



»... ein erstklassiges Lernerlebnis mit
gewaltigem Nutzen: Nicht verpassen!«
Louis F. Springer, Sun Microsystems

peter ARMSTRONG

FLEXIBLE RAILS

FLEX 3 AUF RAILS 2



 MANNING

HANSER





Inhalt

Vorwort	XV
Zum Geleit	XVII
Danksagungen	XIX
Über dieses Buch.....	XXI
Roadmap.....	XXI
Was dieses Buch nicht ist.....	XXIII
Ein Wort zu den Iterationen.....	XXIV
Welches Flex?	XXIV
Welches Rails?	XXV
Die Codebeispiele.....	XXV
Author Online.....	XXVI
Über den Autor.....	XXVI
Teil 1: Erste Schritte	1
1 Warum sind wir hier? Und wohin gehen wir?	3
1.1 Die Features und Stärken von Flex 3 und Rails 2.....	5
1.1.1 Überblick über Flex 3	5
1.1.2 Überblick über Rails 2	5
1.1.3 Überblick über die Kombination von Flex 3 und Rails 2.....	7
1.2 Flash 9? Soll das ein Witz sein?	8
1.3 Die Geschichte	9
1.4 Eine Vorschau auf das Buch.....	12
1.5 Zusammenfassung	13
2 Hello World.....	15
2.1 Die Installation	16
2.1.1 Ruby installieren.....	16
2.1.2 Rails installieren	17
2.1.3 MySQL installieren	18

2.1.4	Flex installieren	19
2.1.5	Editor oder IDE installieren (nur SDK-Benutzer)	22
2.1.6	Das Rails-Projekt erstellen	23
2.1.7	Lesehinweise für den Rest dieser Iteration	25
2.2	Windows oder Mac OS X + Flex Builder 3	25
2.2.1	Das Flex-Projekt erstellen.....	25
2.2.2	„Hello World“ von Flex	27
2.2.3	„Hello World“ von Rails	29
2.2.4	„Hello World“ von Flex und Rails	32
2.3	Windows + Flex SDK	33
2.3.1	Das Aptana RadRails-Projekt erstellen.....	33
2.3.2	„Hello World“ von Flex	35
2.3.3	„Hello World“ von Rails	39
2.3.4	„Hello World“ von Flex und Rails	42
2.4	Mac OS X (oder Linux) + Flex SDK	43
2.4.1	Das TextMate-Projekt erstellen (oder Emacs oder vi starten)	43
2.4.2	„Hello World“ von Flex	44
2.4.3	„Hello World“ von Rails	47
2.4.4	„Hello World“ von Flex und Rails	49
2.5	Zusammenfassung.....	51
3	Erste Schritte	53
3.1	Wenn Sie hier zu lesen beginnen.....	54
3.2	Einfrieren der Rails-Version	56
3.3	Integration der Browser-Navigation deaktivieren	56
3.4	Benutzeranmeldung in Rails hinzufügen.....	57
3.4.1	Installation und Ausführung von <i>restful_authentication</i>	58
3.4.2	CreateUsers-Migration bearbeiten, ausführen und Resultate prüfen	59
3.4.3	RESTful-Routen hinzufügen	62
3.4.4	Includes und <i>before_filter</i> anweisungsgemäß modifizieren.....	63
3.4.5	Anlegen von Benutzerkonten in HTML testen	66
3.4.6	Anmeldung in HTML testen.....	67
3.5	Benutzeranmeldung in Flex hinzufügen.....	68
3.5.1	„Hello World“, diesmal mit Bedeutung!	68
3.5.2	Bindung? Was ist das nun schon wieder...?	70
3.5.3	Dieses MXML sieht seltsam aus	71
3.5.4	Flex 3-Dokumentation? Wo denn?	72
3.5.5	Dummy-UI für Benutzerkonten und Anmeldung in Flex	73
3.5.6	Funktionalität für Benutzerkonten und Anmeldung.....	78
3.6	Den Test-Fixtures Daten hinzufügen.....	92
3.7	Die Tests überprüfen	94
3.8	Flex Builder für Ausführung und Debugging von pomodo konfigurieren	95
3.9	Zusammenfassung.....	97

