



FACHHOCHSCHULE **TRIER**

Hochschule für Technik, Wirtschaft und Gestaltung
University of Applied Sciences

Informatik

Project Tapir

Firefox Add-on

Jonathan Hoschkara

Abgabe zur Veranstaltung Tool- und Pluginprogrammierung

Betreuer: Prof. Dr. Christof Rezk-Salama

Trier, 4. April 2014

Inhaltsverzeichnis

1	Einleitung	1
2	Installation	2
3	Entwicklung	3
4	Test	5
5	Fazit	6
6	Quellen.....	7

Einleitung

Im Rahmen der Veranstaltung Tool- und Pluginprogrammierung habe ich ein Firefox Add-on entwickelt. Ich habe mich für ein Firefox Add-on entschieden, da ich ein Add-on entwickeln wollte, welches eine Alltagsaufgabe erledigen soll. Und da der Internetbrowser, in meinem Fall Firefox, eines meiner am häufigsten aufgerufenen Programme ist, war es interessant für diesen zu entwickeln.

Die Aufgabe des Add-ons ist das regelmäßige überprüfen einer Webseite auf Updates. Dabei habe ich mich von vornherein nur auf Webseiten beschränkt, die nicht bei jedem Besuch neu generiert werden und eher einen klassischen Aufbau besitzen. Außerdem ist das Add-on dazu gedacht Updates von Webseiten anzuzeigen, die selten aktualisiert werden, und damit Webseiten, die man sonst unregelmäßig aufsuchen würde, da sie vielleicht nur einmal pro Woche aktualisiert werden.

Schwierig war es eine Lösung zu finden, die mit möglichst vielen Webseiten funktioniert. Da der Aufbau und die verwendeten HTML-Elemente, Attribut und Klassennamen von Webseite zu Webseite sehr unterschiedlich sind. Deshalb ist meine Lösung eine eher spezielle Lösung und auf wenige Webseiten zugeschnitten.

Installation

Die Datei `project-tapir.xpi` muss auf einen geöffneten Firefox-Browser gezogen werden, um sie zu installieren. Standardmäßig wird das Add-on-Symbol nach der Installation auf der Add-on-Leiste angezeigt, von der aus das Symbol dann verschoben werden kann. Die Add-on-Leiste kann mit dem Tastaturkürzel „*Strg* + /“ oder über Ansicht → Symbolleisten angezeigt werden. Sie erscheint dann an der Unterseite des Browsers.

Alternativ kann das Add-on SDK heruntergeladen und eingerichtet werden. Mit diesem kann über die Kommandozeilen-Konsole jeweils eine frische Firefox-Instanz zum Installieren und testen, gestartet werden.

Aufgrund eines Bugs im Add-on-SDK kann es dazu kommen, dass bei der Deinstallation des Add-ons die Preferences-Datei nicht gelöscht wird. Dadurch können bei einer erneuten Installation fehlerhafte Einträge angezeigt werden.

Entwicklung

Das Add-on wurde unter Windows 7 und Firefox 27 und 28 entwickelt und getestet. Da mit Firefox 29 einige API-Änderungen bevorstehen, muss das Add-on für diese und nachfolgende Firefox Versionen eventuell angepasst werden, um problemlos zu funktionieren.

Entwickelt wurde das Add-on mithilfe des Firefox Add-on SDKs. Programmiert wurde es in Javascript, und Teile des Add-ons werden mithilfe von HTML/CSS dargestellt. Außerdem wurde die Javascript-Bibliothek jQuery eingebunden, vor allem um Elemente im DOM einfacher zu finden und zu bearbeiten als mit Javascript.

Das Add-on ist folgendermaßen aufgebaut: Ein Widget wird als Icon in der Add-on-Leiste dargestellt. Das Icon des Widgets zeigt auch die Anzahl an neuen Updates als Zahl an. Bei einem Klick mit der linken Maustaste auf das Icon öffnet sich ein Panel über diesem. In diesem Panel werden alle neuen Headlines als link dargestellt. Bei einem Linksklick auf einen link, öffnet sich dieser in einem neuen Tab und der Eintrag wird aus dem Panel entfernt. Bei einem Rechtsklick wird der Eintrag nur aus dem Panel entfernt ohne den link zu öffnen. Der clear-all Button entfernt alle Einträge aus dem Panel. Unter den Add-on Einstellungen kann die zu untersuchende url, sowie das Aktualisierungs-Intervall eingestellt werden.

