

Groovy und Scriptrunner Basics

Agenda

- Einführung in Groovy
- Vorstellen der Funktionen im Scriptrunner
- Benutzung der Jira API
- Komplexes Beispiel

Über mich

- Chris Langenberg von Scitotec GmbH
- Entwickler von Webanwendungen und Plugins aller Art
- Zusammenarbeit mit Davaso seit ca. 2 Jahren
- Beratung zu Jira Administration
- Umsetzung von Funktionswünschen in Jira



Scriptrunner Überblick

- Entstanden aus dem Bedürfnis der Funktionserweiterung
- Erstellung von Plugins hat großen Konfigurationsaufwand
- → Erlauben von Ausführung von Skripten
- Einstiegspunkt ist die Jira Administration
- Als Sprache wird Groovy verwendet

Test JIRA

- <https://app-jira.demo.scito.dev>
- admin170 → 123123
- ...
- admin175 → 123123



Groovy

- Skriptsprache auf Java Basis
- Erweitert die Standard Java Syntax
- Alle Beispiele finden sich im Repository im Ordner „1. Groovy“



Groovy

- Schreibe eine kleine Taschenrechner Klasse
- Diese soll 4 Methoden besitzen: Addieren, Subtrahieren, Multiplizieren, Dividieren
- Berechne jeweils einen Wert mit diesen Methoden und gib diesen aus

Scriptrunner Überblick - Konsole

- Ermöglicht direktes Eingeben von Groovy Skripten
- Können auch direkt ausgeführt werden
- Können aber auch Skripte aus dem Script Editor geladen werden



Scriptrunner - Jobs

- Wird verwendet, wenn Skripte periodisch ausgeführt werden sollen
- Skript kann inline eingegeben oder aus Script Editor geladen werden
- Ausführungszeiten mittels Cron-Syntax festgelegt
- <https://www.freeformatter.com/cron-expression-generator-quartz.html>



Scriptrunner - Listeners

- Ausführung von Skripten, wenn bestimmte Dinge im Jira passieren
- Zum Beispiel Erstellen von Vorgängen, bearbeiten von Vorgängen, ...
- Kann auf bestimmte Projekte beschränkt werden

Scriptrunner - Fields

- Berechnete Customfields
- Können nur Werte anzeigen
- Aussehen kann frei definiert werden



Scriptrunner - Behaviours

- Beeinflussung des Verhaltens von Feldern
- Dynamisches Ändern, ob Feld ausgefüllt werden muss oder nicht
- Automatisches Ausfüllen aufgrund von anderen Angaben im Vorgang
- Anzeigen und Ausblenden aufgrund von Bedingungen



Scriptrunner – WF Functions

- Erstellung von Bedingungen, Validatoren und Post functions die in einem Workflow verwendet werden können
- **Bedingungen:** Wann darf ich einen Übergang durchführen
- **Validator:** Sind alle Felder korrekt ausgefüllt?
- **Post Functions:** Funktionen, die nach dem Übergang ausgeführt werden



Scriptrunner – Fragments

- Platzieren von HTML Fragmenten im Jira
- Es gibt verschiedene Typen
- **Web-Item**: Link, der im Jira platziert werden kann
- **Web-Panel**: Zeigen von zusätzliche Informationen im Jira
- **Web-Section**: Schaffung neuer Abschnitte, wo Web-Items eingefügt werden können

Scriptrunner – JQL Functions

- Definition von Suchfunktionen, die in der erweiterten Suche verwendet werden können
- Viele vom Scriptrunner vorgegeben
- Es können aber auch neue Implementiert werden

Scriptrunner – REST Endpoints

- Bereitstellung von Informationen über eine URL
- Erweiterung der Standard REST-API von Jira

Scriptrunner – Resources

- Anlegen neuer Anbindungen in Fremdsysteme
- Datenbanken, LDAP, Slack

Scriptrunner – Mail Handler

- Erlaubt Aktionen beim Empfang von Mails
- Es können zb. neue Vorgänge erstellt werden oder der Inhalt der Mail als Kommentar an einen bestehenden Vorgang angehängen werden



Die JIRA - API

- Die Javadoc findet man unter <https://docs.atlassian.com/software/jira/docs/api/9.11.0/>
- Es muss auf die korrekte Version am Ende der URL geachtet werden
- Anlaufstelle für Fragen: Atlassian Developer Community
- <https://community.developer.atlassian.com/>
- Die Beispiele findet man im Repo im Ordner „3. Jira API“



Festlegen der Priorität anhand einer Matrix

- Anhand einer Matrix soll die Priorität des Vorganges gesetzt werden
- Ausgangspunkt sind die Werte von 2 Custom fields
- Das soll sowohl beim Erstellen als auch beim Ändern des Vorganges passieren

Festlegen der Priorität anhand einer Matrix

		Priorität			
Dringlichkeit	Sofort (sofort zu lösen)	Wichtig (3) Wiederherstellungszeit: 3d (3d x 10h = 30h) Reaktionszeit: 2h	Kritisch (2) Wiederherstellungszeit: 8h Reaktionszeit: 30m	Blocker (1) Wiederherstellungszeit: 4h Reaktionszeit: 15m SLA E-Mail und MIT: 30m	Blocker (1) Wiederherstellungszeit: 4h Reaktionszeit: 15m SLA E-Mail und MIT: 30m
	Hoch (heute zu lösen)	Normal (4) Wiederherstellungszeit: 7d (7d x 10h = 70h) Reaktionszeit: 4h	Wichtig (3) Wiederherstellungszeit: 3d Reaktionszeit: 2h	Kritisch (2) Wiederherstellungszeit: 8h Reaktionszeit: 30m	Blocker (1) Wiederherstellungszeit: 4h Reaktionszeit: 15m SLA E-Mail und MIT: 30m
	Mittel (in den nächsten 7 Tagen zu lösen)	Gering (5) Wiederherstellungszeit: 30d (30d x 10h = 300h) Reaktionszeit: 8h	Normal (4) Wiederherstellungszeit: 7d Reaktionszeit: 4h	Wichtig (3) Wiederherstellungszeit: 3d Reaktionszeit: 2h	Kritisch (2) Wiederherstellungszeit: 8h Reaktionszeit: 30m
	Niedrig (in den nächsten 30 Tagen zu lösen)	Gering (5) Wiederherstellungszeit: 30d Reaktionszeit: 8h	Gering (5) Wiederherstellungszeit: 30d Reaktionszeit: 8h	Normal (4) Wiederherstellungszeit: 7d Reaktionszeit: 4h	Wichtig (3) Wiederherstellungszeit: 3d Reaktionszeit: 2h
		Niedrig	Mittel	Hoch	Sehr Hoch
		Auswirkung			



Festlegen der Priorität anhand einer Matrix

- Erstelle ein Konzept, wie der Sachverhalt umgesetzt werden kann
- Du musst keine konkrete Implementierung erstellen
- Du kannst Pseudo-Code verwenden um Teile der Implementierung anzudeuten



Sicherheitsaspekte

- Man kann Skripte nur als Jira Administrator anlegen und konfigurieren
- Werden aber potentiell durch alle Benutzer ausgeführt
- Um die korrekten Zugriffsrechte zu gewährleisten, sollten alle Funktionen immer als der aktuelle Nutzer ausgeführt werden (wenn möglich)



Sicherheitsaspekte

- Es muss immer darauf geachtet werden, dass die Browseransicht nicht kaputt gemacht wird
- Das ist besonders wichtig bei Fragmenten
- Es ist theoretisch möglich, die komplette Datenbank zu löschen, daher Vorsicht!