Teil 2: Die Anwendung erstellen	99
4 Die Flex-Benutzeroberfläche	101
4.1 Anforderungen.....	101
4.2 Design	102
4.3 Code	103
4.4 Zusammenfassung	111
5 Rails mit REST	113
5.1 Eine kurze Notiz über REST	113
5.1.1 Disclaimer: den REST falsch machen.....	113
5.1.2 Was ist REST?.....	114
5.1.3 Warum RESTful??.....	116
5.2 Benutzer beim Namen rufen.....	117
5.2.1 Eine einfache Debug-Konsole für Flex.....	118
5.2.2 Die Sache mit dem fehlenden Vornamen.....	121
5.2.3 to_xml wird vorläufig repariert.....	124
5.2.4 to_xml wird endgültig repariert	125
5.3 Die neuen Ressourcen anlegen (einschließlich Migrationen, Modelle und Controller).....	128
5.3.1 Ressourcen für Aufgaben, Projekte und Orte erstellen	128
5.3.2 Das Modell mit Assoziationen versehen.....	130
5.3.3 Der TasksController.....	134
5.3.4 Format einstellen durch Routing.....	137
5.3.5 Ein RESTful-UsersController und -SessionsController	140
5.3.6 Migrationen bearbeiten und ausführen.....	143
5.3.7 Ad-hoc-Tests mit den HTML-Views	146
5.4 Sicherheit.....	150
5.4.1 POST für destruktive Aktionen	151
5.4.2 Anmeldung verlangen.....	151
5.4.3 Zugriffssteuerung.....	153
5.5 Fixtures erweitern und trotzdem Tests bestehen.....	166
5.6 Zusammenfassung	171
6 Flex auf Rails	173
6.1 Das Setup.....	173
6.2 Aufgaben in Flex auflisten	174
6.2.1 to_xml mit :include?	178
6.3 Aufgaben in Flex erstellen.....	180
6.4 Projekte und Orte in Flex erstellen und auflisten.....	184
6.5 Die ComboBoxen für Projekte und Orte in der TaskCreateBox zum Laufen bringen.....	191
6.5.1 Verlagerung der Listendaten durch Refactoring	191
6.5.2 Projekte und Orte anzeigen.....	196
6.5.3 Der ComboBox-Prompt	199
6.5.4 Ein None-Objekt für die ComboBox dataProvider	200
6.5.5 Ausgewählte Projekte und Orte speichern	202
6.6 Wenn Projekt oder Ort „None“ sind.....	205

6.7	Aufgaben, Projekte und Orte ändern und löschen	208
6.7.1	Die Methoden update_xml und destroy_xml den Rails-Controllern hinzufügen.....	208
6.7.2	ComboBox-itemRenderers in der TasksListBox implementieren	210
6.7.3	Löschen-Schaltflächen hinzufügen.....	220
6.7.4	Ausgefüllte Kontrollkästchen zur TasksListBox hinzufügen	222
6.7.5	Namen und Notizen von Aufgaben in der TasksListBox bearbeiten	225
6.7.6	Schaltflächen zum Löschen und für das Erledigt-Kontrollkästchen für ProjectsListBox und LocationsListBox.....	228
6.8	Funktionieren die Tests noch?.....	234
6.9	Zusammenfassung.....	234
6.10	Übungen für den Leser	234
7	Validierung.....	235
7.1	Der HTML-Anmeldebildschirm.....	236
7.2	Validierung in Rails und Flex mit DRY?	239
7.3	Rails-Validierung verstehen und eigene XML-Fehlermeldungen erstellen.....	240
7.4	Kurzüberblick über Validierung in Flex 3.....	246
7.5	Validierung von Rails und Flex 3 integrieren.....	248
7.6	Flex-Validatoren, die Zweite.....	255
7.7	Funktionieren die Tests noch?.....	261
7.8	Zusammenfassung.....	261
7.9	Übungen für den Leser	262
	Teil 3: Refactoring.....	263
8	Refactoring auf Cairngorm	265
8.1	Hintergrund und Setup	266
8.1.1	Die Geschichte von Cairngorm.....	266
8.1.2	Muss ich unbedingt Cairngorm verwenden?	267
8.1.3	Cairngorm 2.2.1 herunterladen	268
8.1.4	Die Cairngorm-Quellen in Flex Builder importieren	268
8.1.5	Der ModifiedCairngormStore.....	270
8.1.6	Cairngorm zu pomodo hinzufügen	271
8.1.7	Standardverzeichnisse anlegen	273
8.2	Überblick über die Event-Folge von Cairngorm	274
8.3	Der com.pomodo.model.PomodoModelLocator	275
8.4	com.pomodo.control.* erstellen	278
8.4.1	EventNames.as	278
8.4.2	PomodoController.as	279
8.5	CairngormUtils und ServiceUtils zu com.pomodo.util.* hinzufügen	280
8.5.1	CairngormUtils.as.....	280
8.5.2	ServiceUtils.as	281
8.6	com.pomodo.command.*	283
8.6.1	CreateTaskCommand.as	283
8.6.2	CreateProjectCommand.as.....	284
8.6.3	CreateLocationCommand.as.....	285

