

Entwicklungsprojekt Interaktive Systeme SS2015

von Robert Schumann

MS3 – Proof of Concept

Zunächst habe ich recherchiert. Da node.js als Technologie gewählt wurde habe ich Tutorials zu node.js und den npm nach guten Ansätzen/Erweiterungen durchsucht.

Ich habe nach der Installation der nötigen libraries die Verbindung zu einem Server aufgebaut, sowohl über tcp als auch über die Eingabeaufforderung und http. Dazu wurde code aus den Quellen [1] und [2] benutzt.

Die Recherche im npm hat nach reiflicher Überlegung zu der Entscheidung geführt, node.js doch nicht als Technologie zur Umsetzung zu verwenden, da die Qualität, vor allem der Dokumentationen der dort zur Verfügung gestellten Erweiterungen, aber auch der packages selbst nicht befriedigend ist. Ich hätte die Lösung von Grund auf selbst programmieren müssen und mich gegen diesen Schritt entschieden, da der Aufwand, vor allem in Einarbeitung, Testen und Fehlerbehebung zu groß schien.

Nach einigem Überlegen wurde dann der Plan geändert und die SmackAPI und openfire als neue Umgebung gewählt, wozu vor allem das Durchforsten der alten Projekte angeregt hat. Da ich mit PubSub schon im WBA2-Projekt gearbeitet habe viel die Zweitwahl dann schnell auf diesen Lösungsansatz.

Dann fing der von mir als Arbeitsgerät verwendete Windows-Rechner an andauernd (etwa alle 10 Minuten) einen sogenannten Random Reboot zu machen. Die Versuche diesen Fehler zu beheben haben drei Tagein Anspruch genommen, aber leider nicht zum gewünschten Erfolg geführt. Der Rechner musste als Ressource aufgegeben werden und viel Zeit, die für die Entwicklung eingeplant wurde war vertan. Da die Deutsche Bahn streikte konnte ich nicht auf die Computer der Fachhochschule zurückgreifen. Nach einem weiteren Tag wurde mir dann zum Glück der Rechner einer Freundin zur Verfügung gestellt, der aber mit dem Linux-Ableger Ubuntu als Betriebssystem arbeitet. Die SmackAPI wurde in der Zwischenzeit upgedatet und die benötigten Bibliotheken (.tar.gz) wurden leider von jivesoftware nicht als Archiv zur Verfügung gestellt. [3] Außerdem war die Einarbeitung in ein für mich neues Betriebssystem Zeitaufwendig. Die neue Version der Bibliotheken führte leider dazu, das die Abhängigkeiten vieler Bibliotheken unklar und anders als in der Dokumentation des Projektes beschrieben waren. Seit Smack 4 werden diese Bibliotheken von Drittanbietern nicht mehr mitgeliefert. Es mussten Beispielsweise zur Benutzung der smack.tcp-Pakete Abhängigkeiten installiert werden (xp3 und jxmpp). Sämtliche im Internet verfügbaren Codes zum Thema funktionierten nicht mehr. ([4] und [5])

Nach einer weiteren Recherche, wie mit diesem Problem umzugehen sei versuchte ich Maven zu

nutzen, um die geforderten Abhängigkeiten automatisch erkennen und installieren zu lassen. Doch leider konnte der Rechner Maven nicht erfolgreich ausführen, da der Arbeitsspeicher nicht ausreichte und der Rechner sich stets „aufgehängt“ hat. Nachdem hier wieder viel Zeit verloren gegangen war und nichts zählbares erreicht wurde stand keine Zeit mehr zur Verfügung den Proof-of-Concept zum Ende des MS3 fertig zustellen.

Das bedeutet nun, dass der Workload des dritten Meilensteins zeitlich nach hinten verschoben werden muss, was das Risiko mit sich bringt, dass der folgende Meilenstein auch nicht eingehalten werden kann. Um dem entgegenzuwirken werden mehr Arbeitsstunden pro Tag als eingeplant nötig sein.

Quellenverzeichnis

- [1] [Node.js for beginners, part 1 - Hello world, and a bit of fun!](http://www.theprojectspot.com/tutorial-post/Node-js-for-beginners-part-1-hello-world/2),
<http://www.theprojectspot.com/tutorial-post/Node-js-for-beginners-part-1-hello-world/2>,
29.04.2015
- [2] [Hello Node!](http://howtonode.org/hello-node) - How To Node – NodeJS, <http://howtonode.org/hello-node>, 30.04.2015
- [3] Ignite Realtime: Downloads, <https://www.igniterealtime.org/downloads/index.jsp#smack>,
09.05.2015
- [4] XMPPConnection (Smack 4.1.1 API),
<http://www.igniterealtime.org/builds/smack/docs/latest/javadoc/org/jivesoftware/smack/XMPPConnection.html>, 09.05.2015
- [5] How to Write a simple XMPP (Jabber) client using the Smack API,
<http://www.javaprogrammingforums.com/java-networking-tutorials/551-how-write-simple-xmpp-jabber-client-using-smack-api.html>, 09.05.2015