



Finaler Testbericht

Inhaltsverzeichnis

1 Einführung	2
1.1 Warum Softwaretests?	2
2 Durchgeführte Tests	2
2.1 Test: allgemeine Funktionalität	2
2.1.1 Beschreibung	2
2.1.2 Vorgehen	2
2.1.3 Ergebnis	3
2.2 Test: Autovervollständigung	3
2.2.1 Beschreibung	3
2.2.2 Vorgehen	4
2.2.3 Ergebnis	4
2.3 Test: Validierung der E-Mail-Adresse (update)	5
2.3.1 Beschreibung	5
2.3.2 Vorgehen	5
2.3.3 Ergebnis	5
2.4 Test: Mailversand und Link zur Instanz	5
2.4.1 Beschreibung	5
2.4.2 Vorgehen	6
2.4.3 Ergebnis	6
2.5 Test: Maximal 10 Einladungen pro Tag	6
2.5.1 Beschreibung	6
2.5.2 Vorgehen	6
2.5.3 Ergebnis	6
2.6 Test: Generierung eines eindeutigen Einladungscode	6
2.6.1 Beschreibung	6
2.6.2 Vorgehen	7
2.6.3 Ergebnis	7
2.7 Test: Mehrere Einladungen gleichzeitig	7
2.7.1 Beschreibung	7
2.7.2 Vorgehen	7
2.7.3 Ergebnis	7
3 Resultat	7



1 Einführung

1.1 Warum Softwaretests?

Softwaretests dienen dazu, die Funktionalität und Qualität einer entwickelten bzw. in der Entwicklung begriffenen Software zu überprüfen und letztlich zu gewährleisten. Wichtig ist hierbei, die Tests so früh wie möglich durchzuführen (das bedeutet: bereits einzelne Komponenten testen, lange bevor das Endprodukt fertig gestellt ist), da Fehler in frühen Stadien, in denen die Komponenten noch für sich stehen und einen übersichtlichen Umfang haben, leichter konkret zu bestimmen und günstiger zu beheben sind. Testet man lediglich das Endprodukt, werden zahlreiche Fehler oft gar nicht entdeckt und, selbst wenn dies doch der Fall ist, ist es oft mit einem erheblichen Aufwand verbunden, sie zu beheben.

2 Durchgeführte Tests

Im Laufe der Entwicklung haben wir verschiedene Tests zu verschiedenen Zwischenversionen des Plugins durchgeführt. Dies haben wir in zwei Testberichten erläutert. Das vorliegende Dokument gibt eine Zusammenfassung der bereits beschriebenen Tests, sowie eine Abschlussbetrachtung.

2.1 Test: allgemeine Funktionalität

2.1.1 Beschreibung

Dieser Test soll zeigen, ob unser Plugin korrekt angezeigt wird. Gemäß den Anforderungen soll unser Plugin stets dann zu sehen sein, wenn der User eine Ontologie-Instanz geöffnet hat (und es soll **nicht** dargestellt werden, wenn keine Instanz geöffnet ist, da eine Einladung an dieser Stelle keinen Sinn machen würde). Dabei soll in der OntoWiki ein Fenster mit der Einladungsfunktion angezeigt werden.

2.1.2 Vorgehen

Aufgrund der Trivialität erfolgt dieser Test manuell, also ohne Tools und anderen Testwerkzeugen.

The screenshot displays the OntoWiki interface. On the left, there are three vertical panels: 'OntoWiki (jojojo)' with a search bar, 'Knowledge Bases' with a 'View' button, and 'Navigation: Classes' with a search bar and 'od:Person' listed. The main area is split into two views. The top view, 'Model info', shows 'Testdaten' with sections for 'Comments, Descriptions and Notes' (no comments), 'Properties' (no predicates), and 'Actions' (view all resources, jump to resource). It also includes 'Latest Discussions' and 'Latest Changes' sidebars. The bottom view, 'Properties of Auer, Sören', shows a resource with tabs for 'Properties', 'History', 'Community', and 'Source'. It lists 'rdf:type' as 'od:Person' and 'rdfs:label' as 'Auer, Sören'. Action buttons for 'Edit Properties', 'Clone Resource', and 'Delete Resource' are visible. A right sidebar shows an 'Invitation' form and 'Similar Instances' for 'od:Person'.

Unten sieht man das Plugin bei einer geöffneten Ontologie, oben ist es richtigerweise nicht zu sehen, da keine Instanz geöffnet ist

2.1.3 Ergebnis

Das Plugin wird korrekt angezeigt, und zwar genau dann, wenn der OntoWiki-User eine Instanz geöffnet hat.

