TecFeeds – Hochkonzentriertes Wissen zu aktuellen Technologien

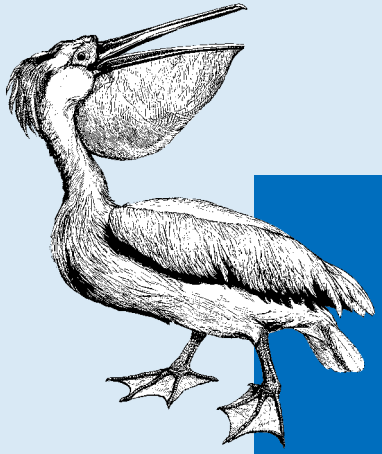
Sie sind Webentwickler, Programmierer oder Systemadministrator? Sie suchen intelligente Lösungen und Informationen zu aktuellen Technologien?

O'Reilly TecFeeds liefern konzentriertes Know-how im praktischen PDF-Format zum sofortigen Download: stets verfügbar, wann und wo interessierte IT-Profis es benötigen. TecFeeds bringen Trendthemen auf den Punkt – kompakt, praxisorientiert und in der gewohnten O'Reilly-Qualität. Einfach herunterladen, anwenden und ein Stück weiter sein.

Sie suchen kompakte Informationen zu anderen aktuellen IT-Themen? Senden Sie uns Ihren Themenwunsch per Mail an proposals@oreilly.de.

O'REILLY®

INHALT



TecFeeds

www.tecfeeds.de

Tipps zur Benutzung

Um die optimale Lesbarkeit und Funktionalität der TecFeeds sicherzustellen, empfehlen wir den Reader ab Version 7. Die aktuelle Version des Acrobat Readers können Sie kostenlos unter *http://www.adobe.de/products/acrobat/readstep.html* herunterladen.

Die TecFeeds sind durch das Querformat, Schriftart und Schriftgröße für das Lesen am Bildschirm optimiert. Selbstverständlich sind Links aktiv. Über das Inhaltsverzeichnis können Sie so gezielt Überschriften anspringen und externe URLs aufrufen (sofern Sie gerade Zugang zum Internet haben).

Code-Beispiele können Sie per copy & paste für eigene Projekte direkt übernehmen. Die TecFeeds können beliebig oft ausgedruckt werden. Um einen ansprechenden Ausdruck der gesamten Seite zu erhalten, wählen Sie in Ihrem Reader unter *Datei / Drucken / Seiteneinstellungen* unter dem Punkt *Seitenanpassung* die Einstellung *An Druckränder anpassen*. Unter *Drucken / Eigenschaften / Größenänderungsoptionen* behalten Sie die Voreinstellung von *100%* bei.

Beim Kauf eines TecFeeds wird für Sie ein persönliches Exemplar erstellt, das Ihren Namen enthält. Bitte beachten Sie den Copyright-Vermerk.

Hinweise zum Kauf

Die TecFeeds können direkt auf unserer Webseite gekauft und heruntergeladen werden. Dazu arbeiten wir mit dem Bezahlservice ClickandBuy zusammen, der eine sichere Abrechnung über Kreditkarte oder Lastschrift ermöglicht. Falls Sie noch kein ClickandBuy-Kunde sind, müssen Sie sich zunächst unter *www.clickandbuy.com* registrieren. Unter Angabe Ihrer Nutzerdaten können Sie das gewünschte TecFeed dann auf unserer Webseite kaufen. Weitere Informationen erhalten Sie unter *www.clickandbuy.com*.

Copyright-Vermerk

Leseproben

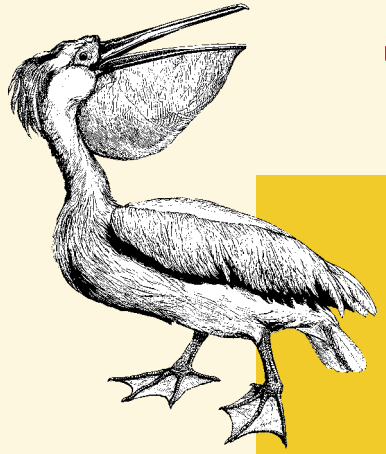
Im folgenden finden Sie Leseproben der ersten sechs TecFeeds. Künftig ist das Erscheinen von jeweils zwei bis drei neuen TecFeeds pro Monat geplant.

Google Web Toolkit für Ajax

Bruce W. Perry
Übersetzung von Sascha Kersken

Das Google Web Toolkit (GWT) ist ein kompaktes Framework, das Java-Programmierer zum Erstellen von Ajax-Anwendungen einsetzen können. GWT ermöglicht Ihnen in Ihrer bevorzugten IDE, etwa in IntelliJ IDEA oder Eclipse, das Erstellen einer Ajax-Anwendung mit Hilfe ähnlicher Paradigmen und Mechanismen wie beim Programmieren einer Java-Swing-Anwendung. Nachdem Sie die Anwendung in Java programmiert haben, erzeugen die GWT-Tools den JavaScript-Code, den die Anwendung benötigt. Sie können auch typische Java-Projekt-Tools wie JUnit und Ant verwenden, wenn Sie GWT-Anwendungen erzeugen. GWT steht kostenlos zum Download bereit, und Sie können den client- und den serverseitigen Code, den Sie mit Hilfe des Frameworks erstellen, frei verbreiten. Dieses TecFeed erläutert, wie Sie die Arbeit mit GWT beginnen, und demonstriert dann die Erzeugung einer einfachen Ajax-Anwendung.

INHALT



O'REILLY®

TecFeeds

www.tecfeeds.de

Der Begriff Ajax ist allgegenwärtig in der Softwarewelt. Das Akronym stand ursprünglich für »Asynchronous JavaScript and XML«, bezieht sich aber inzwischen auf eine größere Auswahl von Verfahren, von denen einige beispielsweise ohne XML auskommen.

Wenn man vom Begriff absieht, steht Ajax für eine Softwarearchitektur, die so gestaltet wurde, dass sie in jedem Webbrowser läuft, aber eine Benutzeroberfläche mit dem reaktionsschnellen Verhalten einer Desktop-Anwendung aufweist. Das bedeutet zum Beispiel, dass eine Gitternetzkomponente wie etwa eine Kalkulationstabelle, die auf der Webseite angezeigt wird, unmittelbar auf die Manipulation der Daten durch den User reagiert, ohne Zeitverzögerung durch das Aktualisieren oder den Neuaufbau der Seite mit Hilfe neuer HTTP-Anfragen. Googles Gmail und Kalender und Yahoo! Maps sind drei Beispiele für typische Ajax-Anwendungen.

Ajax funktioniert durch Kommunikation mit der mittleren und/oder der Datenbankschicht der Anwendung über ein JavaScript-Objekt. Dieses Objekt heißt XMLHttpRequest (XHR). Die Webseite wird mit Hilfe üblicher Standardtechnologien aufgebaut und programmiert – HTML/XHTML, Cascading Style Sheets (CSS), JavaScript und der Document Object Model-API, die der Struktur jeder Webseite zu Grunde liegt. Die Daten, die die Seite mit den Serverschichten – etwa mit einer Produktdatenbank – austauscht, können reiner Text, XML oder ein Format namens JavaScript Object Notation (JSON) sein.

Der Benutzer könnte beispielsweise einen Produktnamen aus einem Auswahllisten-Steuerelement auf einer Webseite auswählen. Als Reaktion auf dieses Ereignis verwendet die Anwendung XHR, um eine Anfrage nach Daten über dieses Produkt zu versenden. Die Seite wird nie gewechselt, weil die Anwendung die Anfrage asynchron hinter den Kulissen sendet. Die Server-Komponente, etwa eine PHP-Datei oder ein Servlet, empfängt die HTTP-Anfrage und sendet eine Antwort mit Informationen über das Produkt zurück, beispielsweise im XML- oder JSON-Format. Die Anwendung verarbeitet die Antwort mit Hilfe der JavaScript-DOM-API oder durch Konvertieren des JSON-Werts in ein clientseitiges Objekt (siehe weiter unten in diesem TecFeed). Der Benutzer sieht die neuen Informationen auf der Webseite, hoffentlich ohne längere Verzögerung.

Der Ajax-Ansatz des Google Web Toolkits

Ein Entwickler erstellt seine Ajax-Anwendung typischerweise, indem er XHTML-Seiten und JavaScript-Code mit Hilfe seiner bevorzugten integrierten Entwicklungsumgebung (IDE) oder sogar einem Texteditor anfertigt. Es gibt eine Auswahl verschiedener Bibliotheken und Frameworks, die Programmierern vorgefertigte JavaScript-Klassen zur zeitsparenden Implementierung dynamischer Verhaltensweisen wie Drag and Drop oder komplexe visuelle Baumstrukturen zur Verfügung stellen. Zu diesen Bibliotheken gehören das Dojo-Toolkit, die Yahoo! User Interface Library und script.aculo.us. Sie sind für Entwickler mit recht fortgeschrittenen JavaScript-Kenntnissen gedacht.

Diese einfachen Methoden beginnen sich zu ändern, weil sich leistungsfähige Tools für Ajax-Entwickler verbreiten. GWT bietet einen anderen Ansatz für Ajax als die meisten Toolkits.

Über das GWT-Framework können Sie Ihre Benutzeroberfläche nur mit Hilfe der Sprache Java gestalten und programmieren. Sie können die GWT-Kommandozeilentools verwenden, um die Syntax der Java-Klassen zu prüfen, und dann automatisch das JavaScript für die Clientseite der Anwendung erzeugen. Das Gestalten der Benutzeroberfläche erinnert stark an den Einsatz der Java-API Swing.

Sie können das GWT daher als JavaScript-Generator für Java-Programmierer sowie als Framework zur Erstellung wiederverwendbarer oder erweiterbarer Benutzeroberflächen-Elemente betrachten. Sie brauchen nicht unbedingt JavaScript-Kenntnisse, können aber mit Hilfe spezieller Programmierkonstrukte, die das GWT bereitstellt, auf Wunsch auch eigenes JavaScript in Ihren Code einfügen.

Einige Vorteile des GWT im Überblick:

- Sie können Ihre bevorzugten Tools und Ihr vorhandenes Wissen nutzen, um komplexe Ajax-Anwendungen zu erstellen, oft sogar ohne sich mit JavaScript zu beschäftigen.
- Sie können Ihre Anwendung in rein objektorientierter Weise gestalten und entwickeln, da Sie Java verwenden. JavaScript besitzt nicht alle in Java verfügbaren OO-Features.
- Der Einsatz des GWT befreit Sie von vielen der Probleme, die bei der Anpassung von Ajax-Code für all die unterschiedlichen Browser vorkommen. Diese Probleme können ein echtes Ärgernis für Entwickler darstellen, wenn zwei Browser dasselbe Stück Code vollkommen unterschiedlich behandeln. Sie brauchen im Allgemeinen nicht über diese potenziellen Inkompatibilitäten nachzudenken, während Sie Ihre Anwendung erstellen, sondern erst dann, wenn Sie sie nachher in den diversen Browsern testen.
- Das JavaScript, das der GWT-Compiler erzeugt, ist recht verworren bzw. sehr schwierig zu lesen, was ein geringes Maß an Sicherheit oder Schutz Ihres geistigen Eigentums zu Ihrer Anwendung hinzufügt. Sie können diese Verwirrungsfunktion optional »ausschalten«, was weiter unten in diesem TecFeed beschrieben wird.
- Google und einige Drittanbieter fügen nach und nach Bibliotheken zum GWT hinzu oder machen bestehende verfügbar, die mit dem GWT zusammenarbeiten können. Diese Aktivitäten erweitern Features und Einsatzgebiete des GWT ständig. Ein Beispiel ist eine Open Source-Bibliothek namens GWT Widget Library. Diese JAR-Datei enthält Klassen, die als Wrapper für die Scriptaculous-Effekt-Objekte dienen und es Ihnen ermöglichen, diese interessanten, dynamischen visuellen Effekte zu Ihrer GWT-Anwendung hinzuzufügen. Mehr über diese Bibliothek können Sie unter *http://gwt-widget.sourceforge.net* erfahren.

Okay, Schluss mit dem allgemeinen Geplänkel. Schreiben wir ein Programm und schauen wir uns etwas Code an!

Loslegen

Zunächst sind ein paar Voraussetzungen zu erfüllen. Sie brauchen ein installiertes Java Development Kit (JDK) in Version 1.4 oder neuer. Zum Zeitpunkt des Schreibens dieses TecFeeds kann man unter Verwendung der GWT-Distribution *gwt-mac-1.2.22* keine Java 5-Features wie Generics oder Autoboxing für den clientseitigen Code verwenden, aber Sie können natürlich den Java 5 JDK-Compiler nutzen, wenn Sie diese Klassen schreiben.

