Aufgabenblatt 2

Projektnummer: swp10-7

Projekttitel: jQuery Plugin - Erweiterte Autovervollständigung auf der Basis von SPARQL Endpunkten

Abgabe: 19.04.2010

Projektleiter: Frederik Baumgardt

Verantwortlicher für die Recherche: Thomas Schöne

Recherchebericht

4. Quellen

1 Begriffe

Im Folgenden werden die wichtigsten Begriffe unseres Projektes erläutert.

1.1 URI

Eine URI ist ein eindeutiger Identifikator für Ressourcen. Dies kann eine einfache Webseite mit einer URL als Identifikator sein, aber auch etwas, das nicht über das Internet erreichbar ist, wie ein Buch mit einer ISBN als URI. Durch diese URI's werden im Semantic Web die Ressourcen ausgezeichnet. Eine URI besteht aus 5 Teilen, wovon „scheme", z.Bsp. „http" und „path", z.Bsp. „www.example.com" vorhanden sein müssen.

1.2 Ressource

Eine Ressource ist ein Objekt, das durch einen eindeutigen Bezeichner, die URI, identifiziert wird. Dabei kann eine Ressource eine einfache Website (ausgezeichnet durch URL) sein, der Autor dieser Website, aber auch nicht im Web erreichbare Dinge, wie ein Buch (ausgezeichnet durch ISBN) oä. Unbenannte Ressourcen dienen zum Gruppieren von benannten Ressourcen.

1.3 Ontologie

Eine Ontologie dient der Wissensrepräsentation, in dem sie die Zusammenhänge und Beziehungen zwischen verschiedenen Objekten beschreibt. Dabei kommen zur Prüfung der Gültigkeit der Daten auch Integritätschecks, oder andere Konzepte wie die Vererbung zur Anwendung. Auf der Grundlage von Ontologien kann das Semantic Web seine Inhalte zur Verfügung stellen.

1.4 Method Chaining

Unter der als Method Chaining, zu deutsch Methodenverkettung, bezeichneten Programmiertechnik gibt jede Methode einer Klasse, die keinen konkreten Rückgabewert hat, das aufgerufene Objekt zurück, sodass man direkt einen weiteren Funktionsaufruf anhängen kann.

1.5 SPARQL

SPARQL ist eine Anfragesprache um spezielle Informationen aus RDF Dokumenten zu extrahieren. Dabei ist der Syntax stark an dem von SQL angelehnt, mit dem bekannten „SELECT– FROM – WHERE" Grundsyntax. Des weiteren gibt es SQLähnliche Elemente wie „ORDER BY". Durch SPARQL ist es möglich die komplexen Zusammenhänge, die durch das Semantic Web gebildet werden, zu durchsuchen. Je nach SPARQL Endpoint, ein Server, der die SPARQL Anfragen bearbeitet, werden die Ergebnisse in Form von XML oder vereinzelt auch JSON geliefert.

1.6 RDF

RDF ist ein Dokumentenformat zur Beschreibung von Informationen, auf dessen Basis das Semantic Web seine Inhalte formalisiert. Dabei besteht ein RDF Dokument aus Tripeln, nach dem Muster „Subjekt – Prädikat – Objekt". Zum Beispiel „Alice – ist verwandt mit – Bob". Jeder Bestandteil bildet eine s.g. Ressource, welche eindeutig durch eine URI ausgezeichnet ist. Zum Beispiel „http://www.example.com/Alice". Desweiteren basiert RDF auf XML, wodurch es möglich ist durch eigens festgelegte Ontologien neue Sprachelemente hinzuzufügen.

1.7 Callback Function (Rückruffunktionen)

Rückruffunktionen sind Funktionen, die anderen Funktionen als Parametern übergeben werden. Diese Art der Programmierung folgt dem Entwurfsmuster der Inversion of Control. Mit Hilfe von Rückruffunktionen lassen sich lose Kopplungen zwischen Komponenten erreichen.