2.2 Test: Autovervollständigung

2.2.1 Beschreibung

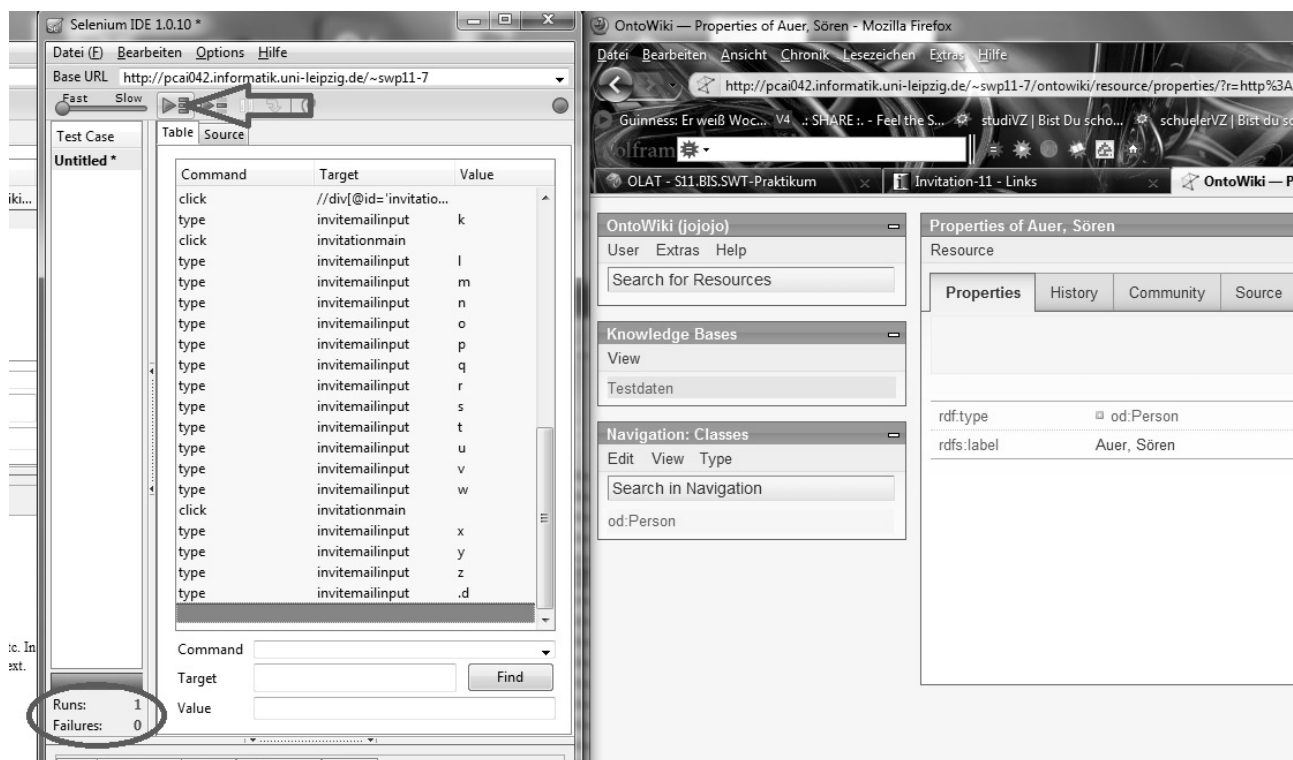
Die Autovervollständigung soll auf Eingaben des Benutzers reagieren und wortvervollständigende Vorschläge anbieten, während der User schreibt. Ein Vorschlag soll nur dann gemacht werden, wenn die angefangene User-Eingabe vollständig im Vorschlag enthalten ist. Umgekehrt sollen keine Vorschläge angeboten werden, die nicht zur Eingabe passen. Natürlich sollten auch keine Vorschläge nicht gemacht bzw. "vergessen werden". Neben dem objektiven Test auf Korrektheit soll auch ein eher subjektiver Test hinsichtlich Usability erfolgen.

Vorgeschlagen werden nur E-Mail-Adressen, die bereits schon einmal (vom selben Benutzer) eingeladen wurden. Die Vorschläge dienen also nur als Eingabehilfe, falls eine Person erneut (zu einer anderen Ontologie) eingeladen werden soll, und nicht als Vorschlagsliste von "fremden Personen".

2.2.2 Vorgehen

Aufgrund vielfältiger Eingabemöglichkeiten, bietet sich für den ersten Testabschnitt ein Framework an, das automatisiert eine große Anzahl an Begriffen "eintippt". Dabei verwenden wir das Firefox-Plugin "Selenium", mit welchem Eingabemuster aufgezeichnet werden und (vielfach wiederholt) abgespielt werden können. Besonderes Augenmerk sollte auf die Wahl der zu vervollständigenden Begriffe gelegt werden: neben "primitiveren" (sprich: eindeutigen) Wörtern, die ein klares Ergebnis erwarten lassen, müssen auch "schwierigere" Fälle untersucht werden (Groß-/Kleinschreibung? Was passiert, wenn auf eine angefangene Eingabe sehr viele Adressen infrage kommen?).