GWT 1.3

Die GWT-Entwickler veröffentlichten GWT Release Candidate (RC) 1.3 zu dem Zeitpunkt, als das englischsprachige Original dieses TecFeeds veröffentlicht wurde. Dieses Release hat jedoch mit der GWT-Open-Source-Ankündigung zu tun und bietet keine neue Funktionalität. Die Beispielanwendung wurde mit *gwt-mac-1.3.1 RC* und dem offiziellen Release *gwt-mac-1.2.22* getestet.[1]

Sie müssen das GWT-Framework herunterladen und entpacken. Die Download-Adresse ist *http://code.google.com/webtoolkit*. Sie können das Framework für Windows, Linux oder Mac OS X herunterladen.

Das entpackte Archiv enthält

- die Dokumentation einschließlich eines Javadoc-Formats für die GWT-API-Klassen,
- das Java Archive File (JAR) *gwt-user.jar* mit den Benutzeroberflächen-Klassen und anderen Teilen der GWT-API (etwa Modulen zur Verarbeitung der Formate JSON und XML oder zur Durchführung von HTTP-Anfragen in Ihrem Client-Code),
- die JAR-Datei (z.B. *gwt-dev-windows.jar*) mit den proprietären GWT-Entwicklungs-Tools, etwa den Java-Klassen, die Ihren Code »kompilieren« und JavaScript erzeugen,
- die JAR-Datei *gwt-servlet.jar*, die Sie in Tomcat oder anderen Servlet-Containern bereitstellen können (sie enthält die GWT-API für die Benutzeroberflächen-Elemente und andere Objekte. Es handelt sich im Wesentlichen um *gwt-user.jar* ohne die Servlet-API-Klassen aus dem Package *javax*. Die Java Servlet Specification, ein anerkannter Standard für Servlet-Container, erlaubt nicht, dass Packages wie *javax.servlet* in bestimmten Webanwendungen aus JARs geladen werden. Deshalb kann *gwt-servlet.jar* in *WEB-INF/lib* innerhalb Ihrer Java-Webanwendungen platziert werden, wenn Ihr serverseitiger Code erfordert, dass sich die GWT-Klassen im Classpath der Anwendung befinden.), und
- diverse Kommandozeilentools (siehe unten).

Nun sind wir fast so weit, dass wir mit dem Coden anfangen können.

[1] Anmerkung des Übersetzers: Die Übersetzung wurde unter Verwendung von GWT 1.3.3 für Windows erstellt.

Clientseitiger Code in veröffentlichten Anwendungen

Sie brauchen die Datei *gwt-servlet.jar* nicht in Ihre Java-Webanwendungen einzufügen, solange Sie keine serverseitigen Klassen haben, die die GWT-API nutzen, etwa die mit Remote Procedure Call (RPC) verbundenen Klassen und Interfaces. Das liegt daran, dass die Benutzeroberflächen-Java-API des GWT nur während des Deployments verwendet werden, bevor die GWT-Tools oder Compiler das JavaScript erzeugen. Danach veröffentlichen Sie einfach das erzeugte JavaScript mit Ihrer Webanwendung. Das JavaScript wird letztendlich im Browser ausgeführt und hat keine Verbindung mehr zur GWT-Java-API (und der zugehörigen JAR-Datei). Es ist zu diesem Zeitpunkt nur noch JavaScript!

Eine Anwendung mit Hilfe der Kommandozeilentools erstellen

Wenn Sie in das entpackte Verzeichnis des GWT schauen, das in Abbildung 1 gezeigt wird, bilden vier der Dateien die Kommandozeilentools des Toolkits.

Die Kommandozeilenskripten sind dafür gedacht, über ein Kommandozeilenfenster wie die Windows-Eingabeaufforderung oder das Terminal in Linux oder in Mac OS X ausgeführt zu werden.

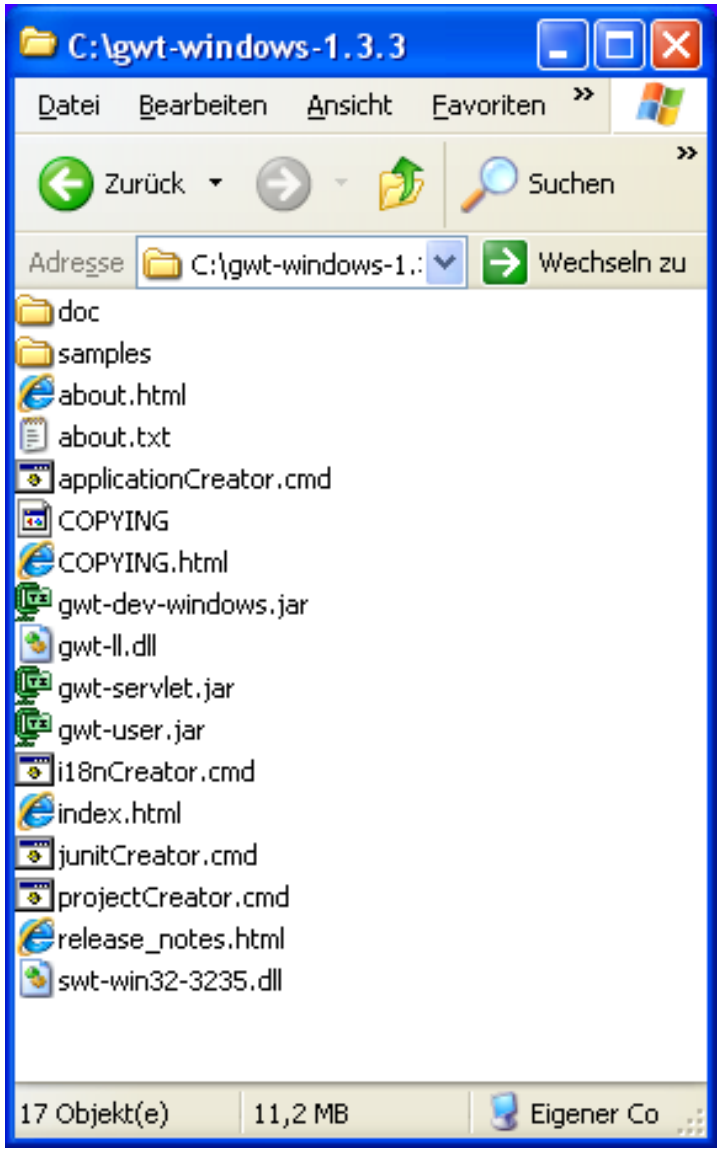


Abb. 1 *Die GWT-Kommandozeilentools innerhalb des entpackten Toolkits*

`projectCreator`

Dieses Skript erzeugt ein Projektgrundgerüst sowie Ant-Build-Dateien oder Eclipse-Projekte, je nachdem, was die Kommandozeile vorgibt.

`i18nCreator`

Dies erzeugt einige der Dateien, die zur Verwendung internationaler Meldungen mit GWT benötigt werden. Das TecFeed beschreibt diesen Anwendungsaspekt in einem späteren Abschnitt.

`junitCreator`

Das Skript kann verwendet werden, um mit dem Einsatz von JUnit mit GWT zu beginnen. Ein späterer Abschnitt beschreibt den Einsatz dieses Kommandozeilentools.

Wir verwenden `applicationCreator` für unsere Anwendung. Ich habe ein Terminal- oder Eingabeaufforderungsfenster geöffnet und Folgendes eingegeben:[2]

```
applicationCreator -out /users/bruceperry/1ebooks/gwt
-overwrite com.parkerriver.gwt.intro.client.GwtAjax
```

Dieser Befehl verwendet das mitgelieferte Shell-Skript `applicationCreator`, um eine Ajax-Anwendung innerhalb einer Datei namens *GwtAjax.java*. zu erstellen. Die erste Option des Befehls `applicationCreator`, `-out`, gibt das Verzeichnis für Ihre Anwendung oder Ihr Projekt an (Standard ist das aktuelle Verzeichnis, in dem sich `applicationCreator` befindet, also das oberste Verzeichnis nach dem Entpacken von GWT).

Die Option `-overwrite` gibt an, dass der Befehl alle bestehenden Dateien überschreiben soll. Das letzte Element ist der voll qualifizierte Name der Klasse, in der sich Ihre Anwendungslogik befindet; sie stellt in der Regel die »Einstiegspunkt«-Klasse dar.

Der Einstiegspunkt ist der erste Bildschirm, mit dem der User innerhalb der Ajax-Anwendung interagiert, etwa ein Login-Fenster oder ein Steuermenü.

projectCreator für Ant und Eclipse verwenden

Das Kommandozeilentool `projectCreator` erzeugt optional ein Eclipse-Projekt oder eine Ant-Build-Datei mit Hilfe der Optionen `-eclipse` oder `-ant`. Zum Beispiel, wenn Ihr Projektname myproject lautet:

```
projectCreator -ant myproject -out /users/bruceperry/
1gwt/test -overwrite
```

Dieser Befehl erzeugt eine Ant-Datei, die die Java-Dateien in */users/bruceperry/1gwt/src* nach */users/bruceperry/1gwt/bin* kompilieren wird. Der Befehl erzeugt auch die Verzeichnisse */users/bruceperry/1gwt/src* und */users/bruceperry/1gwt/test*. ▸

[2] Unter Windows können Sie als Pfadtrennzeichen bei solchen Aufrufen sowohl den plattformeigenen Backslash (\) als auch den Unix-Slash (/) verwenden; Java-Programme verstehen beides.

Schnelleinstieg in Flex 2

Roger Braunstein
Übersetzung von Lars Schulten

ADOBE DEVELOPER LIBRARY

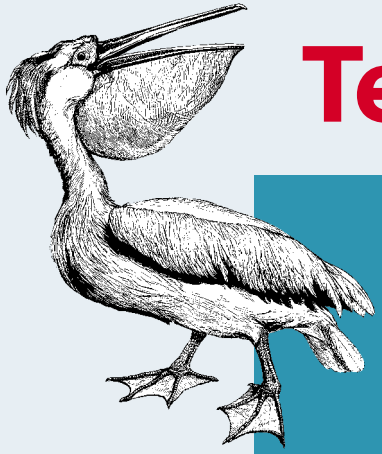
Adobe

Flex ist ein umfassendes Framework zur Entwicklung Flash-basierter Rich Internet Applications. Obwohl die Umgebung noch sehr jung ist, gibt es schon jetzt viele moderne Webanwendungen, die mit Flex 2 realisiert wurden. Auch für Desktop-Anwendungen ist Flex optimal geeignet, wenn es zusammen mit Adobe Apollo zum Einsatz kommt.

Dieses Dokument bietet eine vollständige Einführung in die Programmierung mit Flex 2 und berücksichtigt dabei auch MXML und ActionScript 3.0. Es ist für alle geeignet, die Kenntnisse in einer objektorientierten Programmiersprache und XML haben.

Den Beispielcode zu diesem TecFeed finden Sie auf der Website des Autors unter *http://partlyhuman.com/books/flexshortcut/*.

INHALT



TecFeeds

O'REILLY®

Flex 2 kennenlernen

Flex 2 ist eine deklarative XML-UI-Sprache

Das wahrscheinlich markanteste Kennzeichen von Flex 2 ist MXML, eine neue Sprache, die Sie einsetzen können, um Flex-Anwendungen zu erstellen. Wenn Sie Ihren ersten `<mx:Button>` sehen, können Sie sicher sein, dass Sie MXML und Flex vor sich haben.

Vielleicht haben Sie schon erraten, dass das »ML« in MXML für »Markup-Sprache« steht. Und ein Blick auf MXML-Code macht die Ähnlichkeit mit XHTML oder anderen Markup-Sprachen deutlich. MXML ist eine deklarative XML-Sprache. Das bedeutet, dass Sie, einfach indem Sie deklarieren, schon etwas machen. Wenn Sie Folgendes schreiben:

```
<mx:Panel><mx:Button></mx:Panel>
```

erhalten Sie also einen Button in einem Panel. Die Erzeugung und Zusammensetzung dieser Komponenten wird bereits von der Struktur des Markups impliziert.

Beachten Sie, dass nicht der gesamte Code in einer Flex 2-Anwendung MXML sein muss. In Flex 2 haben Sie immer die Wahl zwischen MXML und ActionScript 3.0, auch wenn Sie nur in Flex 2 MXML verwenden können. Der Gebrauch von deklarativem XML kann bei der Programmierung sehr vorteilhaft sein. Es ist erheblich kompakter, wenn man viele Komponenten aufbaut. Es kann eingesetzt werden, um Design und Code zu trennen. Außerdem kann es extern gestylt und mit visuellen Werkzeugen gestaltet werden.

Einige Unternehmen haben deklarative XML-Sprachen als Mittel zur Erstellung von UIs entdeckt. Man kann also MXML leicht mit Microsofts XAML, AOLs Boxely, Laszlos OpenLaszlo und Netscapes XUL vergleichen. Schon die Vertrautheit mit HTML und JavaScript sorgt dafür, dass Sie sich bei MXML zu Hause fühlen.