1.8 Content Delivery Network

Ein Content Delivery Network (CDN) ist eine besondere Art des Webhosting, bei der statische Daten auf einem externen Serververbund liegen, der besonders schnell ausliefert. CDN findet Anwendung bei Bildern, Stylesheets und JavaScript.

1.9 Prototypen-basierte Programmierung

Wird auch als klassenlose Programmierung bezeichnet. Objekte werden durch das Clonen anderer Objekte erzeugt. Dabei kann jedes Objekt als Prototyp für neue Objekte verwendet werden. Meist wird nicht speziell zwischen Methoden und Attributen eines Objektes unterschieden.
Alle Objekte bleiben zur Laufzeit strukturell veränderbar.

1.10 Document Objekt Model

DOM (Document Object Model) ist eine abstrakte Spezifikation einer Schnittstelle durch das W3C (Gremium zur Standardisierung der das World Wide Web betreffenden Techniken) auf deren Basis sich die Eigenschaften hierarchisch strukturierter Dokumente unabhängig von konkreten Browserplattformen oder Programmiersprachen beschreiben lassen.

2 Konzepte

2.1 Semantic Web

Das Semantic Web ist eine Spezialisierung des Semantischen Netzes und ist als Erweiterung des WWW zu verstehen. Es soll Computern ermöglicht werden, die Bedeutung von Informationen verwertbar zu machen und so zwischen Daten im Internet Verbindungen bezüglich ihrer Bedeutung herzustellen, was sonst nur für den Mensch möglich ist (was wiederum bei der Informationsflut im Internet auch wieder unmöglich ist). Dies wird durch Annotationen (Metadaten) erreicht, durch die die Bedeutung der Inhalte dazugeschrieben werden.
Für das Semantic Web werden verschiedene Techniken wie OWL, RDF, SPARQL, GRDDL und RDFa verwendet.

2.2 Client - Server - Modell

Das Client-Server Konzept bietet die Möglichkeit, Aufgaben innerhalb eines Netzwerkes zu verteilen. Dabei wartet ein Server-Programm, das bestimmte Dienste anbietet, passiv auf Anfragen von Clients, die es verarbeitet und beantwortet. Die Clients können dann die erhaltenen Daten für die Lösung ihrer eigenen Aufgaben verwenden. Client und Server können sich dabei auf verschiedenen Rechnern oder aber auf dem selben befinden. Die ausgetauschten Daten bei Anfrage (Client) und Antwort (Server) hängen von den angebotenen Diensten ab. Beispiele für die Verwendung von dem Client-Server Konzept: Datenbankserver, Mail-Server, Anwendungsserver

2.3 Javascript

JavaScript ist eine dynamisch typisierte, Prototyp-basierte, objektorientierte Skriptsprache, die hauptsächlich für das DOM-Scripting in WEB-Browsern verwendet wird , sie eröffnet Möglichkeiten zur Benutzerinteraktion und zur Gestaltung dynamischer Webseiten.
In JavaScript existiert keine Klassendefinition im eigentlichen Sinne, Klassen werden immer als Funktion erstellt. Als solche haben sie Eigenschaften und mit ihnen kann interagiert werden, wie mit jedem anderen Objekt auch. Funktionen können auch verschachtelt werden. Wird eine äußere Funktion aufgerufen, so erstellt der JavaScript-Interpreter den Code der inneren Funktion. Dabei entsteht ein Closure.
Funktionen können sowohl als Objekt-Konstruktoren, sowie auch als Methoden benutzt werden.
Doch im Gegensatz zu vielen objektorientierten Sprachen gibt es keinen Unterschied zwischen der Definition einer Funktion und der einer Methode.
Wie die meisten Skriptsprachen besitzt JavaScript eine dynamische Typisierung. So werden Typen Werte und keine Variablen zugeordnet. Später können diese dann durch andere Typen ersetzt werden.
JavaScript-Objekte sind assoziative Arrays ergänzt mit Prototypen. Eigenschaften und deren Werte eines Objektes können während der Laufzeit hinzugefügt, verändert oder gelöscht werden.
JavaScript wird in der Regel in einer Laufzeitumgebung verwendet. So gibt es die Möglichkeit Skripte einzubinden. (HTML <script> Elemente) (Das ist keine Sprachfunktion per se, ist aber in den meisten Implementierungen üblich)
Der primäre Einsatz von JavaScript ist es, Funktionen zu schreiben, die in HTML-Seiten integriert sind. Einige Beispiele sind:

- Das Öffnen eines neuen Fensters mit der Kontrolle über Größe, Position und der Eigenschaften des Fensters
- Validierung der Eingabewerte von Webformularen um zu prüfen ob sie akzeptiert werden, bevor sie an den Server geschickt werden
- Das Verändern von Bilder, wenn sich z.B. die Maus darüber bewegt
- Dynamisches Verändern von WebSeiten mittels DOM
- sofortiges Vorschlagen von Suchbegriffen

Da der JavaScript-Code direkt im Browser des Benutzers läuft, kann auf Benutzeraktion sehr viel schneller reagiert werden
Neben all den positiven Möglichkeiten die JavaScript bietet, bietet es auch eine ganze Reihe Möglichkeiten für eine schädliche Nutzung oder ungewünschter Effekte:

- Verschleiern des Quellcodes
- Deaktivieren des Kontextmenüs und der Kopierfunktion
- Pop-Ups
- Code-Einschleusung

2.4 jQuery

jQuery ist eine beliebte JavaScript-Bibliothek. Die Bibliothek wurde 2006 vorgestellt und bis heute weiterentwickelt. Die Besonderheiten von jQuery sind seine leichte Erlernbarkeit und die besonders schlanke Syntax. Dadurch ist es möglich mit wenig Code bereits sehenswerte Ergebnisse zu erreichen.
jQuery bietet zahlreiche und umfangreiche Funktionen zur Navigation und Manipulation der DOM-Syntax. Zusätzlich existieren Funktionen für animierte Effekte, Ajax und Event-Handling.
Bei der Kompatibilität zwischen jQuery und anderen JavaScript Frameworks (zum Beispiel Prototype) gilt es zu beachten, dass JS Frameworks sich bei der Instanziierung im globalen Namensraum die Variable $ reservieren. Sollte die jQuery-Bibliothek vor einer anderen JS Bibliothek eingebunden werden, kann statt des "$" einfach "jQuery" genutzt werden. Anderweitig muss jQuery angewiesen werden auf die $ Variable zu verzichten (jquery.noconflict()).
Das Konzept von jQuery, um viel Zeit und Code zu sparen, ist Method Chaining. Jede jQuery-Methode gibt das Objekt selbst zurück. Infolgedessen können fortlaufend weitere Methoden angewendet werden.
In jQuery werden Elemente über einen Selektor ausgewählt (Sizzle Selector Engine).Der Selektor ist eine Zeichenkette, die verschiedene Regeln zur Auswahl von Elementen enthält, mit denen gearbeitet werden

soll. Es kann der jQuery-Funktion für die Auswahl auch ein Kontext übergeben werden, der auch eine Selektor-Zeichenkette oder das Ergebnis einer früheren Suche ist.
Ein weiterer Grund jQuery zu verwenden, ist die immense Auswahl an Plugins. Wenn von jQuery-Plugins die Rede ist, wird damit in der Regel eine bestimmte Art und Weise gemeint, in JavaScript Objekte zu erweitern. jQuery-Plugins zeichnen sich dadurch aus, dass sie jQuery-Objekte um Funktionen und Methoden erweitern und damit neue Funktionalitäten bereitstellen. Eine Funktion ist statisch, im jQuery-Objekt gespeichert und besitzen keine besonderen Ausführungskontext. Wiederum werden Methoden im jQuery.fn-Objekt gespeichert und haben immer die aktuelle Auswahl von Elementen im Ausführungskontext. Das bedeutet, dass sie nur auf dem Ergebnis eines Selektors funktionieren. Eine weitere Möglichkeit jQuery-Objekte zu erweitern, ist die Verwendung der Methode .extend . jQuery-Erweiterungen werden in eigenen JavaScript-Dateien bereitgestellt ("jquery.plugin.js"). Zu beachten ist, dass die Erweiterungen nach der jQuery-Bibliothek geladen werden müssen.
Um die Möglichkeit der Erweiterung des eigenen Plugins für andere offen zu halten, sollte die Verwendung von Callback Functions (Rückruffunktionen) Verwendung finden.
Weiterhin stellt jQuery nützliche Hilfsmethoden für die Objekte bereit. Zum Beispiel die .each - Methode. Dadurch kann eine manuelle Iteration über den Inhalt eines jQuery-Objektes durchgeführt werden.