8.6.4	CreateSessionCommand.as	286
8.6.5	CreateUserCommand.as	287
8.6.6	UpdateTaskCommand.as	288
8.6.7	UpdateProjectCommand.as	289
8.6.8	UpdateLocationCommand.as	290
8.6.9	DestroyTaskCommand.as	291
8.6.10	DestroyProjectCommand.as	292
8.6.11	DestroyLocationCommand.as	293
8.6.12	ListTasksCommand.as	293
8.6.13	ListProjectsCommand.as	294
8.6.14	ListLocationsCommand.as	295
8.7	Erstellung von com.pomodo.business.*	296
8.7.1	TaskDelegate.as	296
8.7.2	ProjectDelegate.as	297
8.7.3	LocationDelegate.as	298
8.7.4	UserDelegate.as	299
8.7.5	SessionDelegate.as	300
8.8	Das Package com.pomodo.events löschen	300
8.9	Änderungen an com.pomodo.components.*	301
8.9.1	TaskCreateBox.mxml	301
8.9.2	ProjectCreateBox.mxml	303
8.9.3	LocationCreateBox.mxml	304
8.9.4	TasksListBox.mxml	305
8.9.5	ProjectsListBox.mxml	309
8.9.6	LocationsListBox.mxml	311
8.9.7	AccountCreateBox.mxml	313
8.9.8	LoginBox.mxml	315
8.9.9	MainBox.mxml	316
8.9.10	SplashBox.mxml	318
8.10	Anpassen von Pomodo.mxml an Cairngorm	319
8.11	pomodo ausführen	321
8.12	Fallstricke von HTTPService	321
8.13	Zusammenfassung	326
8.14	Übung für den Leser	326
9	Zustandsspeicherung auf dem Client	327
9.1	Refactoring auf Samurai-Art	328
9.2	Modellklassen erstellen	328
9.2.1	Task.as	329
9.2.2	Project.as	331
9.2.3	Location.as	333
9.2.4	User.as	335
9.3	Anpassen des PomodoModelLocator	336
9.4	Anpassen von ServiceUtils	340
9.5	Anpassen der Geschäftsdelegates	342
9.5.1	TaskDelegate.as	342

9.5.2	ProjectDelegate.as	343
9.5.3	LocationDelegate.as	344
9.5.4	SessionDelegate.as	346
9.5.5	UserDelegate.as	346
9.6	Anpassen der Befehle.....	347
9.6.1	CreateSessionCommand.as.....	347
9.6.2	CreateUserCommand.as	347
9.6.3	DestroyTaskCommand.as.....	348
9.6.4	ListLocationsCommand.as	349
9.6.5	ListProjectsCommand.as	349
9.6.6	ListTasksCommand.as.....	350
9.6.7	UpdateLocationCommand.as.....	350
9.6.8	UpdateProjectCommand.as	351
9.6.9	UpdateTaskCommand.as	351
9.7	Anpassen der Komponenten.....	352
9.7.1	MainBox.mxml.....	352
9.7.2	TaskCreateBox.mxml	353
9.7.3	ProjectCreateBox.mxml	354
9.7.4	LocationCreateBox.mxml.....	355
9.7.5	AccountCreateBox.mxml	355
9.7.6	TasksListBox.mxml.....	356
9.7.7	ProjectsListBox	363
9.7.8	LocationsListBox.mxml	366
9.8	Zusammenfassung.....	367
Teil 4: Schlussarbeiten.....		369
10	Fertigstellung der Anwendung.....	371
10.1	Notely.....	371
10.2	Mehr Sicherheit mit attr_accessible	383
10.3	GTD-Semantik, Next Action und :dependent	385
10.4	Aufgaben filtern	392
10.5	Die Kommandozeile.....	397
10.6	Abmeldung.....	398
10.7	Marketing!.....	400
10.8	Benutzer löschen	403
10.9	Übungen für den Leser	409
11	Refactoring auf RubyAMF.....	411
11.1	Warnung: Voreingenommener Autor.....	412
11.2	Hello RubyAMF.....	413
11.3	Umstellung auf RubyAMF im Schnelldurchlauf.....	418
11.3.1	Anpassen von rubyamf_config.rb.....	418
11.3.2	Anpassen der Rails-Controller.....	421
11.3.3	Services.mxml erstellen und Pomodo.mxml anpassen	430
11.3.4	Erstellen der Wertobjekte	432