Flex 2 ist ein Windowing-Toolkit

Flex 2 fügt den Fähigkeiten von Flash Player 9 ein mächtiges, erweiterbares, gut organisiertes und gut dokumentiertes Windowing-Toolkit hinzu. Ein großer Teil des Werts von Flex 2 steckt im Framework: den Hunderten von Klassen, die bereitstehen, damit Sie sie nutzen und erweitern können.

Die meisten dieser Klassen kümmern sich gleichermaßen um Layout und Anzeige vertrauter GUI-Komponenten wie Buttons, Textfelder und Werkzeugleisten. Wie jedes Windowing-Toolkit soll es Ihnen ersparen, jemals eine weitere Scrollleiste (oder irgendeine andere GUI-Komponente) schreiben zu müssen. Sie können Flex eine Menge lästiger Details überlassen – das Zentrieren von Objekten, das Ersetzen des Mauszeigers des Benutzers oder das Füllen von Auswahllisten mit Daten – und sich gleich den wichtigen Teilen Ihrer Anwendung zuwenden.

Neben den Komponenten, die Flex 2 von Haus aus mitbringt, können Sie auch fast alle Klassen nutzen, die verwendet werden, um Flex 2 selbst aufzubauen. Die Quellen aller Klassen werden in rohem ActionScript angeboten. Das ermöglicht Ihnen, eigene Komponenten oder unkonventionelle Benutzer-

schnittstellen zu erzeugen, die Funktionalitäten von beliebigen Teilen der Flex 2-Bibliothek ausleihen.

Wenn Sie sich einmal daran gewöhnt haben, Flex 2 an Ihrer Seite zu haben, empfindet man es als geradezu lästig, Benutzerschnittstellen in Flash zu programmieren. Das soll jedoch nicht heißen, dass Flex Flash vollständig ersetzt. Flex besitzt keinen Grafikeditor, keine Timeline, keine Animationswerkzeuge und keine Bibliotheksverwaltung wie Flash. Es soll auf einer höheren Ebene arbeiten und bietet Komponenten statt Shapes. Generell könnte man sagen, dass Flex 2 am besten für den Aufbau datengesteuerter, benutzerfreundlicher Benutzerschnittstellen geeignet ist.

Flex 2 basiert auf Flash

Wenn Sie eine Flex 2-Anwendung ausführen, lassen Sie sie im Flash Player (Version 9 und höher) laufen. Egal ob es das Flash Player-Plugin in Ihrem Browser, der Standalone Flash Player oder eine Desktop-Anwendung ist, die über Appolo läuft, Flex 2-Anwendungen laufen innerhalb des Flash Players. Das bedeutet, dass alles, was Sie in Flex 2 schreiben, an den gleichen Orten läuft wie ein Flash-Film: in unterschiedlichen Browsern, auf unterschiedlichen Computern und unter unterschiedlichen Betriebssystemen.

Außerdem sind Flex 2-Anwendungen Flash-Filme. Obwohl Sie sie nicht in der Flash-IDE entwerfen und einen anderen Compiler nutzen, sind Flex 2-Anwendungen in jeder Weise ganz gewöhnliche SWFs. Flex ist also eine Flash-Technologie. Beim Aufbau von Flex 2-Anwendungen können Sie in ActionScript 3.0 programmieren und jeden Teil der Flash 9-API aufrufen. Flex ist eine Obermenge von Flash.

Weniger Worte, mehr Code

Kürzen wir das Geschwatze ab und sehen wir uns eine einfache (allerdings etwas unnütze) Flex 2-Anwendung an. Indem Sie sich etwas Beispielcode ansehen, bekommen Sie ein Gespür dafür, wie Flex 2-Anwendungen erstellt werden, bevor wir uns den Einzelheiten zuwenden. Abbildung 1 zeigt ein Werkzeug zur Temperaturumrechnung, das Celsius in Fahrenheit und Fahrenheit in Celsius umrechnet.

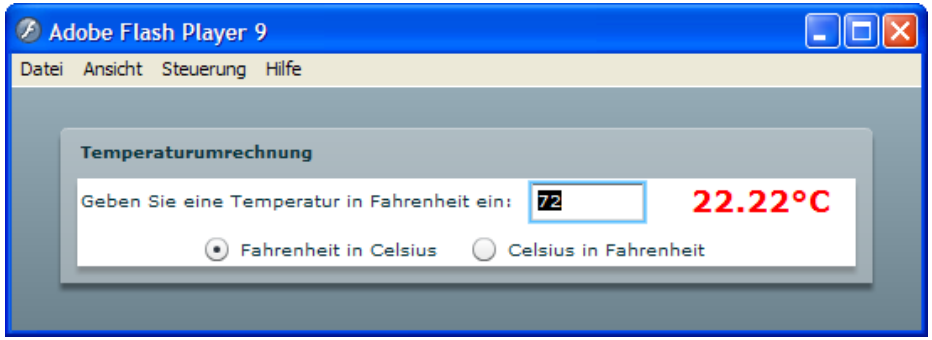


Abb. 1 Ein Temperaturumrechner

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
currentState="FtoC">

   <mx:Script>
      <![CDATA[
```

```
private const C_INPUT_TEXT:String =
"Geben Sie eine Temperatur in Celsius ein:";
private const F_INPUT_TEXT:String =
"Geben Sie eine Temperatur in Fahrenheit ein:";
private const CF_STATE:String = "CtoF";
private const FC_STATE:String = "FtoC";

private function convertTemp():void
{
   var inputTemp:Number =
   parseFloat(inputTempField.text);
   var outputTemp:Number;

   switch(currentState)
   {
      case CF_STATE:
         outputTemp = convertCtoF(inputTemp);
         outputTempField.text =
         outputTempFormatter.format(outputTemp);
         outputTempField.text += "°F";
         break;
      case FC_STATE:
         outputTemp = convertFtoC(inputTemp);
         outputTempField.text =
         outputTempFormatter.format(outputTemp);
         outputTempField.text += "°C";
            break;
         default:
            throw new Error("Unerwarteter Zustand");
      }
   }

   private function clearTemp():void
   {
      outputTempField.text = "";
   }

   private function convertCtoF(celsius:Number):Number
   {
      return 9/5 * (celsius + 32);
   }

   private function convertFtoC(fahrenheit:Number):
   Number
   {
      return 5/9 * (fahrenheit - 32);
   }
]]>
</mx:Script>

<mx:Style>
   .outputTemp {font-size: 18; font-weight: bold;
   color: red}
</mx:Style>
```

```
<mx:states>
   <mx:State name="{FC_STATE}" enterState="convertTemp()">
      <mx:SetProperty target="{inputLabel}" name="text"
      value="{F_INPUT_TEXT}"/>
   </mx:State>
   <mx:State name="{CF_STATE}" enterState="convertTemp()">
      <mx:SetProperty target="{inputLabel}" name="text"
      value="{C_INPUT_TEXT}"/>
   </mx:State>
</mx:states>

<mx:NumberValidator source="{inputTempField}"
property="text"
   triggerEvent="change" valid="convertTemp()"
   invalid="clearTemp()"/>
<mx:NumberFormatter id="outputTempFormatter"
precision="2"/>

<mx:Panel title="Temperaturumrechnung"
layout="vertical" horizontalAlign="center">

   <mx:HBox horizontalAlign="center"
   verticalAlign="middle">
      <mx:Label id="inputLabel"/>
      <mx:TextInput id="inputTempField" maxChars="5"/>
      <mx:Spacer width="10"/>
      <mx:Text id="outputTempField"
      styleName="outputTemp" width="100"/>
   </mx:HBox>

      <mx:HBox horizontalAlign="center"
      creationComplete="conversionMode.selectedValue =
      currentState;">
         <mx:RadioButtonGroup id="conversionMode"
         change="currentState =
         conversionMode.selectedValue as String"/>
      <mx:RadioButton group="{conversionMode}"
         label="Fahrenheit in Celsius" value="FtoC"/>
      <mx:RadioButton group="{conversionMode}"
         label="Celsius in Fahrenheit" value="CtoF"/>
      </mx:HBox>

   </mx:Panel>
</mx:Application>
```

Dieser Hauruck-Temperaturumrechner nutzt Elemente aus vielen verschiedenen Teilen von Flex 2. Mit dieser und den anderen Anwendungen in diesem TecFeed können Sie auf der Begleit-Website zu diesem TecFeed experimentieren (dort können Sie auch den jeweiligen Code herunterladen oder einsehen): *http://www.partlyhuman.com/books/flexshortcut/*.

Inhalte dieses TecFeeds

Im Rest dieses Dokuments werden Sie alle Elemente sehen, die in das vorangehende Beispiel mit eingeflossen sind, und noch vieles mehr. Dieser TecFeed ist in vier übergeordnete Abschnitte eingeteilt. Der erste ist eine Einführung in ActionScript 3.0, die Sie brauchen, um Flex 2-Anwendungen zu schreiben. Der zweite Abschnitt befasst sich mit den am häu-

figsten verwendeten und nützlichsten Komponenten, die Sie in der Flex 2-Bibliothek finden können, und damit, wie Sie direkt aus ihnen Anwendungen aufbauen. Im dann folgenden Abschnitt geht es darum, wie Sie erreichen, dass diese Anwendungen gut aussehen. Der letzte große Teil dieses TecFeeds dreht sich darum, wie man an das Flex-Framework anknüpft, um eigene Komponenten zu erstellen.

Wollen Sie sich so schnell wie möglich in die eigentliche Flex-Entwicklung stürzen, können Sie das folgende Kapitel zu ActionScript 3.0 erst mal überspringen und gegebenenfalls zurückblättern, wenn die Beispiele komplexer werden und mehr Scripting erfordern.

Konventionen

Die Codeauszüge in diesem TecFeed wurden gelegentlich gekürzt, um den Schwerpunkt mehr auf die eigentliche Bedeutung des Auszugs zu legen. Sie können nebeneinander Abschnitte mit ActionScript 3.0 und MXML enthalten. Das impliziert, dass das MXML Teil einer MXML-Datei ist und dass das AS3 in einen `<mx:Script>`-Block oder eine externen ActionScript-Datei eingeschlossen ist. Gleichermaßen sollte CSS in eine externe CSS-Datei gepackt werden. Importanweisungen und die XML-Deklaration `<?xml version="1.0"?>` werden in die Auszüge nicht eingeschlossen.

ActionScript 3.0

Die gemeinsame Sprache von Flex 2-Anwendungen ist ActionScript 3.0. Sie können keine älteren Versionen von ActionScript verwenden, und es ist nicht möglich, ganz ohne ActionScript 3.0 eine funktionierende Anwendung zu erstellen. Außerdem werden wir bald sehen, dass MXML und AS3 grundlegend miteinander verwandt sind. ActionScripts dritte Inkarnation ist eine ausgereifte Sprache und substantiell anders als ActionScript 2.

Dieses Kapitel wird alle wesentlichen Aspekte von ActionScript 3.0 kurz behandeln, ist aber in keiner Weise eine vollständige Einführung. Am Ende dieses TecFeeds befindet sich eine Liste mit weiterem nützlichen Lesestoff, der ergänzende Anleitungen bietet.

Was bringen Sie mit?

Leser, die ActionScript 2 verstehen, dürften mit dem Übergang zu ActionScript 3.0 keine Probleme haben. Das folgende Kapitel widmet sich speziell den Konzepten, die seit AS2 eingeführt wurden. Lesen Sie also weiter!

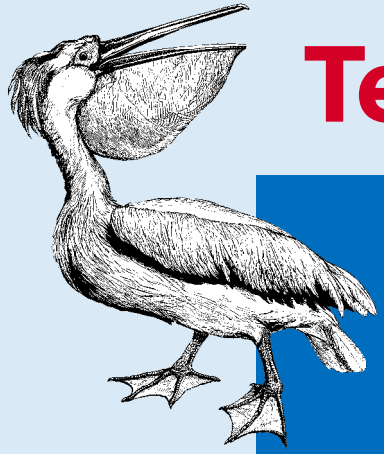
Leser mit Erfahrungen in C# oder Java werden sich bei AS3 nach kurzer Zeit zu Hause fühlen, sollten aber auf ein paar Unterschiede in der Syntax und bei den Features achten. Auf alle Fälle setzt dieses Kapitel zumindest Grundkenntnisse in der objektorientierten Programmierung voraus.

RJS-Templates für Rails

Cody Fauser
Übersetzung von Peter Klicman

RJS-Templates sind spannende und leistungsfähige neue Template-Typen, die in Rails 1.1 integriert wurden. Im Gegensatz zu konventionellen, HTML oder XML generierenden Rails-Templates, erzeugen RJS-Templates JavaScript-Code, der ausgeführt wird, sobald er an den Browser zurückgegeben wird. Diese JavaScript-Generierung erlaubt mit Hilfe von Ajax die Durchführung mehrerer Seiten-Updates, ohne die Seite neu laden zu müssen. Der gesamte JavaScript-Code wird durch einfache, in Ruby geschriebene Templates generiert. Dieses Dokument zeigt Ihnen, wie sich RJS-Templates in das Rails-Framework einfügen, und es erleichtert Ihnen den Einstieg mit einigen einfach zu verstehenden Beispielen.

INHALT



O'REILLY®

Einführung

Was ist RJS?

Ajax ist eine Technik, die dem Benutzer eine Desktopartige, interaktive Erfahrung im Webbrowser beschert. Grundsätzlich ermöglicht Ajax dem Browser, entfernte Requests im Hintergrund auszuführen. Diese Requests aktualisieren die aktuelle Seite, ohne sie neu zu laden. Ruby on Rails unterstützt Ajax direkt in seinem Framework. Remote JavaScript-Templates (RJS) bauen auf der bei Rails 1.0 bestehenden Ajax-Unterstützung auf, gehen aber noch einen Schritt weiter, indem sie das problemlose Aktualisieren mehrerer Seitenelemente ermöglichen.

RJS-Templates sind eine leistungsfähige neue Erweiterung von Rails 1.1. Im Gegensatz zu anderen Rails-Templates, die gerendert und an den Browser gesendet werden, nutzt man RJS-Templates, um Seiten zu aktualisieren, die bereits gerendert wurden.

RJS wurde vom Rails-Kernentwickler Sam Stephenson entwickelt. Sam ist auch der Autor der Prototype-Bibliothek, die zusammen mit Rails ausgeliefert wird.

Rails unterstützte bereits vor der Version 1.1 die Aktualisierung eines einzelnen Seitenelements durch das von einem entfernten Ajax-Aufruf zurückgelieferte Ergebnis, aber die Dinge wurden kompliziert, sobald mehrere Seitenelemente aktualisiert werden mussten. RJS erlaubt die Aktualisierung mehrerer Seitenelemente per Ruby-Code in einem einzelnen Ajax-Request. Sie können eine beliebige Anzahl visueller Scriptaculous-Effekte in einem einzelnen Template nutzen, ohne überhaupt JavaScript verwenden zu müssen. In den meisten Fällen besteht keine Notwendigkeit mehr umzudenken und in JavaScript zu programmieren. Der gesamte JavaScript-Code, der durch den Rails-JavaScript-Generator für Sie erzeugt wird, sowie die Ajax-Response werden automatisch durch die Prototype-Bibliothek evaluiert.

Wie Sie sehen werden, macht es RJS wesentlich einfacher, mehrere Seitenelemente zu aktualisieren, und komplexe visuelle Effekte mit einem einzelnen Ajax-Request aufzubauen.

Wer sollte dieses Dokument lesen?

Dieses Dokument richtet sich an jeden, der Ajax mit dem Ruby on Rails-Framework nutzen möchte. Dieses Dokument geht davon aus, dass Sie mit Ruby on Rails, Ruby und den von beiden verwendeten Konventionen vertraut sind. Auch ein grundlegendes Verständnis von JavaScript, was Ajax ist, und warum Sie es in Ihrer Anwendung nutzen wollen, ist hilfreich. Sollten Sie an irgendeiner Stelle des Buches das vorgestellte Material nicht verstehen, bietet *Agile Web Development with Rails* von Dave Thomas et al. (Pragmatic) die beste Referenz. Dieses Buch bietet eine umfassende Übersicht aller Aspekte von Rails und liefert hervorragendes Hintergrundwissen zu dem hier vorgestellten Material.

Rails-Version

RJS-Templates verlangen Rails 1.1 oder höher. Wenn Sie eine Version älter als Rails 1.1 nutzen, müssen Sie auf die neueste

Version umsteigen. Wenn Sie ein existierendes Projekt um RJS-Templates erweitern wollen, müssen Sie die JavaScript-Bibliotheken Ihres Projekts aktualisieren. Sie erreichen das mit einem bei Rails mitgelieferten Rake-Task:

```
cody> rake rails:update
```

Dieser Rake-Task aktualisiert Ihre Projekt-Konfigurationsdateien, -Skripten und JavaScripts auf die neueste Version von Rails.

Danksagungen

Zuerst einmal möchte ich jedem danken, der sich mit meiner Abwesenheit abfinden musste, während ich dieses Dokument schrieb. Dazu gehören Jaded Pixel, Michael Goodman, alle meine Freunde und insbesondere meine wunderbare Frau Maria. Ich möchte auch Tobias Lütke und Daniel Weinand danken, die meine Arbeit gegengeprüft und mich mit unzähligen Ideen versorgt haben. Dank an Rick Olson und Marcel Molina Jr. für das technische Korrekturlesen. Ich möchte mich auch bei Bridgehead in der Elgin St. in Ottawa bedanken, wo ein Großteil dieser Arbeit getan wurde. Die Bridgehead-Mitarbeiter haben mein Herumtrödeln nie kommentiert und nur einmal versucht, meinen Nachschub an Americanos zu unterbrechen. Schließlich möchte ich noch dem Rails-Kernteam und den zahllosen anderen Mitwirkenden danken, die Rails zu einem so hervorragenden Framework und zu so einer unglaublichen Community gemacht haben.

Einstieg mit einer einfachen Anwendung

Zum Aufwärmen wollen wir mit einem einfachen, einführenden Beispiel beginnen. Ich nenne die Anwendung »Thought Log« (deutsch etwa »Gedanken-Log«). Thought Log nimmt einfach die Eingabe aus einem Textfeld und bindet sie auf der aktuellen Seite ein, ohne diese neu zu laden. Zuerst legen wir ein neues Rails-Projekt an.

```
cody> rails thought_log
```

Nun generieren wir einen neuen Controller, *ThoughtsController*, der unsere Aktionen enthält.

```
cody> ruby script/generate controller Thoughts
    exists app/controllers/
    exists app/helpers/
    create app/views/thoughts
    create test/functional/
    create app/controllers/thoughts_controller.rb
    create test/functional/thoughts_controller_test.rb
    create app/helpers/thoughts_helper.rb
```

Der Generator hat den Controller, den Helper und einen Ordner für die Views des Controllers angelegt. Nun fügen wir zwei Aktionen hinzu. Die erste Aktion, *index()*, gibt die anfänglich leere Liste unserer Gedanken aus. Die zweite Aktion, *log()*, wird per Ajax im Hintergrund aufgerufen und die Response fügt den neuen Gedanken in die bereits gerenderte Index-seite ein.

```
class ThoughtsController < ApplicationController
    def index
    end

    def log
        @thought = params[:thought]
    end
end
```

Die Aktion *log()* weist einfach den Wert von *params[:thought]* der Instanzvariablen *@thought* zu und rendert dann das View-Template *app/views/thoughts/log.rjs*. Wir folgen der Rails-Konvention, dass der Controller standardmäßig ein View-Template mit dem gleichen Namen wie der gerade ausgeführten Aktion rendert. Weil *.rjs*-Templates einfach weitere Templates sind, sucht Rails im View-Ordner des Controllers nach einem Template namens *log.rjs*. Der einzige Nachteil besteht hier darin, dass *.rhtml*- und *.rxml*-Templates vor *.rjs*-Templates gefunden und gerendert werden, wenn mehrere Templates mit dem gleichen Namen im View-Ordner existieren. Das Folgende ist für Rails identisch:

```
def log
end
def log
    render :action => 'log'
end
```

Beachten Sie, dass Sie die Aktion *index()* nicht deklarieren müssen, weil Rails standardmäßig die Aktion *index()* und das

View-Template *index.rhtml* im View-Ordner des Controllers rendert.

Vergessen Sie nicht, die Rails JavaScript-Bibliotheken einzubinden.

RJS-Templates basieren auf der JavaScript-Bibliothek *prototype.js*. Wenn Sie alle visuellen Effekte und Controls von Scriptaculous nutzen wollen, benötigen Sie auch *effects.js, controls.js* und *dragdrop.js*. Sie können sie alle in Ihr Projekt einbinden, indem Sie *javascript_include_tag :defaults* in Ihren View aufnehmen. Das fügt auch Ihre *application.js*-Datei ein, wenn eine existiert.

Als Nächstes erzeugen wir den anfänglichen View. Das ist die Seite, die den entfernten Ajax-Request generiert. Normalerweise würden Sie für Ihre Anwendung ein Layout entwerfen, aber weil dieses Beispiel eine einmalige Angelegenheit ist, packen wir einfach alles in das View-Template. Legen Sie index.rhtml im View-Ordner *app/views/thoughts* an.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="Content-type" content="text/html;
           charset=utf-8" />
        <title>Thought Log</title>
        <%= javascript_include_tag :defaults %>
    </head>
    <body>
        <h1>Meine Gedanken</h1>

        <%= form_remote_tag :url => { :action => 'log' }, :html
              => { :id => 'thought-form' } %>
        <%= text_field_tag 'thought', nil, :size => 40 %>
        <%= submit_tag 'Log thought' %>
        <%= end_form_tag %>

        <div id="thoughts"></div>
    </body>
</html>
```

Das Einfügen der Rails JavaScript-Bibliotheken ist sehr einfach, wenn Sie die Helper-Methode *javascript_include_tag* nutzen. Mit der Übergabe des Symbols *:defaults* als Parameter weisen Sie den Helper an, alle Rails JavaScript-Bibliotheken einzufügen, einschließlich Ihrer eigenen application.js, wenn diese denn existiert.

Nachdem die benötigten JavaScript-Dateien eingebunden wurden, legen wir ein einfaches Formular mit einem einzelnen Textfeld an. Anstelle des traditionellen *form_tag()*-Helpers verwenden wir den Helper *form_remote_tag()*. Der Unterschied besteht darin, dass *form_remote_tag()* das Formular serialisiert und die Daten per Ajax im Hintergrund überträgt, statt die Seite wie bei *form_tag()* zu posten und einfach neu zu laden. Die Controller-Aktion, die die Formulardaten empfängt, wird mit der *:url*-Option festgelegt. In unserem Fall

übertragen wir die Daten an die *log()*-Aktion des aktuellen Controllers, also *ThoughtsController*. Wir weisen dem Formular auch eine *id* über die Option *:html* zu. Das erlaubt uns das Rücksetzen des Formulars aus dem RJS-Template heraus, nachdem die Übertragung des Formulars abgeschlossen ist.

Schließlich besitzt der View einen leeren <*div*>-Tag mit der *id* von thoughts. Erneut bietet die *id* uns die Möglichkeit, das Element aus dem RJS-Template heraus anzusprechen. Das *thoughts*- ist ein Container für angehängte Gedanken, die im Formular festgehalten wurden. Als Nächstes wollen wir das eigentliche RJS-Template aufbauen und Sie werden erkennen, wie alles zusammenpasst.

Nachdem das Haupt-View-Template einsatzbereit ist, benötigen wir ein kleines partielles Template (kurz Partial) zum Rendern der festgehaltenen Gedanken. Der innerhalb des partiellen Templates enthaltene Code könnte auch direkt in das RJS-Template eingefügt werden, aber das RJS-Template bleibt übersichtlich, wenn ein Partial verwendet wird. Legen Sie das partielle Template *app/views/thoughts/_thought.rhtml* mit dem folgenden View-Code an.

```
<p>
   <span style="font-size: 0.8em;">[<%= Time.now.to_s(:db) %>]
      </span>
   <%=h thought %>
</p>
```

Das Partial ist sehr simpel. Es gibt einfach den festgehaltenen Gedanken zusammen mit der aktuellen Uhrzeit innerhalb eines Paragraph-Elements aus. Es ist eine vernünftige Vorgehensweise, Texteingaben von Benutzern mit Hilfe der *h()*-Methode einem Escaping zu unterziehen, um das Ausführen unerwünschter Skripten auf Ihren Seiten zu unterbinden.

Verwenden Sie niemals die :update-Option mit RJS-Aufrufen.

Verwenden Sie niemals die Option *:update* bei entfernten Aufrufen von Aktionen, die RJS-Templates rendern. Die *:update*-Option weist Rails an, ein *Ajax.Updater*-Prototyp-Objekt anstelle eines Ajax.Request-Objekts zu generieren. Der *Ajax.Updater* aktualisiert ein einzelnes DOM-Element durch den Inhalt der HTML-Antwort. RJS-Templates liefern JavaScript-Code an den Browser zurück, der evaluiert werden muss, um die gewünschten Effekte zu erzielen.