Der Haupt Add-on Code befindet sich in der Datei main.js im Ordner lib. In main.js werden die einzelnen Komponenten erstellt. Außerdem regelt es die Kommunikation zwischen den einzelnen Komponenten des Add-ons, da eine direkte Kommunikation zwischen ihnen nicht möglich ist. Es kümmert sich um das laden und speichern der persistenten Daten, wie der zuletzt gefundenen Headline und dem Array der ungelesenen Headlines. Es wird die zuletzt gefundene Headline einer Webseite gespeichert und dann periodisch die vorgegebene Webseite geöffnet und überprüft, ob es neue Headlines gibt. Falls es Neue gibt wird die zuletzt gefundene Headline aktualisiert und die ungelesenen Einträge in einem Array gespeichert. Die Komponenten Widget, PageWorker und Panel werden über content-scripts gesteuert, die ihnen übergeben werden. Die Funktion update erstellt bei jedem Aufruf ein PageWorker-Objekt, welches im Hintergrund eine Webseite laden und manipulieren kann. Der PageWorker sucht mithilfe von jQuery, h3-tags die als Kind einen a-tag besitzen. Diese zeichnen eine Headline aus. Ein array aller gefunden

Headlines wird dann an `main.js` zurückgesendet, und dort in Objekten der Klasse `Headline` gespeichert.

Angezeigt werden die ungelesenen Headlines in einem Panel. Das Aussehen des Panels ist in der Datei `headlinePanel.html` gespeichert. Neben der Überschrift befindet sich noch ein leerer `div`-Container mit der `id` „`head`“ auf der Seite. In diesen `div`-Container werden dann über das `content-script` mit Hilfe von `jQuery` die Headlines eingesetzt. Außerdem wird an jede Headline ein „`mousedown`“-Listener angehängt. Bei einem Klick auf eine Headline meldet dieser Listener das geklickte Element an `main.js`. Dort wird abhängig von der Art des Mausklicks die Headline in einem neuen Tab geladen oder nur das Element aus dem Array der ungelesenen Headlines gelöscht. Außerdem wird es innerhalb des Panels ausgeblendet.

Test

Zum Testen des Add-ons ist die Seite „<http://www.allgamesbeta.com/>“ voreingestellt. Da sich diese Seite zu Testzwecken nicht oft genug aktualisiert, habe ich das Add-on mit einer eigenen erstellten Webseite getestet.

Die HTML-Datei „myTestPage.html“ liegt dem Abgabeordner bei. Die Datei kann entweder bei einem Webservice hochgeladen werden oder als lokale Datei bei dem Add-on als url eingetragen werden. Ich habe sie in den public-Ordner meiner dropbox gelegt, und von dieser Datei den public-link in das Add-on eingetragen.

In dieser Webseite sind einige Überschriften voreingestellt. Wenn nun Überschriften über der Headline, die beim Ersten Überprüfen der Seite von dem Add-on gespeichert wurde, hinzukommen, werden diese als ungelesene Headlines gespeichert. Dazu können in der Datei fertige div-Blöcke einkommentiert werden.

Fazit

Das entwickeln von Firefox Add-ons ist sehr interessant und die Tatsache, dass diese mit Hilfe von Webtechnologien wie Javascript und HTML entwickelt werden können ist sehr praktisch. Mein Add-on erfüllt seinen Zweck, könnte aber noch weiter ausgebaut werden. So könnte es z.B. mehrerer Seiten gleichzeitig auf Updates überprüfen. Weitere Probleme die zu beheben wären sind das, auch nur bei einer kleinen Änderung, eine Überschrift als neue Überschrift registriert wird und somit alle Überschriften der Seite zu den ungelesenen hinzufügt.

Quellen

- <http://www.w3schools.com/jquery/default.asp>
- <http://api.jquery.com/>
- <https://developer.mozilla.org/en-US/Add-ons/SDK/Tutorials>
- <https://developer.mozilla.org/en-US/Add-ons/SDK/Tutorials/Installation>