3 Beschreibung der zu studierenden Applikation

3.1 JavaScript, jQuery und jQuery UI

JavaScript ist ein Dialekt des ECMAScript-262-3 Standards der besonders - aber nicht nur - zur Realisierung von dynamischen und interaktiven Websites zum Einsatz kommt. Als solcher ist JavaScript eine dynamisch und schwach typisierte und prototypen-basierte Skriptsprache in der Funktionen wie Objekte behandelt werden. JavaScript-Engines werden heute in allen relevanten Webbrowsern eingesetzt.
jQuery ist ein in JavaScript geschriebenes Framework, daß insbesondere der Navigation in, und Manipulationen auf, der DOM-Syntax von Webseiten dient. Dafür bietet jQuery als Herzstück eine mächtige Selektor-Methode mit vielen Möglichkeiten zur Definition von Filtern auf Mengen von DOM-Elementen an. Desweiteren stehen Entwicklern eine Reihe Funktionen zur Bearbeitung von HTML- und CSS-Elementen zur Verfügung. Auf die API wird unten genauer eingegangen.
jQuery UI ist eine Erweiterung von jQery um Bausteine für Benutzerschnittstellen und fertige Widgets. Im speziellen bietet jQuery UI komplexe Interaktionsmöglichkeiten mit sichtbaren DOM-Elementen und umfangreiche visuelle Effekte.

3.1.1 Entwicklungsumgebung

Zur Unterstützung der Entwicklungstätigkeit werden ein Editor mit Syntax-Highlighting (und evtl. -Korrektur) sowieso die folgenden Browser-integrierter Debugger eingesetzt:
- Firefox / Firebug + Firequery
- IE / integrierter Debugger
- Opera / Dragonfly
- Chrome / Chrome Developer Tools

Ausserdem wird zum Rapid Prototyping die Online-Umgebung auf http://jsbin.com verwendet.

3.2 jQuery im Detail

3.2.1 Die $-Funktion

Mit der jQuery-Funktion können Gruppen von DOM-Elementen gewählt werden. Das können CSS-Tags und -Klassen sein oder HTML-Markup. So gibt beispielsweise $("a") eine Liste aller Anchor-Elemente des HTML-Dokuments zurück.