Um das RJS-Template anzulegen, folgen wir schließlich der Rails-Konvention, dem Template den gleichen Namen zu geben wie der Controller-Aktion. Auf diese Weise müssen Sie *render()* im Controller nicht explizit aufrufen. Legen Sie *app/views/thoughts/log.rjs* an und fügen Sie den folgenden Code ein.

```
page.insert_html :bottom, 'thoughts', :partial => 'thought'
page.visual_effect :highlight, 'thoughts'
page.form.reset 'thought-form'
```

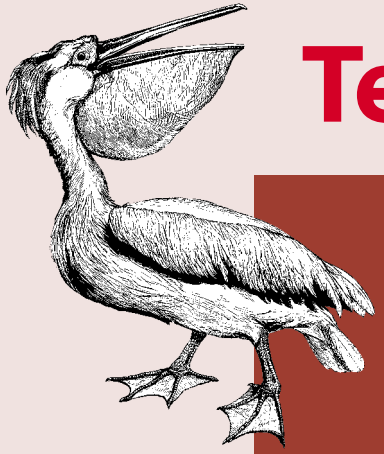
Die erste Frage, die sich stellt, ist, wo dieses *page*-Objekt herkommt. Das *page*-Objekt ist tatsächlich eine Instanz

Capistrano und der Rails Application Lifecycle

Tom Mornini und Marc Loy
Übersetzung von Denny Carl

Erfahren Sie, wie Sie das Deployment Ihrer Rails-Applikationen mit Capistrano problemlos meistern können. Dieses TecFeed zeigt Ihnen, wie Sie Capistrano nutzen können, um das Deployment Ihrer Rails-Applikationen zu automatisieren. Es zeigt Ihnen die Grundlagen, geht aber auch weit darüber hinaus. Sie werden realistische Deployment-Szenarien erleben, die selbst komplexe Serververbünde umfassen. Eine Capistrano-Kurzreferenz ist ebenfalls enthalten. Je größer Ihre Rails-Anwendung wird, desto wichtiger wird es, das Deployment zu automatisieren und Ihre Entwicklungsumgebung gut zu organisieren. Capistrano ist genau das richtige Tool für diese Aufgabe. Und dieses TecFeed zeigt Ihnen, wie Sie es effektiv nutzen können.

INHALT



O'REILLY®

TecFeeds

www.tecfeeds.de

Capistrano

Abgesehen von einigen ausgereiften Frameworks war es bisher sehr aufwändig, mehrere Anwendungsumgebungen zu berücksichtigen, und man war weit entfernt von dem Komfort, den *Ruby on Rails* und *Capistrano* heute bieten. Das Ergebnis ist, dass viele Anwendungen unter nicht gerade idealen Umständen entwickelt und veröffentlicht werden. Tatsächlich ist schon die bloße Entwicklung von Webapplikationen an sich voller Fallen und Probleme.

Ruby on Rails ist es gelungen, die Komplexität, die eine ausgewachsenen Webapplikation mit sich bringt, zu verlagern: weg von der *Konfiguration*, hin zur *Konvention*. Rails umgeht viele der Probleme, die während der Entwicklung auf Sie warten, auf eine einfache, saubere und leicht zu verstehende Art und Weise. Ruby on Rails übernimmt viele Aufgaben, die während der Entwicklung einer Webapplikation immer wieder anfallen, und macht so das Programmieren zur wahren Freude. Einen ähnlichen Ansatz verfolgt Capistrano beim *Deployment* – zwar nicht perfekt in jedem Szenarium, doch stets unkompliziert und vor allem nutzbar für die meisten Anwendungen und Deployment-Umgebungen im produktiven Einsatz.

Aber was macht Capistrano eigentlich? Es kopiert Code. Capistrano schnappt sich Ihre Anwendung, packt sie auf einen entfernten Server und erledigt viele Aufgaben automatisch, die nötig sind, um Ihre Anwendung dort zum Laufen zu bekommen. Es kopiert Code sehr intelligent. Sollte Capistranos Intelligenz für Ihre Zwecke einmal nicht ausreichen, erhalten Sie Zugriff auf interne Tools, so dass Sie Capistrano mit wenig Mühe Ihren speziellen Bedürfnissen entsprechend erweitern können.

Das erste Deployment

Capistrano wurde geschaffen, um Ihnen beim Deployment Ihrer Rails-Anwendungen zu helfen. Wir können uns schon vorstellen, dass Sie es kaum abwarten können, diesen Helfer in Aktion zu erleben. Wir möchten Ihre ganze Aufmerksamkeit, wenn wir Ihnen die besten Verfahren und die Einzelheiten eines guten Deployment-Konzepts zeigen. Lassen Sie uns also Ihr erstes Deployment auf den Weg bringen.

Voraussetzungen

Capistrano setzt auf beiden Seiten des Deployment-Prozesses gewisse Dinge voraus. Das betrifft Ihren Entwicklungsrechner und den Server, auf dem Ihre Anwendung laufen soll. Zugegeben, einige dieser Anforderungen sind lediglich Empfehlungen. Doch wenn Sie diesen Empfehlungen folgen, wird das sonst so fehleranfällige Veröffentlichen einer Webapplikation garantiert zu einem sehr angenehmen Erlebnis.

VORAUSSETZUNGEN AUF DEM ENTWICKLUNGSRECHNER

Sie erledigen Ihre ganze Entwicklungsarbeit während irgendwelcher Nachtflüge, die Sie in die entlegensten Ecken dieser Welt bringen, damit Sie dort Ihre Kunden besuchen können. Ihr ultraportables iBook besitzt zwei Ersatzakkus und Sie haben gerade 30 € dafür bezahlt, dass Sie auf Ihrem Lufthansa-Flug in

einer Höhe von 11.000 km drahtlos ins Internet können. Möglicherweise hocken Sie aber auch daheim in der Dachkammer und arbeiten an Ihrem frisierten Windows 2000-Rechner, den Sie nicht eintauschen wollen (oder nicht mehr in Zahlung geben können). Wie auch immer. In jedem Fall sollten Sie die folgenden wichtigen Punkte berücksichtigen:

Zugang zur Kommandozeile

Capistrano ist ein Tool, das über die *Kommandozeile* bedient wird und das anderer Kommandozeilenprogramme bedarf. Sie sollten also nichts dagegen haben, Befehle von Hand einzugeben. Einige Entwicklungsumgebungen wie *TextMate* oder *Eclipse* sind Meister darin, die Kommandozeile vor Ihnen zu verstecken. Solange diese Tools aber noch keine Capistrano-Unterstützung bieten, müssen Sie in der Lage sein, eine *Shell* oder ein *Terminalfenster* öffnen und nutzen zu können.

Root- oder sudo-Zugang

Damit es nicht zu kompliziert wird, sollten Sie *Root*- oder zumindest *sudo*-Zugang auf Ihrem Entwicklungsrechner haben. (Benutzer von Windows benötigen *Administrator*- oder *Power-User*-Rechte.) Mit Root-Zugang können Sie die benötigte Software einfach und an den vorgesehenen Orten installieren. Das ist wichtig, um einen Entwicklungsrechner, der von mehreren Benutzern verwendet wird, zu verwalten.

ssh

Capistrano nutzt ssh für die Kommunikation mit der Außenwelt. Sie sollten ohnehin *ssh* bei Ihrer täglichen Arbeit nutzen, wenn Sie dabei mit Linux/Unix/Mac OS X-Servern zu tun haben. Wenn Sie allerdings mit Windows-basierten Servern arbeiten, haben Sie möglicherweise andere Mittel für den Fernzugriff. Dennoch müssen Sie ssh auf Ihrem Entwicklungsrechner zum Laufen bringen. (Dem Thema Windows-Server und ssh widmen wir uns im nächsten Kapitel.)

Sie sollten auch Zeit darauf verwenden, Ihre ssh-Schlüssel auf beiden Seiten festzulegen. In diesem TecFeed haben wir leider keine Zeit, diesbezüglich auf Details einzugehen. Doch wir haben für Sie einige Grundlagen im Kapitel »SSH« festgehalten.

Subversion

Ein weiterer Vertreter der »Das haben Sie doch schon längst, oder?«-Kategorie ist ein Tool zur Versionskontrolle Ihrer Anwendung. Capistrano geht davon aus, dass Sie bereits ein einsatzfähiges *Subversion-Repository* besitzen. Clients für Subversion gibt es für jedes Betriebssystem. An das Entwickeln ohne Versionskontrolle möchten wir gar nicht erst denken.

Wir wollen zwar nicht den Kopf in den Sand stecken und die Existenz anderer Versionskontrollsysteme ignorieren, aber Sie sollten ernsthaft erwägen, zu Subversion zu wechseln. Sie werden begeistert sein. An dieser Stelle wieder der

Hinweis: Einige Details, die Sie für die Nutzung von Subversion benötigen, haben wir im Kapitel »Subversion« untergebracht.

Wie Sie sich bestimmt denken können, benötigen Sie natürlich auch eine Webapplikation, die Sie veröffentlichen können. Um dem Titel dieses TecFeeds gerecht zu werden, werden wir mit einer Rails-Anwendung arbeiten. Sie sollten aber wissen, dass Sie mit Capistrano auch Anwendungen veröffentlichen können, die nicht mit Ruby on Rails erstellt worden sind.

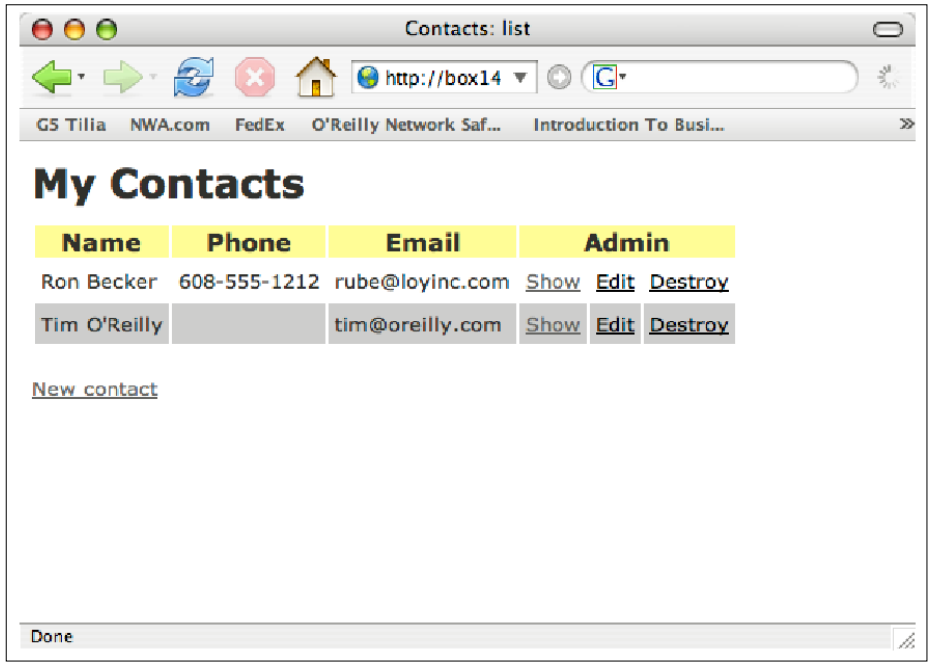


Abb. 1 *Unsere Rails-Anwendung. Diese Liste mit Kontaktinformationen dient uns als Basisanwendung, mit der wir in diesem TecFeed arbeiten werden.*

Unsere Anwendung soll ein einfaches Adressbuch sein, das hauptsächlich auf das gerade bei Rails-Anfängern so beliebte Scaffolding aufbaut. Sie können einen Screenshot der Liste des Programms in Abb. 1 sehen. Diese Anwendung ist bewusst einfach gehalten. Schließlich wollen wir uns auf Capistrano und nicht auf Ruby on Rails konzentrieren, wenngleich beide ziemlich eng miteinander verbunden sind.

In Abb. 2 können Sie sehen, dass wir der eher textbasierten Anwendung eine Zusatzfunktion spendiert haben. Fotos können hochgeladen und mit einem Kontakt verknüpft werden. Die Behandlung solcher hochgeladenen Ressourcen wird immer dann interessant, wenn Sie eine neue Version Ihrer Anwendung veröffentlichen. Dann geht es darum, wie mit neuen und bereits auf dem Server existierenden Bildern verfahren werden soll. Diese Erweiterung unserer Anwendung werden wir benutzen, um später auf den Umgang mit (verzeihen Sie den überbeanspruchten Begriff) persistenten Ressourcen eingehen zu können. In diesem Zusammenhang soll es auch um die Möglichkeiten der Erweiterung von Capistrano gehen. Bedenken Sie: Alles hier dreht sich um das einfache Veröffentlichen und die Instandhaltung Ihrer Anwendung. Alles, was Capistrano bis jetzt nicht für Sie erledigt, ist nur ein Rezept weit entfernt.

Abb. 2 *Unsere Rails-Applikation benutzt Bilder. Jedem Kontakt kann ein Symbol zugeordnet werden, damit das Anwendungsmanagement ein bisschen interessanter wird.*

VORAUSSETZUNGEN AUF DEM WEBSERVER

Wir werden im weiteren Verlauf dieses TecFeeds noch genauer auf die Details einer Deployment-Umgebung, die den Ansprüchen kommerziell eingesetzter Anwendungen genügt, eingehen. Für unsere ersten Schritte sollen uns aber ein Webserver auf Basis von *Apache 2.2*, der ein paar *Mongrel*-Instanzen steuert, und ein einfacher *MySQL*-Datenbankserver genügen.