Zudem soll getestet werden, ob das System verzögerungsarm und intuitiv funktioniert. Diese subjektive Komponente erfolgt nicht mit dem Test-Framework, sondern manuell. Als Testbegriffe bieten sich Eingaben an, die eine Vielzahl an Vervollständigungsmöglichkeiten zulassen, da somit erwartungsgemäß die Performance am schlechtesten sein sollte und wir damit den Worst Case analysieren können.



Nachdem man in Selenium eine Aktionsfolge per Record-Button aufgenommen hat, kann man sie - wie im Bild zu sehen - mit einem Mausklick in beliebiger Geschwindigkeit ausführen lassen (siehe Pfeil). Anschließend erhält man eine Übersicht über die aufgetretenen Fehler (siehe Kreis)

2.2.3 Ergebnis

Der Test der Autovervollständigung lieferte immer sinnvolle Vorschläge, die aus der Eingabe herleitbar waren. Es wurden immer alle sinnvollen Vorschläge gemacht (und nichts vergessen).

Die Liste mit Vorschlägen aktualisiert sich sehr schnell und lässt den Benutzer nicht warten.

2.3 Test: Validierung der E-Mail-Adresse (update)

2.3.1 Beschreibung

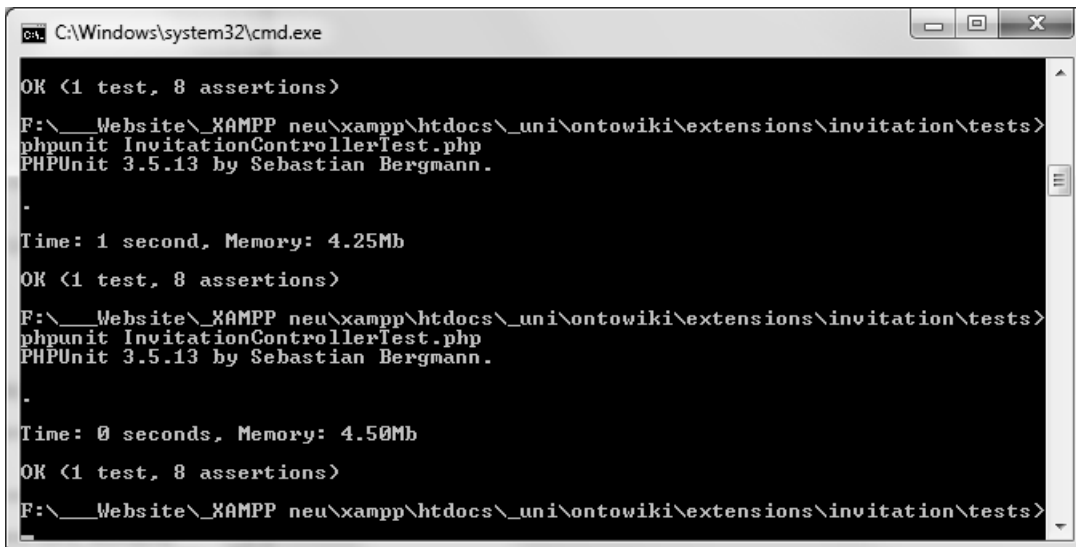
Grundlage für eine sinnvolle Einladung ist ein Empfänger - und damit eine korrekt eingegebene E-Mail-Adresse. Um fehlerhafte Nutzereingaben zu vermeiden, wird das Plugin die E-Mail-Adressen der Empfänger intern überprüfen und bei einem Fehler eine entsprechende Meldung zeigen. Dafür wird eine bereits vorhandene Funktion des Zend-Frameworks verwendet.

2.3.2 Vorgehen

Wir haben sowohl einen phpUnit-Test verwendet, der die Mail-Validierung auf Quellcode-Ebene testet, als auch einen Test mittels Selenium-Plugin.

2.3.3 Ergebnis

Das Prüfen auf Korrektheit der E-Mail-Adresse verlief stets zuverlässig und fehlerfrei. Mails mit unerlaubten Sonderzeichen, fehlendem @, etc. wurden nicht akzeptiert.



```
C:\Windows\system32\cmd.exe

OK <1 test, 8 assertions>
F:\__Website\XAMPP neu\xampp\htdocs\_uni\ontowiki\extensions\invitation\tests>
phpunit InvitationControllerTest.php
PHPUnit 3.5.13 by Sebastian Bergmann.
.
Time: 1 second, Memory: 4.25Mb
OK <1 test, 8 assertions>
F:\__Website\XAMPP neu\xampp\htdocs\_uni\ontowiki\extensions\invitation\tests>
phpunit InvitationControllerTest.php
PHPUnit 3.5.13 by Sebastian Bergmann.
.
Time: 0 seconds, Memory: 4.50Mb
OK <1 test, 8 assertions>
F:\__Website\XAMPP neu\xampp\htdocs\_uni\ontowiki\extensions\invitation\tests>
```

Der phpUnit-Test verläuft fehlerfrei.