3.2.2 Die Filter

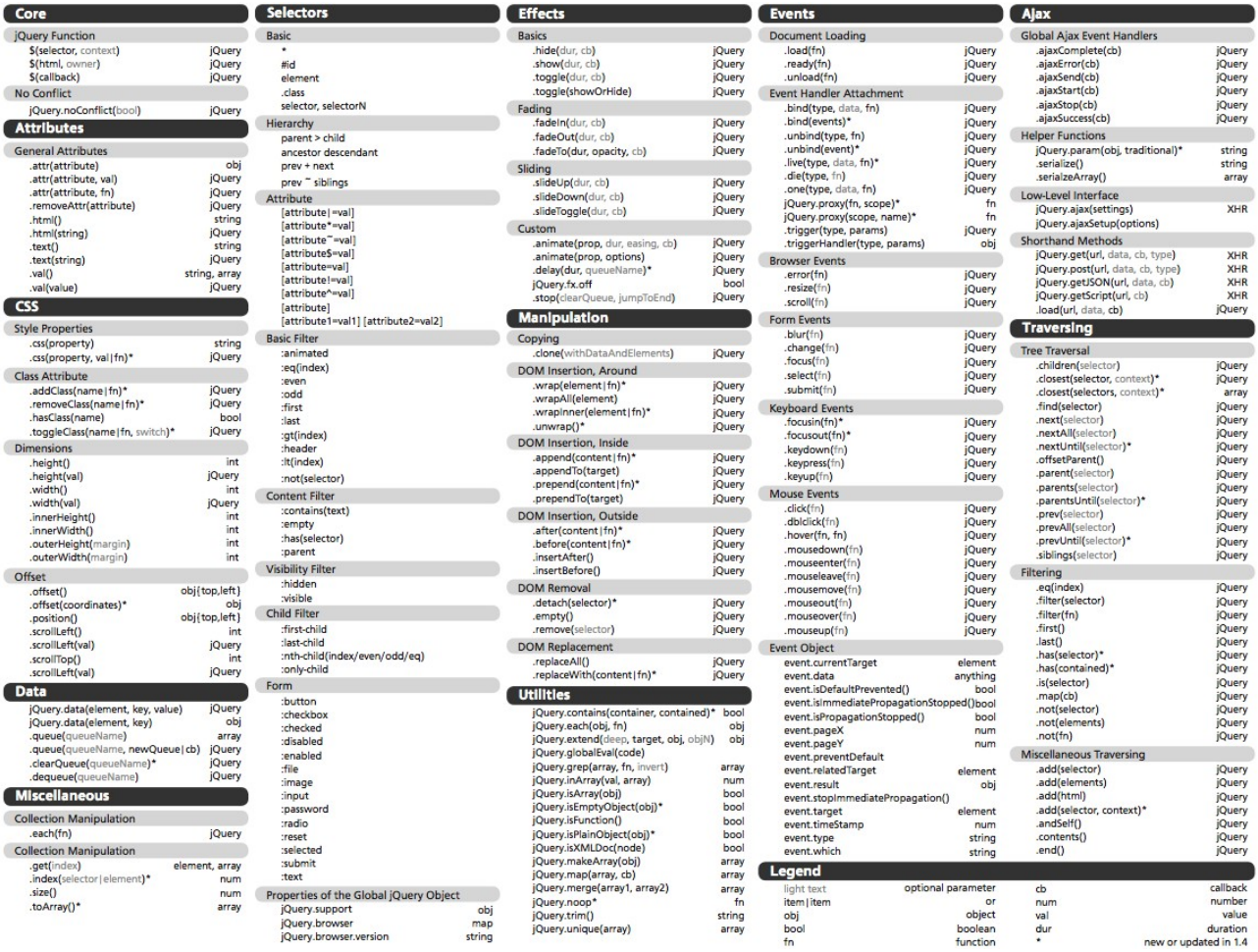
Mit Filtern lassen sich die selektierten Gruppen anhand ihrer Attribute weiter einschränken. Um alle Anchor-Elemente mit der Zeichenfolge OntoWiki im Text zu erhalten ergänzt man den o.g. Selektor folgendermaßen - $("a:contains('OntoWiki')")

3.2.3 Die Manipulatoren

Manipulatoren sind Methoden die auf eine Selektion angewandt werden. Dazu wird der Funktionsaufruf an eine $-Funktion angehängt. Der Aufruf $("a:contains('OntoWiki')").detach() entfernt alle Anchor-Elemente deren Text OntoWiki enthält aus der DOM-Struktur.

3.2.4 Die gesamte API

Die folgende Grafik stellt die jQuery 1.4.2 API übersichtlich dar. Die Parameter der einzelnen Methoden sind unter http://api.jquery.com/browser/ zu finden.