Capistrano ist in der Lage, Deployments durchzuführen, bei denen die Datenbank auf einem eigenen Server läuft, der von einer ganzen Menge an Webservern angesprochen werden kann. Sie können Ihre Anwendung an mehreren, voneinander unabhängigen Orten veröffentlichen. Sie können *LightTPD* oder Mongrel ohne Apache als Server nutzen. Jetzt soll es aber nur um ein ganz einfaches Deployment gehen, damit wir uns ganz auf die Basisfunktionalitäten von Capistrano konzentrieren können. Aber wir garantieren Ihnen: Wir werden zu den komplizierten Deployments zurückkehren und dabei garantiert einen Blick auf den Umgang mit LightTPD und Mongrel werfen.

Der Server hat seinerseits ein paar Voraussetzungen zu erfüllen:

sshd

Ihr Server muss eingehende SSH-Verbindungen zulassen. Die meisten Unix-basierten Systeme haben eine ziemlich unkomplizierte Art und Weise, damit umzugehen. (Bei Mac OS X müssen Sie beispielsweise nur die Checkbox *Remote Access* unter *System Preferences* aktivieren.) Bei Windows hingegen müssen Sie ein wenig unter die Arme greifen. Im Internet finden Sie sowohl frei verfügbare SSH-Server-Software für Windows (*http://sshwindows.sourceforge.net/*) als auch kommerzielle (*http://www.bitvise.com/winsshd.html*). Die Links sind übrigens nur Beispiele, keine Empfehlungen.

Für welche Variante Sie sich entscheiden, hängt ganz von Ihren Bedürfnissen ab. Wichtig ist nur, dass Sie anschlie-

ßend via SSH eine Verbindung zwischen Ihrem Entwicklungsrechner und dem Webserver, auf dem die Anwendung veröffentlicht werden soll, aufbauen können, ohne dass die Eingabe eines Passworts dabei nötig ist.

POSIX-Konformität

Ihre eingehenden SSH-Verbindungen müssen in einer Shell münden, die Befehle im *POSIX*-Stil versteht. Das bedeutet, dass Sie *bash* oder Ähnliches *csh* vorziehen sollten. Ein weiteres Mal bedarf es an dieser Stelle einiger Schritte, wenn Sie einen Windows-Server einsetzen. Mit ein bisschen Mühe können die *Cygwin*-Tools an dieser Stelle genutzt werden.

Ein Deployment-Benutzerkonto

So ein Benutzerkonto macht die ganze Sache einfacher, auch wenn Sie der einzige Nutzer auf dem Server sind, der dort seine Anwendungen veröffentlicht. (Wenn mehrere Leute Sachen auf einen bestimmten Server hochladen können, ist die Einrichtung eines Benutzers, dem das Deployment ausschließlich zugeordnet ist – eines »dedizierten Deployment-Benutzers« –, unabdingbar, damit Sie nicht den Verstand verlieren.) Stellen Sie sicher, dass dieser Benutzer Zugriff auf das Quell-Repository hat.

WEITERE VORAUSSETZUNGEN

Es gibt noch einige Werte, die Sie gleich benötigen werden. Am besten, Sie legen sich schon einmal Folgendes zurecht:

- Name des Deployment-Rechners
- Name des Deployment-Benutzers
- URL des Subversion-Repository, in dem sich Ihre Anwendung befindet

Und wenn wir schon von Subversion sprechen, darf der Hinweis nicht fehlen, dass sich Ihre Anwendung in einem ortsungebundenen, portablen Zustand im Repository befinden sollte. Was meinen wir mit »ortsungebundener Zustand«? Nun, Ihr Repository sollte keine systemabhängigen Konfigurationsdateien enthalten oder diese an einem Ort speichern, an dem sie keinen Einfluss auf die Anwendung nehmen können. Ein Beispiel: Sind die Einträge für Benutzername und Passwort in Ihrer *database.yml* auf Ihren Entwicklerrechner zugeschnitten? Wir werden Ihre Anwendung auf andere Systeme schieben und wollen doch nicht jedes Mal systemspezifische Änderungen von Hand vornehmen, wenn es sich vermeiden lässt. Den Inhalt Ihres Repository allgemein gültig zu gestalten ist etwas, was Sie wahrscheinlich eh schon gemacht haben, zum Beispiel durch die Subversion-Eigenschaft `ignore` in Ihren Konfigurationsdateien. Falls nicht, lohnt es sich, jetzt darüber nachzudenken.

Sehen Sie's mal so: Wie schwierig ist es denn für Sie, sich auf einem unbekannten System umgehend zurechtzufinden, dort eine Kopie Ihrer Anwendung von Subversion zu holen und diese dann zum Laufen zu bringen? Meinen Sie, es ist so leicht wie die sieben Schritte im Abschnitt »Entwickeln und Testen«? Falls Sie darauf mit Nein antworten, dann ahnen Sie

Geodaten-Mashups

Tools, Frameworks & praktische Anwendungen

Andrew J. Turner
Übersetzung von Jørgen W. Lang

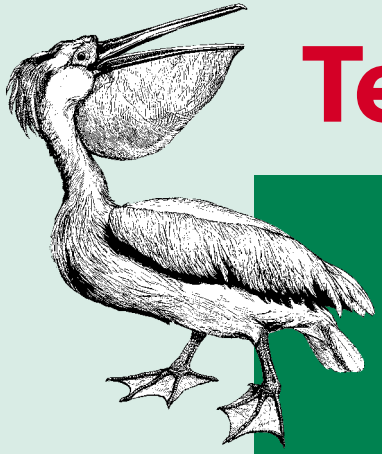
Geodaten-Mashups verbinden die komplexen Techniken der Kartografie und Geoinformationssysteme (GIS) und machen sie für Benutzer und Entwickler leichter zugänglich.

Dieses TecFeed gibt Ihnen eine Einführung in die wachsende Zahl der verfügbaren Werkzeuge, Frameworks und Ressourcen, die das Erstellen von Karten und das Verknüpfen von Ortsinformationen mit Ihren Interessengebieten und Erlebnissen stark vereinfachen.

Lernen Sie die Bedeutung bereits existierender oder gerade entstehender Standards wie GeoRSS, KML und Mikroformate kennen. Finden Sie heraus, wie Sie Ihre Website um dynamische Karten und Ortsangaben erweitern können; wie Sie ermitteln können, von welchem geografischen Ort die Besucher Ihrer Website kommen; wie Sie genealogische Karten und GoogleEarth-Animationen zu Ihrer Familiengeschichte erstellen können oder wie das Geotagging und Veröffentlichen Ihrer Urlaubsfotos funktioniert.

O'REILLY®

INHALT



TecFeeds

www.tecfeeds.de

Einführung

In diesem TecFeed behandeln wir die aktuellen Techniken der Online-Kartografie und der Geodaten-Mashups. Wir gehen auf die wichtigsten Werkzeuge ein, mit denen Sie Ihre eigenen Karten erstellen können, und stellen Ihnen eine Reihe von Beispielprojekten vor, mit denen Ihnen der Einstieg erleichtert werden soll und die Sie hoffentlich zu eigenen Projekten inspirieren.

Unter *http://mapsomething.com* finden Sie eine vollständige Liste der Websites und Ressourcen, die hier genannt werden, damit Sie sie leichter zu Ihren Bookmarks hinzufügen können. Falls Sie Fragen haben oder mir Ihre selbstentwickelten Projekte vorstellen möchten, können Sie mich auch direkt unter der Adresse *andrew@highearthorbit.com* erreichen.

Was sind Geodaten-Mashups?

Seit Jahrhunderten zeichnet die Kartografie Forschungen und Entdeckungen auf. Sie wies Seeleuten den Weg über die Ozeane und half dabei, die Wildnis zu erkunden. Ähnlich wie der Schritt vom Papier zum Hypertext, haben sich auch die Landkarten mit ihren geheimnisvollen Linien und Bögen auf Pergament zu dynamischen Darstellungen entfernter geografischer Informationen entwickelt. Während die meisten Geoinformationssysteme (GIS) immer noch kostspielige Programme sind, die sich nur von speziell ausgebildeten Experten bedienen lassen, stehen der Öffentlichkeit mit Werkzeugen wie MapQuest und Yahoo! Maps neuerdings auch Kartografie-Werkzeuge zur Verfügung, die von Normalsterblichen verstanden und bedient werden können. Die Veröffentlichung von GoogleMaps eröffnete Web-Entwicklern und -Benutzern neue Möglichkeiten, ihre Umgebung zu verstehen und darzustellen und löste damit eine wahre Interessenflut an der Online-Kartografie aus.

Obwohl es anfangs noch kein *Application Programming Interface* (API) für GoogleMaps gab, fanden Entwickler nach und nach heraus, wie sie die Karten für ihre eigenen Zwecke einsetzen konnten. Nachdem Google dann eine öffentliche API zur Verfügung stellte, war es Entwicklern und Nutzern möglich, geografische Daten schnell und einfach auf gemeinsam benutzbaren Karten abzubilden. Waren diese Formen der Nutzung früher eine klare GIS-Domäne, so sorgte diese neue Form der Karten (engl. *Sharable Maps* oder auch *Mashups*) für ein massives Ansteigen der Verbreitung dieser Anwendungen. Das Wort *Mashup* stammt ursprünglich aus dem Verknüpfen von Informationen, wie etwa der Wohnsituation in Verbindung mit der Verbrechenshäufigkeit, z.B. in Chicago. Aber was fasziniert die Leute so an Karten? Warum haben »echte« Orte gerade in der Online-Welt der sofortigen globalen Kommunikation und Anonymität so einen hohen Stellenwert?

Geodaten-Mashups umfassen Techniken und Werkzeuge, die sich außerhalb der Anwendungen traditioneller geografischer Informationssysteme bewegen. Im Englischen wurde dafür der Begriff »Neogeography« (neue Geografie) geprägt. Während ein Kartograf vielleicht ArcGIS verwendet hat, über die

Unterschiede zwischen Mercator- und Mollweide-Projektion diskutierte und möglicherweise Landbesitzstreitigkeiten schlichtete, verwendet ein Web-Geograf eine Kartografie-API wie GoogleMaps, diskutiert den Unterschied zwischen GPX und KML und versieht seine Fotos mit Geotags, um eine Karte seines letzten Sommerurlaubs zu erstellen.

Hauptsächlich geht es bei Geo-Mashups also um Leute, die ihre eigenen Karten erstellen und benutzen – zu eigenen Bedingungen und indem sie existierende Werkzeuge nach Bedarf kombinieren. Bei Geo-Mashups werden Ortsinformationen mit Freunden und Besuchern geteilt, wodurch Zusammenhänge leichter klar werden und Verständnis durch Ortskenntnis vermittelt wird.

Außerdem machen Geo-Mashups Spaß. Warum sonst sollten Leute eine Karte der Orte in der Fernsehserie *24* erstellen oder anderen den Ort ihres ersten Kusses mitteilen wollen? Plötzlich ist die Frage »Wo war das noch?« ganz leicht zu beantworten.

Einige Grundbegriffe

Wenn wir von Geodaten-Mashups sprechen, ist es wichtig, einige hilfreiche Fachbegriffe zu kennen. Manche kennen Sie vielleicht schon, während andere neben einer allgemeinen Bedeutung auch noch in anderen Zusammenhängen verwendet werden können. Diese Liste soll Ihnen helfen, beim Lesen dieses TecFeeds leichter voran zu kommen. Sie hat eine logische Reihenfolge, sodass die Begriffe aufeinander aufbauen können:

Koordinaten

Geografische Koordinaten bezeichnen eine absolute Position auf der Erde (oder einem anderen Körper). Typischerweise handelt es sich dabei um Breiten- und Längenangaben, die sich auf den so genannten *WGS84-Ellipsoiden* (siehe *http://de.wikipedia.org/wiki/WGS84*) beziehen. Gelegentlich werden hierfür die Abkürzungen *Lat* (Breite) und *Lon* (Länge) verwendet. Der Breitengrad variiert von Norden nach Süden. Auf dem Äquator beträgt die Breite 0 Grad. Zu den Polen steigt sie nach Norden bis auf 90 Grad (Nordpol), bzw. sinkt nach Süden hin bis auf -90 Grad (Südpol). Der Längengrad variiert dagegen von Ost nach West. Auf dem so genannten *Nullmeridian* beträgt die geografische Breite 0 Grad und kann bis zu 180 Grad (Osten) bzw. –180 Grad (Westen) betragen.

