

Java aktuell

Das Magazin der Java-Community

Die „Bohne“ in voller Blüte

Besser mit Java entwickeln

Java EE 7 für die Cloud, Seite 14

Apache Camel Security,
Seite 30

Eclipse Code Recommenders,
Seite 44

Tools für Web-Applikations-
Lasttests, Seite 52

Java aktuell Update

Fortschritte bei JAX-RS 2.0,
Seite 9

Neue Features in Java FX 2.1,
Seite 18



iJUG
Verbund

Sonderdruck

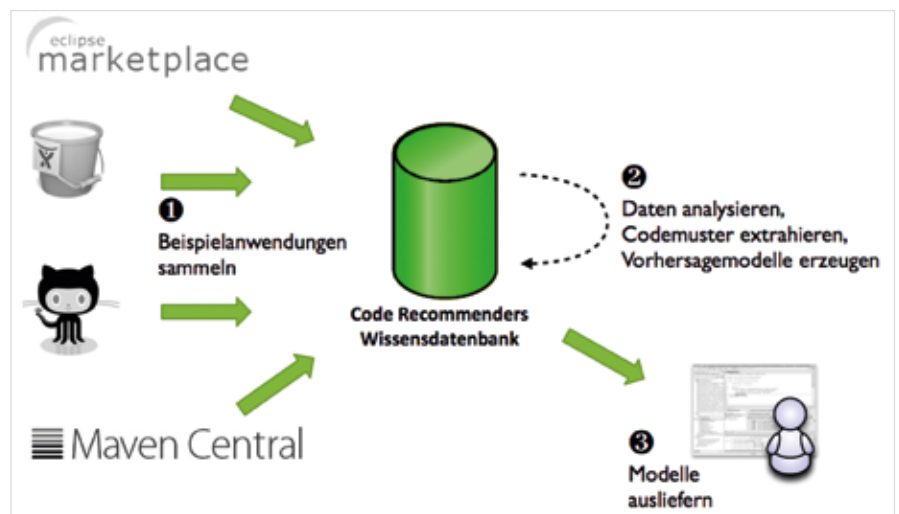


- 3 Editorial
Wolfgang Taschner
- 5 Das Java-Tagebuch
Andreas Badelt
- 9 Fortschritte bei JAX-RS 2.0:
Neuer Standard für REST-basierte
Web-Services für Java
Peter Doschkinow
- 14 Java EE 7: Entwickeln für die Wolken
Markus Eisele
- 17 Inserentenverzeichnis
- 18 Mit JavaFX entwickeln
Wolfgang Weigend
- 23 Integration à la Apache
Tobias Unger und Jochen Traunecker
- 30 Apache Camel Security –
Route Security
Dominik Schadow
- 36 „Ich hasse Kaffee, aber ich mag Java
und Mac OS X ...“, Interview mit Scott
Kovatch, Mac OS X Projektleiter für
OpenJDK
- 38 Impressum
- 39 Windows Azure Reloaded – neue
Möglichkeiten für Java-Entwickler
Holger Sirtl
- 44 Eclipse Code Recommenders:
Liest du noch [Quellcode] oder
programmierst du schon?
Marcel Bruch und Andreas Sewe
- 48 Webservice-Lasttests mit loadUI
Sebastian Steiner
- 52 Tools für Web-Applikations-Lasttests
Michael Jerger

- 57 Unbekannte Kostbarkeiten des SDK
Heute: JavaScript
Bernd Müller
- 59 SAP NetWeaver Cloud:
Eine OpenSource-basierte Java-
Cloud-Plattform
Matthias Steiner
- 62 Java Forum Stuttgart –
ein voller Erfolg
Stefanie Nolda
- 63 Cross-Origin-Resource-Sharing
Dr. Fabian Stäber
- 65 „Java ist so etwas wie ein guter
Kollege, auf den ich mich (zumeist)
verlassen kann ...“, Interview mit Uwe
Sauerbrei, Vorsitzender der Java User
Group (JUG) Ostfalen



Mit Java EE 7 für die Cloud entwickeln, Seite 14



Eclipse Code Recommenders: eine Erweiterung der Java-Development-Tools, Seite 44

Dies ist ein Sonderdruck aus der Java aktuell. Er enthält einen ausgewählten Artikel aus der Ausgabe 04/2012. Das Veröffentlichen des PDFs bzw. die Verteilung eines Ausdrucks davon ist lizenzfrei erlaubt.

