandsschätzung: 3 h

Komplette Dokumentation auf Codeebene gemäß unseres Qualitätskonzepts.

/R5/2 Testcases

Verantwortlich: Felix & Lukas

Aufwandsschätzung: 4 h

Tests für die bis zum Start des Releasebündel R3 fertigen Klassen. (Controller & Services.)

/R5/3 Datentypeeigenschaft: Daten & Uhrzeiten

Verantwortlich: Nico & Sigg

Aufwandsschätzung: 6 h

Vergleiche: Datums- (*xsd:dateTime*, *xsd:date*) und Zeiträume (*xsd:time*), sowie Übersetzung zu SPARQL.