Die Koordinaten können in verschiedenen Formaten angegeben werden:

- Dezimalgrad (engl. *decimal degrees* DD): 29.975
- Grad-Minuten-Sekunden (engl. *degrees-minutes-seconds*, DMS): N29° 58' 30''
- Grad-Dezimalminuten (engl. *degrees-minutes*, DM): 29° 58.8'

Man spricht auch von Bogengraden (°), Bogenminuten (') und Bogensekunden (''), wobei 1° aus 60' (Bogenminuten) besteht sowie 1 Bogenminute aus 60'' (Bogensekunden). Bei Dezimalgrad/-minuten/-sekunden werden dezimale

Nachkommastellen angegeben. Siehe dazu auch *http://de.wikipedia.org/wiki/Geographische_Breite*

Projektion

Um die dreidimensionale Erde auf einer zweidimensionalen Karte abzubilden, ist eine Projektion notwendig. Tatsächlich ist die Erde übrigens keine perfekte Kugel, sondern ein abgeflachter Rotationsellipsoid. Typischerweise wird dieser Körper auf eine zweidimensionale Kartenfläche abgebildet. Hierbei werden die Mercator- und die Rechteck-Projektion (d.h. ohne Transformationen) am häufigsten verwendet. GoogleMaps verwendet beispielsweise die Mercator-Projektion, die sich gut für vergrößerte Ansichten eignet, bei größeren Maßstäben aber nicht ganz akkurat ist. Es ist wichtig, die Eigenheiten der verschiedenen Projektionen und ihre Auswirkungen auf die jeweilige Anwendung zu verstehen. Dies gilt besonders, wenn Sie die Karten verschiedener Dienste miteinander kombinieren möchten.

POI (Points of Interest)

Die so genannten Points Of Interest (POI) bezeichnen bedeutende Orte, wie öffentliche Gebäude, Dienstleistungen für Reisende (Sehenswürdigkeiten, Hotels, Restaurants, etc.) oder benutzerdefinierte Wegmarken. Dies können beispielsweise Hotels, Restaurants, der Startpunkt einer Wanderroute (engl. trail head), das Haus eines Freundes, Aussichtspunkte oder ein Tauchrevier sein. Außerdem gibt es noch so genannte AOI (Areas Of Interest, interessante Gebiete), die aus mehreren POIs bestehen, oder das geografische Gebiet um einen bestimmten Punkt herum bezeichnen können.

Extent (Ausdehnung)

Extent ist das Begrenzungsrechteck (engl. bounding box, oder kurz *bbox*) oder die größte Breite und Länge einer AOI. Die Ausdehnung definiert die nördliche und südliche Breite und die östliche und westliche Länge. Mit Hilfe von Extents lassen sich AOIs am einfachsten ausdrücken. Daher kommen sie besonders bei Abfragen von Webservices zum Einsatz.

Kacheln (engl. tiles)

Dynamische, bzw. verschiebbare (engl. *slippy*) Karten bestehen aus mehreren quadratischen Einzelbildern, die als »Kacheln« bezeichnet werden. Diese werden neben- und übereinander angeordnet und vermitteln so den Eindruck einer größeren, verschiebbaren Karte.

Geolocation

Dieser Begriff bezeichnet die Technik, anhand bestimmter Daten die Position von etwas zu bestimmen. So ist es beispielsweise möglich, mithilfe der IP-Adresse darauf zu schließen, in welcher Stadt ein bestimmter Computer steht, oder anhand des verwendeten Mobilfunkmastes zu ermitteln, wo sich ein bestimmtes Handy gerade befindet. Geolocation ist nützlich, wenn herausgefunden werden soll, wo sich ein Benutzer oder ein Gerät gerade befindet, ohne dass hierfür zusätzliche Daten eingegeben werden müssen. GPS ist eine spezielle Umsetzung dieser Geolocation-Technologie.

GPS – Global Positioning System

GPS steht eigentlich für ein Netzwerk aus Satelliten des U.S.-Militärs, mit dem man einen Standort in drei Dimensionen bestimmen kann. Die Abkürzung GPS bezeichnet gelegentlich auch verschiedene Hilfsmittel zum Ermitteln von geografischen Koordinaten.

Einfach gesagt besteht das GPS-System aus einer Reihe von Satelliten, die in großer Höhe die Erde umkreisen. Dabei senden sie ständig ihre gegenwärtige Position und Zeit. GPS-Empfänger benutzen mehrere dieser Signale, um ihre eigene aktuelle Position und Zeit zu bestimmen.

Das europäische GPS-System Galileo ist vermutlich 2010 einsatzfähig und soll ähnliche, aber auch weiterführende Funktionen als das gegenwärtige GPS bieten.

Geotag

Das *Geotagging* bezeichnet das Hinzufügen von geografischen Informationen zu einem Dokument, Foto, einer Audiodatei oder anderen Daten. Die für Geotags verwendeten Formate hängen davon ab, auf welche Art von Dokument sie sich beziehen. So ist es bei Fotografien möglich, Ortsinformationen direkt im EXIF-Header der Datei selbst einzubetten. In vielen Web-Applikationen können geografische Informationen, z.B. RSS-Feeds, als Werte-Triplets in den benutzerdefinierten Tags weitergegeben werden.

Webservice

Als *Webservice* bezeichnet man eine Ressource, über die von einem Anbieter Daten oder bestimmte Funktionalitäten über das Web zur Verfügung gestellt werden. Beispiele für Webservices sind so genannte Geocoder, die die geografische Breite und Länge für eine bestimmte Adresse zurückgeben (z.B. *http://www.active-value.de/geocoder/*), oder Dienste, die alle Fotos ausgeben, die in 50 km Umkreis von einem bestimmten Ort aufgenommen wurden. Webservices verwenden die Protokolle REST (Representational State Transfer) oder SOAP (Simple Object Access Protocol), um Programmen oder anderen Sites den Zugriff auf diese Daten zu ermöglichen.

Datenformate

Ortsangaben können in vielen verschiedenen Formaten gespeichert werden. Aufgrund ihrer Wurzeln in der Programmierung wird bei Geodaten-Mashups Wert darauf gelegt, Daten in einer einfachen, auch von Menschen lesbaren »Sprache« zu speichern und weiterzugeben. Da immer mehr Menschen Zugang zu Breitbandanschlüssen haben, können die Daten schneller übertragen und zusätzliche Informationen für immer neue Dateiformate bereitgestellt werden.

Auch wenn die verwendeten Datenformate recht gut verständlich sind, kann das Schreiben dieser Daten von Hand schnell zu einer komplexen und zeitaufwändigen Aufgabe werden. Daher werden wir uns in den folgenden Abschnitten auch mit Werkzeugen beschäftigen, die das Hinzufügen und Auslesen dieser Daten für Ihre Projekte vereinfacht. In diesem Abschnitt geht es aber erst einmal um die grundsätzlichen Konzepte der verschiedenen Datenformate, ihre Möglichkeiten und sinnvolle Einsatzgebiete.

GPX

Softwarehersteller und Werkzeuge verwenden eine Reihe verschiedener Standards, um geografische Informationen zu beschreiben. Am häufigsten kommt hierbei GPX, das GPS-Austauschformat (engl. ***G**PS e**x**change **f**ormat*) zum Einsatz. Dieser Standard verwendet eine XML-Dateibeschreibung, um Wegmarken und -strecken aus GPS-Geräten zu speichern und zu übertragen. Hierbei verwenden die meisten GPS-Empfänger intern ihr eigenes Dateiformat. Der GPX-Standard ermöglicht es Entwicklern, Werkzeuge zu schreiben, mit denen die Formate verschiedener Geräte ineinander umgewandelt werden können.

In der Kopfzeile (dem »Header«) einer GPX-Datei werden allgemeine Informationen gespeichert. Hierzu gehört beispielsweise die verwendete GPX-Version, das Programm oder das Gerät, das die Datei erzeugt hat und eine Reihe weiterer Informationen zum XML-Namensraum:

```
<?xml version="1.0" encoding="UTF-8"?>
   <gpx version="1.0"
   creator="GPSBabel - http://www.gpsbabel.org"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="http://www.topografix.com/GPX/1/0"
   xsi:schemaLocation="http://www.topografix.com/GPX/1/0
   http://www.topografix.com/GPX/1/0/gpx.xsd">
```

Unterhalb des Headers stehen die eigentlichen Informationen zu Wegmarken und Routen.

Eine *Wegmarke* (engl. *waypoint*) speichert die Breite, Länge und (optional) die Höhe eines Ortes. Im so genannten »Body« der Wegmarke werden zusätzliche Informationen wie dessen Name (*name*), Kommentare (*cmt*) und Erkennungsmerkmale (*symbol*, *sym*) gespeichert:

```
<wpt lat ="42.277881" lon ="-83.740791">
   <name>01WiFi</name>
   <cmt>ESPRESSO ROYAL</cmt>
   <sym>Hoher Turm</sym>
</wpt>
```

Ein *Track* (*trk*) bezeichnet eine Liste von Wegmarken und enthält außerdem meist noch ein Element für Zeitangaben (*time*). Das ist praktisch für ein späteres Geotagging von Fotos oder Videos, sowie für die Erstellung von Animationen Ihrer Wegstrecken. Durch die Zeitangabe lassen sich die Daten beispielsweise mit einer Kamera oder einem anderen Aufnahmegerät synchronisieren:

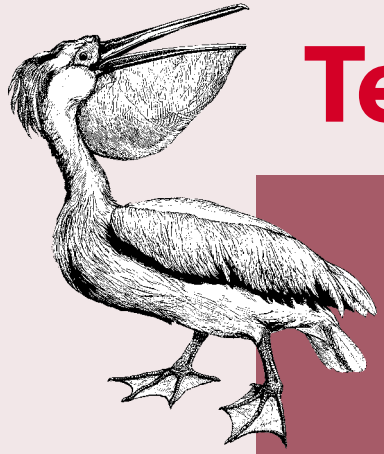
```
<bounds minlat="42.423656691" minlon ="-83.493977330"
      maxlat="42.502436669" maxlon="-83.144850082" />
<trk>
   <trkseg>
      <trkpt lat="42.500937812" lon="-83.147198063">
         <ele>-0.846357</ele>
         <time>2006-06-26T21:42:38Z</time>
      </trkpt>
```

Mikroformate

Brian Suda
Übersetzung von Nikolaus Schüler

Mit Mikroformaten lassen sich strukturierte Informationen in HTML-Seiten verstauen. Diese Informationen sind für Menschen lesbar, können aber auch von Software extrahiert und ausgewertet werden. Das vorliegende TecFeed ist eine allgemeine Einführung in das Konzept und die Verwendung dieser Mikroformate. Sie erfahren in zahlreichen Beispielen, welche Mikroformate es gibt und wie Sie die heute verbreiteten Mikroformate in Ihre Seiten einbauen. Es wird auch erläutert, woher die Idee zu dieser Technologie stammt, was an ihr so zwingend ist, und inwiefern ihr Entwicklungsprozess für alle offen steht. Mikroformate werden schon millionenfach verwendet – nun haben auch Sie die Gelegenheit zu erfahren, worum es hier eigentlich geht.

INHALT



TecFeeds

O'REILLY®

www.tecfeeds.de

Was sind Mikroformate?

Bei Mikroformaten bringt man einfach strukturierte Daten in den Attributen von HTML-Elementen unter. Das Ziel ist es, eine Webseite mit semantischem Markup zu versehen, das von anderen verwendet werden kann, um zusätzliche, nützliche Informationen zu extrahieren, auszuwerten und weiterzuverarbeiten.

Ein Attribut eines HTML-Elements kann bekanntlich eine fast beliebige Zeichenkette enthalten, man bringt ja unter anderem auch Skripte darin unter, zum Beispiel JavaScript. Dies nutzt man bei Mikroformaten aus, um Daten in diesen Attributen zu verstauen, so etwa Adressdaten (Nachname, Vorname, Straße etc.), Bewertungen (»Ich gebe diesem Buch fünf von fünf Sternen«) und Abstimmungen (»bin dafür«, »bin dagegen«, »enthalte mich der Stimme«), oder Daten für Lebensläufe (»habe hier studiert«, »habe dort gearbeitet«). Allen diesen Daten ist gemeinsam, dass sie recht einfach strukturiert sind; es handelt sich also um die Sorte Daten, die man sonst in einer CSV-Datei (mit Kommas getrennte Felder) unterbringt. Geschachtelte oder sonstwie strukturierte Daten sind nicht die erste Wahl für Mikroformate.

Was sind die Vorteile von Mikroformaten?

Sie sind eine einfache Technik. Schreibe eine Zeichenkette in ein HTML-Attribut und interpretiere sie als Daten. Das war's. Sie erfordern kein »Aufbohren« von HTML um neue Elemente und somit auch keine neue oder erweiterte DTD oder zusätzliche Namensräume. Sie erfordern somit auch kein Komitee, das den Standard nach langen Sitzungen und vielen Jahren für gut befindet (oder eben auch nicht). Das ist einer der Hintergründe ihrer Entstehung: Mikroformate sind als »Grasswurzelrevolution« entstanden, weil manche Leute nicht auf die Umsetzung und Standardisierung komplexerer Lösungen durch das W3C warten wollten. Das W3C arbeitet in einer eigenen Arbeitsgruppe an Plänen für ein »Semantisches Web«, aber das sind schwergewichtige Technologien, über die man sich erst einmal einig werden muss. Inzwischen kann man die einfachen Ontologien wie Adressen, Wahlergebnisse und Bewertungen »auf dem kleinen Dienstweg« mit Mikroformaten erledigen – und genau das ist ihre ökologische Nische.