Weitere Informationen unter www.ijug.eu

Unbekannte Kostbarkeiten des SDK

Heute: JavaScript

Bernd Müller, Ostfalia, Hochschule für angewandte Wissenschaften

Das Java SDK enthält eine Reihe von Features, die wenig bekannt sind. Wären sie bekannt und würden sie verwendet, könnten Entwickler viel Arbeit und manchmal sogar zusätzliche Frameworks einsparen. Wir stellen in dieser Reihe solche Features des SDK vor: die unbekannten Kostbarkeiten.

Das Java-Scripting-API, definiert als JSR 223 (Scripting for the Java Platform [1]), hatte ursprünglich die Integration von Skript-Sprachen zur Erzeugung von Web-Seiten zum Ziel. Zum Zeitpunkt der Initiierung des JSR im Jahr 2003 waren dies vor allem PHP und Perl. Wenn man heute von „Scripting“ im Zusammenhang mit Java spricht, werden häufig andere Sprachen, etwa Groovy oder JPython, eventuell auch Scala genannt. Da diese Sprachen sehr gute Schnittstellen von und zu Java besitzen, wird in der Regel das Scripting-API für diese Sprachen selten verwendet. Für PHP und Perl kommt dieses zumindest nach unserer Kenntnis ebenfalls selten vor, es gehört also zu den unbekannten Kostbarkeiten des SDK. Der Artikel stellt das Scripting-API kurz vor und geht insbesondere auf die Verwendung der im SDK enthaltenen JavaScript-Implementierung ein, da diese durchaus nutzbringend eingesetzt werden kann.

Java und JavaScript

Das im JSR 223 [1] definierte Scripting-API ist seit Java 6 Bestandteil des SDK und im Package „javax.script“ durch insgesamt 12 Interfaces, Klassen und Exceptions repräsentiert. Neben dieser allgemeinen Schnittstellen-Definition für verschiedene Skript-Sprachen ist jedoch auch eine Implementierung einer Skript-Sprache im SDK 6 enthalten. Dies ist die JavaScript-Implementierung der Mozilla Foundation mit Code-Namen „Rhino“ [2]. Sie implementiert JavaScript nach der Spezifikation ECMA-262 [3]. Rhino unterliegt, wie für die Mozilla Foundation üblich, einer Open-Source-Lizenz und ist vollständig in Java implementiert.

Das Scripting-API

Die Klasse „ScriptEngineManager“ realisiert den Einstiegspunkt zur Ermittlung al-

ler bekannten Script-Implementierungen. Sie basiert auf dem Konzept eines Service-Loaders, das wir in Java aktuell Q4/2011 im Rahmen dieser Reihe vorgestellt haben [4]. Listing 1 zeigt den Zugriff auf alle bekannten Script-Implementierungen und gibt einige Details dieser Implementierungen aus.

Die im Listing 2 dargestellte Ausgabe erscheint bei der Ausführung des Programms aus Listing 1 auf dem System des Autors (JDK 7u5), wenn Groovy im Klassenpfad ist. Wie man sieht, registriert sich auch Groovy über den Service-Loader.

Zu betonen ist hier nochmals, dass die JavaScript-Implementierung seit Version 6 im JDK enthalten ist und nicht installiert oder in den Klassenpfad aufgenommen werden muss.