2.4 Test: Mailversand und Link zur Instanz

2.4.1 Beschreibung

Ziel des Tests ist das Überprüfen auf zuverlässigen Mailversand, sowie die Kontrolle des verschickten Links. Wichtig ist, dass der Link mit Einladungscode zur Instanz stets korrekt generiert wird.

2.4.2 Vorgehen

Mittels Selenium werden Mails von verschiedenen Instanzen aus an verschiedene E-Mail-Adressen versendet. Anschließend wird manuell geprüft, ob alle Mails ankamen und ob in jeder Mail der Link zur richtigen Ontologie versendet wurde.

2.4.3 Ergebnis

Der Test verläuft erfolgreich. Leider konnte uns seitens der Universität kein Mailversand-fähiger Server zur Verfügung gestellt werden, sodass wir nur auf unseren eigenen PCs testen konnten.

2.5 Test: Maximal 10 Einladungen pro Tag

2.5.1 Beschreibung

Um Spam zu vermeiden, darf ein User (in der Standard-Konfiguration) nur höchstens 10 Einladungen pro Tag versenden. Bei einem Überschreiten der Grenze kommt eine Meldung und die überschüssigen Einladungen werden nicht versendet.

2.5.2 Vorgehen

Der Test erfolgt manuell, indem simplerweise 10 Einladungen (erfolgreich) verschickt wurden.

2.5.3 Ergebnis

Mehr als 10 Einladungen können nicht innerhalb von 24 Stunden verschickt werden.



Meldung, dass keine weitere Einladung verschickt werden darf.

2.6 Test: Generierung eines eindeutigen Einladungscode

2.6.1 Beschreibung

Jede Einladung bekommt einen Einladungscode, der den Eingeladenen zur richtigen Instanz führt. Da dies elementar für unser Plugin ist, muss dieser Code garantiert eindeutig sein, um keine Verwechslungen oder falsche Verweise zu erzeugen.



2.6.2 Vorgehen

Dieser Test wurde manuell durchgeführt. Prinzipiell wird ein 20-stelliger Code zufällig generiert, wobei für jede Stelle über 50 Zeichen zur Auswahl stehen. Dies wird solange wiederholt, bis der Code eindeutig ist (also wahrscheinlich fast immer ohne Wiederholung).

2.6.3 Ergebnis

Manuell konnten wir keine zwei identische Codes erzeugen.

2.7 Test: Mehrere Einladungen gleichzeitig

2.7.1 Beschreibung

Es ist möglich, bis zu fünf Personen gleichzeitig einzuladen. Dafür klickt man nach jedem Eintippen einer Mail-Adresse auf den "+"-Button und hat die Möglichkeit, weitere Adressen einzutippen.

2.7.2 Vorgehen

Dieser Test wurde manuell durchgeführt, indem in zufälliger Abfolge Adressen hinzugefügt und wieder gelöscht wurden.

2.7.3 Ergebnis

Es kam zu keinen Problemen oder Abstürzen. Es wurden stets nur genau die E-Mail-Adressen vom Plugin erfasst, die auch der Nutzer ausgewählt hat.



Von Links nach rechts: Eintippen einer Adresse, nach Bestätigung erscheint diese in der Liste, ein Klick auf das "X" entfernt eine Adresse.

3 Resultat

Das Invitation-Plugin arbeitet sowohl im Allgemeinen als auch im Speziellen fehlerfrei. Alle Funktionalitäten wurden sowohl während der Entwicklung, als auch abschließend im fertigen Plugin, mehrfach überprüft.

Leider waren die Testbedingungen auf dem uns zur Verfügung gestellten Server nicht optimal, da keine E-Mails versendet werden konnten. Dies hat zum einen dazu geführt, dass wir zeitweise das Plugin entwickeln mussten, ohne die korrekte Funktion überprüfen zu können. Weiterhin



mussten wir letzten Endes die Mailfunktion auf einem privaten Server testen.

Trotz dieser Hürde haben uns die Tests geholfen, Fehler und Folgefehler zu minimieren. Entgegen der anfänglichen Planung haben wir jedoch primär manuelle Tests und Selenium-Tests durchgeführt - PHPUnit erschien in den meisten Fällen unnötig.