Mikroformate sind ohne jede Erweiterung mit dem zu machen, was bereits da ist. Und glaubt man dem Eintrag in der Wikipedia, wurden sie bei der Erfindung von HTML bereits vorausgeahnt. (Was man natürlich im Nachhinein leicht sagen kann. Schließlich wurde HTML mit Attributen entworfen und mehr braucht man ja für Mikroformate nicht.)

Vor allem: Sie sind nicht aufdringlich. Wenn Sie die Information nicht wollen, die eine HTML-Seite enthält, in der Mikroformate verwendet werden: Einfach ignorieren und gar nicht hinsehen. Niemand zwingt Sie, Mikroformate zu nutzen und wenn man sie ignoriert, bleibt immer noch der ganze Rest der HTML-Seite. Und das ist meist immer noch ein ganze Menge, sonst würden Sie die Webseite wohl nicht besuchen.

Ihre Definition ist einfach und in wenigen Minuten erklärt. Versuchen Sie das Gleiche mal mit XPath. Obwohl dieser Vergleich ein wenig unfair ist – XPath ist extrem mächtig und so braucht man eben auch etwas länger, um es zu erklären. Aber es ist ein Körnchen Wahrheit dran: XML-Anwendungen neigen dazu, extrem formal und schwer erklärbar zu sein. XPath, XQuery, XInclude und Co. sind Schwergewichte, die nicht umsonst dicke Bücher benötigen, um verstanden zu werden.

Was sind die Grenzen von Mikroformaten?

Die Kehrseite der Einfachheit von Mikroformaten ist zugleich ihr größter Nachteil, vor allem, wenn man versucht, sie zu überlasten und komplexe Daten in ihnen zu verstauen: Alles ist nur Konvention, es gibt keine Validierung. Der Autor und der Nutzer sind sich per Absprache einig, dasselbe zu meinen. Das birgt immer ein Risiko. Man vertippt sich und schreibt statt »vote-against« ein »vote-again«. Ups. Dummerweise gibt es kein Tool, das einem dann auf die Finger klopft. Es funktioniert nicht und man muss erstmal darauf kommen, warum es nicht geht.

Es gibt keine formelle Festlegung wie etwa die DTD oder Schemas bei XML. Man kann also auch keine Parser schreiben, die gegen eine formale Definition prüfen, ob die Daten dieser Definition entsprechen. Natürlich kann man Tools schreiben, die ein bestimmtes Mikroformat überprüfen, aber da geht es wieder nur um Konvention. Man bedenke aber: Es gibt etliche Konventionen, die leidlich bis gut funktionieren. Von den gesellschaftlichen angefangen – man bietet älteren Leuten im Bus seinen Sitzplatz an, man spricht nicht mit vollem Mund – bis zu den Konventionen in Programmiersprachen wie Java. Eine Methode, die eine Eigenschaft abfragt (also ein »Getter«) hat per Konvention den Namen getEigenschaft(), (bzw. isEigenschaft() bei boolean-Werten); eine Methode, die sie setzt (also ein »Setter«), hat den Namen setEigenschaft(). Darauf beruht immerhin ein ganzes Framework, das Java-Beans-Framework, und die Sache funktioniert gut. Und auch hier ist die Konvention eine praktische Alternative zu viel komplizierteren Infrastrukturen, auf die man dasselbe Framework hätte aufbauen können.

Mikroformate haben also ihre Grenzen. Für eine bestimmte Aufgabe, nämlich das Auszeichnen übersichtlicher Daten, funktionieren Mikroformate gut, für komplexere Aufgaben wie Skripte würde das Fehlen von formalen Definitionen sie zum Albtraum machen. Und auch komplizierter strukturierte, etwa verschachtelte Daten, die Verweise aufeinander beinhalten, sind nicht besonders gut für Mikroformate geeignet. So etwas sollte man dann schon XML überlassen. Aber die Entwickler sagen es selber: Mikroformate sind nicht dazu, alles zu können, sondern das, was sie machen, gut zu machen.

Die Zielgruppe dieses TecFeed

Warum sollten Sie über Mikroformate Bescheid wissen?

- Weil Sie neugierig sind und über neue Entwicklungen im Web informiert sein wollen.
- Weil Sie für das Web entwickeln, damit ihr Brot verdienen und Ihre Werkzeugsammlung um Mikroformate erweitern wollen, um auch hier ganz vorne dran zu sein.
- Und vor allem: Weil Sie die Vorteile nutzen wollen und Ihre eigene Website oder die Ihrer Kunden mit Mikroformaten aufpeppen wollen. So können Sie beispielsweise auf einfache Weise Ihren Lebenslauf im Web hinterlegen. Vielleicht geraten Sie ja in die richtige Suche und der Traumarbeitgeber beißt an. Oder Sie hinterlegen Ihre Kontaktdaten und der Traumpartner meldet sich. (Ok, wahrscheinlicher ist es, dass Spammer die Daten durchwühlen und Sie fortan mit Angeboten für alles Mögliche und Unmögliche bombardiert werden.) Und weil Sie das Ganze hier und jetzt, mal eben so, erledigen wollen, statt auf den 300-seitigen RFC des W3C »Lebenslauf-Erweiterung für HTML/XHTML« zu warten und dann noch abzuwarten, bis er sich im Web durchgesetzt hat und allgemein verwendet wird.

Mikroformate sind das Web-Äquivalent zur »Steuererklärung auf einem Bierdeckel«. Während wir auf diese wohl noch eine ganze Weile warten, sind Mikroformate schon da, quicklebendig und an vielen Stellen im Einsatz. Es lohnt sich, einen Blick auf sie zu werfen, und da sie schnell erklärt sind, kostet Sie das nicht allzuviel Zeit.

Hinweis

In diesem Buch sehen Sie, wie man Mikroformate nutzt und sie in die eigenen Webseiten einbaut. Wie man sie auswertet und weiterverarbeitet, besprechen wir hier nicht.

HTML-Semantik

Je nach der betrachteten Version bietet HTML etwa 80 verschiedene Elemente, darunter etliche Elemente mit semantischer Bedeutung. Wenn man ein Mikroformat entwirft, sollte man also schauen, ob es nicht schon passende Elemente gibt. Elemente wie `h1`…`h6` für Überschriften, das `p`-Element für Paragraphen und so weiter haben bereits eine semantische Bedeutung. Müssen wir jedoch eine inhaltliche Bedeutung ausdrücken, für die es kein entsprechendes Element gibt, bietet uns HTML mindestens drei weitere Möglichkeiten, semantische Bedeutungen auszudrücken, und zwar über das `meta`-Element, die Attribute `rel` und `rev` sowie das Attribut `class`.

```
<meta name="beschreibung" content="Über diese Seite"/>
<link href="/css/style.css" rel="stylesheet"/>
<div class="autor">Brian Suda</div>
```

Der springende Punkt bei Mikroformaten ist, dass sie in erster Linie für Menschen lesbar sind. Von den drei Möglichkeiten, Semantik zu einem HTML-Dokument hinzuzufügen, ist eine also das `meta`-Element, das im `head`-Element von HTML versteckt ist. Das verletzt jedoch das Prinzip, alles für das menschliche Auge gut lesbar zu machen, so dass ich mich in diesem TecFeed auf sichtbare Inhalte konzentrieren werde.

Aus ähnlichen Gründen wie beim `meta`-Element sollten Webautoren die CSS-Eigenschaft `display:none` meiden. Sie versteckt Daten vor den Besuchern der Webseite und verletzt damit eines der grundlegenden Prinzipien von Mikroformaten: Daten, die aus dem Blickfeld geraten, geraten in Vergessenheit und werden bald vernachlässigt und fehlerhaft. Wenn Daten für Sie wichtig genug sind, um sie so in HTML aufzunehmen, dass man sie extrahieren kann, dann ist es wahrscheinlich auch wichtig, dass die Besucher sie sehen können. Darum sollten Sie der Versuchung widerstehen, Daten zu verstecken. Es gibt andere Optionen, wie etwa das Include-Pattern (es wird später besprochen), das eher der natürlichen Vorgehensweise beim Schreiben von HTML-Code entspricht.

Ein häufiges Missverständnis über HTML ist, dass man mehrere durch Leerzeichen getrennte Werte in einem einzigen Attribut unterbringen kann. Wenn wir HTML auf diese Weise semantisch erweitern, können einige Daten einen doppelten Zweck erfüllen, mehrere Werte aufnehmen und somit mehrere verschiedene Bedeutungen zur gleichen Zeit haben, etwas, was das gute, alte XML nicht kann. Das `class`-Attribut in diesem Beispiel verwendet eine durch Leerzeichen getrennte Liste, um denselben Daten mehrere Bedeutungen zu verleihen.

```
<a
  href="mailto:brian@suda.co.uk"
  rel="me"
  rev="made"
  class="autor informatiker email"
>Brian Suda</a>
```

In diesem Beispiel steht »Brian Suda« aufgrund des »me«-Wertes zu dieser Seite in Beziehung. Hier handelt es sich um eine XFN-Eigenschaft. XFN ist ein elementares Mikroformat, das eingesetzt wird, um Beziehungen darzustellen. Es ist eines von vielen in diesem Buch besprochenen Mikroformaten. Mit Hilfe des `rev`-Attributs können wir ausdrücken, dass »brian@suda.co.uk« diese Seite gemacht hat. Wir wissen auch, dass Brian Suda ein »Informatiker« ist, dass er der »Autor« ist und dass er die »Email«-Adresse *brian@suda.co.uk* hat. Um die Semantik hinter diesen verschiedenen Werten zu verstehen, müssen Sie auf ihre Definitionen zurückgreifen. Diese Definitionen werden über das `profile`-Attribut von HTML zugänglich gemacht.

Rel versus Rev

Die Attribute `rel` und `rev` findet man nur beim `a`-Element und beim `link`-Element. Diese zwei Attribute werden nicht oft eingesetzt, und ihre Bedeutung wird oft durcheinandergebracht.

Diese Attribute beschreiben die Art des Links und die Beziehung zwischen der aktuellen Seite und der Seite, auf die der Link verweist. Die Möglichkeit, eine Verknüpfung zu beschreiben, ist ein mächtiges Werkzeug, um Semantik zu HTML hinzuzufügen.

Rel

»Rel« ist eine Abkürzung für »Beziehung« (relationship). Der Wert des `rel`-Attributs beschreibt die Beziehung zwischen der Seite und der Ressource, auf die das `href`-Attribut verweist.

```
<link href="/css/style.css" rel="stylesheet"/>
```

Die Semantik von `rel` beschreibt in diesem Beispiel den Wert des `href`-Attributs, nämlich »`/css/style.css`« als den Wert »`stylesheet`« des `rel`-Attributs für diese Seite. Übersetzt heißt das: `style.css` ist das Stylesheet für die aktuelle Seite.

Das `rel`-Attribut ist elementar für etliche Mikroformate, etwa `rel-tag`, `rel-nofollow` und XFN.

Rev

»Rev« ist die Kurzfassung für »umgekehrte Beziehung« (reversed). Dies ist das Gegenteil oder die Umkehrung des `rel`-Attributs. Der Wert des `rev`-Attributs ist die Beziehung zwischen der Ressource und der aktuellen Seite.

```
<link href="mailto:brian@suda.co.uk" rev="made"/>
```

Dies besagt, dass die aktuelle Seite das `rev`-Attribut ist, das vom `href`-Attribut-Wert `brian@suda.co.uk` »`made`« (erstellt) wurde. Übersetzt heißt das: Die Seite wurde von *brian@suda.co.uk* erstellt.

Der Unterschied zwischen `rev` und `rel` besteht darin, wie sich der `href`-Wert auf die aktuelle Seite bezieht. Im Fall von `rel` bezieht sich die aktuelle Seite auf `href`. Bei `rev` bezieht sich `href` auf die aktuelle Seite.

Ein weiteres Beispiel wäre:

```
<link href="/es/" rev="alternate"/>
```

Das `rev`-Atttribut besagt in diesem Fall, dass es zur aktuellen Seite ein »`alternate`«, also eine Alternative, gibt. In diesem Fall ist das eine spanische Übersetzung, die man unter »`/es/`« finden kann.

Die Unterschiede sind subtil, und in vielen Fällen wäre `rel` oder `rev` die korrekte Semantik. Der Wert »`alternate`« ist ein Beispiel, bei dem es egal ist, ob man `rel` oder `rev` verwendet. Die Aussage ist richtig, dass die Seite unter »`/es/`« ein »`alternate`« für die aktuelle Seite ist. Es ist aber auch richtig zu behaupten, dass die aktuelle Seite ein »`alternate`« für »`/es/`« ist. Solche Dinge tragen nicht dazu bei, die Verwirrung um diese wichtigen Attribute zu mindern.