Die Verwendung von JavaScript

Soll JavaScript verwendet werden, ohne nach allen Script-Implementierungen zu suchen, so erlaubt der „ScriptEngineManager“ den direkten Zugriff auf eine Implementierung über den Namen (getEngineByName()), die Dateinamenserweiterung (getEngineByExtension()) oder den

```
ScriptEngineManager manager = new ScriptEngineManager();
List<ScriptEngineFactory> factoryList =
    manager.getEngineFactories();
for (ScriptEngineFactory factory : factoryList) {
    System.out.println(factory.getEngineName()
        + " " + factory.getEngineVersion()
        + " / " + factory.getLanguageName()
        + " " + factory.getLanguageVersion());
}
```

Listing 1

```
Mozilla Rhino 1.7 release 3 PRERELEASE / ECMAScript 1.8
Groovy Scripting Engine 2.0 / Groovy 1.7.2
```

Listing 2

```
ScriptEngine engine = manager.getEngineByName ("JavaScript");
String userInput = "2 + 3 * 4";
engine.eval ("print (" + userInput + ")");
```

Listing 3

```
ScriptEngine engine = manager.getEngineByName ("JavaScript");
engine.put("x", 2);
engine.put("y", 3);
engine.put("z", 4);
engine.eval (new FileReader("./formula.js"));
Double result = (Double) engine.get("result");
```

Listing 4

```
// Datei „formula.js“
// kompliziertere Formel fuer „result“
result = Math.pow(x, y) * Math.sqrt(z);
```

Listing 5

Mime-Type (`getEngineByMimeType()`). Als kleines Beispiel definieren wir die Anforderung, dass der Benutzer einen einfachen arithmetischen Ausdruck eingeben können soll, der dann ausgewertet wird. Wir wollen damit einen einfachen Taschenrechner realisieren.

In Java EE kann man dafür auf die Expression Language zurückgreifen. In Java SE muss der Eingabe-String (etwa ein Textfeld in Swing, SWT oder JavaFX) zunächst in seine Tokens zerlegt, der Syntax-Baum aufgebaut und anschließend ausgewertet werden. Bei JavaScript wertet der Code in Listing 3 eine fiktive Benutzereingabe aus und gibt sie nach „System.out“ aus.

Mehrere Benutzereingaben und deren Kombination in einem komplexeren Ausdruck lassen sich ebenfalls leicht realisieren. In Listing 4 werden fiktive Benutzereingaben über ein Language-Binding an die JavaScript-Variablen x, y und z gebunden. Die „eval()“-Methode wertet die JavaScript-Anweisung in der Datei „formula.js“ aus und stellt das Ergebnis über die „get()“-Methode wieder Java zur Verfügung. Listing 5 zeigt die Datei „formula.js“.

Die verwendete „put()“-Methode bindet die Variablen global für die „ScriptEngine“. Das Interface „Bindings“ erlaubt es, eine Menge von Variablen-Bindungen zusammenzufassen und als weiteren Para-

meter der „eval()“-Methode zu übergeben. Das Interface „ScriptContext“ gestattet die Definition von Kontexten zur Skript-Ausführung. Beispiele beider Konstrukte liegen aber außerhalb des einführenden Charakters dieses Artikels.

Fazit

Die verborgenen Kostbarkeiten „Scripting-API“ und „JavaScript“ sind seit Java 6 im SDK enthalten. Exemplarisch haben wir gezeigt, wie damit einfache, aber auch kompliziertere Arithmetik möglich ist, ohne einen eigenen Interpreter für derartige arithmetische Ausdrücke implementieren zu müssen.

Literatur/Web

- [1] <http://jcp.org/en/jsr/detail?id=223>
- [2] <http://www.mozilla.org/rhino>
- [3] <http://www.ecma-international.org/publications/standards/Ecma-262.htm>
- [4] Bernd Müller, Unbekannte Kostbarkeiten des SDK: Der Service-Loader. Java aktuell Q4/2011

Bernd Müller

bernd.mueller@ostfalia.de

Bernd Müller ist Professor für Software-Technik an der Ostfalia. Er ist Autor des Buches „Java-Server Faces 2.0“ und Mitglied in der Expertengruppe des JSR 344 (JSF 2.2).