3.3 Ablauf eines Autocompletion-Widgets

Funktionsablauf eines Autocompletion-Widgets (anhand des Beispiels von http://www.pengoworks.com/workshop/jquery/lib/jquery.autocomplete.js)

1. Die Daten auf denen gesucht werden soll werden aus einer vorgegebenen Struktur - Array - geladen
2. Aus dem Array wird eine nach den Anfangsbuchstaben sortierte Liste gebaut, mit Buckets für jeden Buchstaben
3. Dem Textfeld wird ein Event-Listener angehängt, zur Verarbeitung von Eingaben
4. Bei Dateneingabe wird geprüft, ob die Eingabe eine valide alphanumerische Zeichenkette ist, falls ja weiter mit 5. (onChange)
5. In der sortierten Liste werden passende Matches gesucht und zusammengefasst (requestData / receiveData)
6. Falls Resultate vorhanden, werden diese zu DOM (drop-down menü) konvertiert (dataToDom)
7. Die DOM-Struktur wird an das Textfeld gefügt (appendChild)
8. Der DOM-Struktur wird ein Event-Listener angehängt, der bei Klick auf ein Item den Itemtext in das Textfeld einfügt (selectItem)

3.4 pro & cons

Autocomplete 1:

http://bassistance.de/jquery-plugins/jquery-plugin-autocomplete/

PRO:
- Suchtrefferzeilen sind farblich unterschieden (bessere Lesbarkeit)
- Matches werden fett hervorgehoben
- Lizenz MIT/GPL
- läuft in allen gängigen Browsern
- geringe Dateigröße
- auf pluginfix für den IE wird hingewiesen
- einfache Einbindung
- Suchdaten lassen sich in Array speichern oder mittels URL übergeben (zur Suche in einer Datenbank)
- optionsparameter beim Aufruf des Plugins (zum Bsp selectfirst, width der Dropdownbox)
- Bildersuche wird unterstützt
- mehrere suchen in einem Inputfeld möglich (durch Komma getrennt)
- Autofill möglich
- maximal angezeigte Suchtreffer einstellbar

Autocomplete 2:

http://www.pengoworks.com/workshop/jquery/autocomplete.htm

PRO:
- Akzeptiert lokales Datenarray
- maximal angezeigte Suchtreffer einstellbar
- Autofill möglich
- Größe der Dropdownbox einstellbar
- gut dokumentiert
- viele Defaultwerte lassen sich überschreiben (Select First, delay, autoFill)
- extra Optionen möglich (findValue() - für AJAX Operationen, setExtraParams(obj)-Erweiterbar)

CONS:
- keine Bildersuche
- keine Mehrfachsuche in einem Inputfeld (siehe Autocomplete 1 - durch Komma getrennt)

Autocomplete 3:

http://www.ajaxdaddy.com/demo-jquery-autocomplete.html
- basiert auf Autocomplete 2, daher Überschneidungen

PRO:
- Dropdown Resultbox lässt sich durch css bearbeiten
- maximal angezeigte Suchtreffer einstellbar
- autofill möglich

CONS:
- keine Imagesearch
- keine Mehrfachsuche in einem Inputfeld (siehe Autocomplete 1 - durch Komma getrennt)

4 Quellen

http://www.informatik.uni-leipzig.de/~auer/publication/ontowiki.pdf
http://www.informatik.uni-leipzig.de/~auer/publication/WissensarbeitMitOntoWiki.pdf
http://de.wikipedia.org
http://pcai042.informatik.uni-leipzig.de/~swp08-4/download/loesung5.pdf
http://ontowiki.net/Projects/OntoWiki
http://www.glossar.de/glossar/1frame.htm?http
%3A//www.glossar.de/glossar/amglos_d.htm
- http://www.slideshare.net/fabien_gandon/sparql-in-a-nutshell
- http://www.cambridgesemantics.com/2008/09/sparql-by-example/#%281%29
- http://www.slideshare.net/seso81/7-sprachen-des-semantic-web-sparql
- http://www.xml.com/pub/a/2005/11/16/introducing-sparql-querying-semantic-web
tutorial.html
- http://www.w3.org/TR/rdf-sparql-query/
- http://de.wikipedia.org/wiki/Resource_Description_Framework
- http://www.w3.org/RDF/
http://api.jquery.com/, http://docs.jquery.com/How_jQuery_Works,
http://matthiasschuetz.com/tag/jquery/, http://labs.impulsestudios.ca/jquery-cheat-sheet,
http://www.pengoworks.com/workshop/jquery/lib/jquery.autocomplete.js,
http://en.wikipedia.org/wiki/JavaScript