Der Bedeutungszuwachs des Internets als primärer Zugangskanal für Informationen stellt Web-Entwickler täglich vor neue Aufgaben und Herausforderungen. Abseits von ständig neuen Technologien gehören auch Fragen des Projektmanagements und des Betriebs von Web-Anwendungen heute zu seinem Grundwissen. Die Web Developer Conference vom 17. Bis 18. September 2012 in Hamburg greift dies auf und vermittelt aktuellstes Wissen zur professionellen Entwicklung von Web-Anwendungen.

Die Veranstaltung richtet sich an Software-Entwickler, Content- und Online-Manager, Agenturen und Webmaster. Auch Projektleiter können sich Inspiration für die Durchführung von großen Web-Projekten holen. In vielen Sessions erfahren die Teilnehmer, wie sie heute Web-Anwendungen entwickeln, die den Herausforderungen der Gegenwart und der Zukunft gewachsen sind. Zwei Workshops zu den Themen „Node.js & Co“ und „HTML5“ für Web-Apps ergänzen das Veranstaltungsprogramm am 19. September 2012.

Aktuelle Informationen und Anmeldung unter www.web-developer-conference.de. Sonderkonditionen gibt es bei Eingabe des Codes „WDC12ijug“.



www.ijug.eu

**JETZT
ABO
BESTELLEN**

Sichern Sie sich 4 Ausgaben für 18 EUR

Für Oracle-Anwender und Interessierte gibt es das Java aktuell Abonnement auch mit zusätzlich sechs Ausgaben im Jahr der Fachzeitschrift DOAG News und vier Ausgaben im Jahr Business News zusammen für 70 EUR. Weitere Informationen unter www.doag.org/shop/

FAXEN SIE DAS AUSGEFÜLLTE FORMULAR AN

0700 11 36 24 39

ODER BESTELLEN SIE ONLINE

go.ijug.eu/go/abo



Interessenverbund der Java User Groups e.V.
Tempelhofer Weg 64
12347 Berlin

Java aktuell

+++ AUSFÜLLEN +++ AUSSCHNEIDEN +++ ABSCHICKEN +++ AUSFÜLLEN +++ AUSSCHNEIDEN +++ ABSCHICKEN +++ AUSFÜLLEN

☐ **Ja**, ich bestelle das Abo Java aktuell – das iJUG-Magazin: 4 Ausgaben zu 18 EUR/Jahr

☐ **Ja**, ich bestelle den kostenfreien Newsletter: Java aktuell – der iJUG-Newsletter

ANSCHRIFT

Name, Vorname

Firma

Abteilung

Straße, Hausnummer

PLZ, Ort

GGF. RECHNUNGSANSCHRIFT

Straße, Hausnummer

PLZ, Ort

E-Mail

Telefonnummer



Die allgemeinen Geschäftsbedingungen* erkenne ich an, Datum, Unterschrift

*Allgemeine Geschäftsbedingungen:

Zum Preis von 18 Euro (inkl. MwSt.) pro Kalenderjahr erhalten Sie vier Ausgaben der Zeitschrift "Java aktuell – das iJUG-Magazin" direkt nach Erscheinen per Post zugeschickt. Die Abonnementgebühr wird jeweils im Januar für ein Jahr fällig. Sie erhalten eine entsprechende Rechnung. Abonnementverträge, die während eines Jahres beginnen, werden mit 4,90 Euro (inkl. MwSt.) je volles Quartal berechnet. Das Abonnement verlängert sich automatisch um ein weiteres Jahr, wenn es nicht bis zum 31. Oktober eines Jahres schriftlich gekündigt wird. Die Widerrufsfrist beträgt 14 Tage ab Vertragserklärung in Textform ohne Angabe von Gründen.