11.3.5	Modellobjekte für die Erstellung von Wertobjekten modifizieren	433
11.3.6	Modifikation der Geschäftsdelegate	436
11.4	Anpassen des PomodoModelLocator	440
11.4.1	Anpassen der Befehle	443
11.5	Zusammenfassung	445
12	Rails on AIR (Adobe Integrated Runtime).....	447
12.1	pomodo wird zur AIR-Anwendung	447
12.1.1	Das alte Projekt löschen.....	448
12.1.2	Das neue Projekt erstellen.....	448
12.1.3	Das Projekt ausführen.....	451
12.2	Refactoring der Event-Auslösung.....	457
12.3	Online/Offline-Unterstützung.....	461
12.4	Zusammenfassung	468
12.5	Übungen für den Leser	468
12.6	Schlusswort	468
	Anhang A: Subversion mit Flex + Rails	471
	Anhang B: Themen, die nicht behandelt werden konnten	475
	Testen	475
	Laden mehrerer Dateien mit Flash, Flex, AIR und Rails oder Merb	476
	WebORB	477
	Register.....	479



1 Warum sind wir hier? Und wohin gehen wir?

*HTML sucks all the joy out of programming for me
HTML+CSS, that is
I'm so glad I don't have to do the design work for our apps
I'm trying to design a simple form
and I'm hating life
It's seriously making me want to not work on this anymore
...
html makes it so easy to write forms that look like crap
and SO HARD to write forms that look nice
that's so backwards*

– Jamis Buck, Signal vs. Noise [Fly on the Wall], 17. Juli 2007¹

Zurzeit gibt es eine Menge Aufregung um Flex und Rails, und auch ich werde in den folgenden Absätzen meinen Teil dazu beitragen.

Ruby on Rails, kurz: Rails, revolutioniert seit seiner Einführung im Jahre 2004 die Entwicklung von Webanwendungen. Mittlerweile scheint es, dass alle zehn Sekunden eine neue „Web 2.0“-Firma aus dem Boden schießt, die Rails verwendet.

Flex ist ein sexy Framework, in dem wir programmieren können, als sei es eine Desktop-Anwendung – nur dass diese im Flash Player läuft! Da Flex den Flash Player im Blick hat, können wir neue Rich Internet Applications (RIAs) erstellen, ohne uns um Browser-Kompatibilität und den ganzen Unsinn, JavaScript, CSS und so weiter, kümmern zu müssen.

¹ <http://www.37signals.com/svn/posts/495-fly-on-the-wall-paying-attention-to-users-mow-the-lawn-vscut-the-grass-chowder-html-forms>

Hinweis

Heutzutage scheint der Begriff „rich Internet applications“ in aller Munde zu sein, aber ich mag ihn nicht besonders, da die Abkürzung rla seltsam aussieht. Als milde Form des Protests (ich bin ein Flex-Developer der alten Schule) nenne ich sie in diesem Buch Rich Internet Applications. Die Großschreibung aller Wörter in „Rich Internet Applications“ könnte sogar wieder in Mode kommen: Nachdem ein Microsoft-Adept versucht hat, RIA für „Rich Interactive Applications“ einzuführen, bin ich vielleicht nur einer der Ersten, die kontern: „Runter von meinem Rasen!“

Da Flex 3 für eine bestimmte Plattform geschaffen wurde (nämlich Flash 9), müssen wir uns um die Kompatibilität von Plattformen keine Gedanken machen. Clientseitige Java-Programmierer träumten einmal von „Write Once, Run Anywhere“ (WORA), eine Hoffnung, die rasch zu „write once, debug everywhere“ zerstob. Dieser Traum kann nun endlich wahr werden, allerdings mit Flex. Flex schafft das, woran frühere Technologien, wie zum Beispiel Java-Applets, elend gescheitert sind: Anwendungen, die wie Desktop-Programme aussehen, aber in jedem modernen Webbrowser auf Windows und Mac laufen.

Hinweis

Write Once, Run Anywhere – einmal geschrieben, überall ausgeführt – wurde im Wesentlichen auf der Serverseite realisiert, aber nicht auf der Clientseite. Dort war AWT schon eine Katastrophe, und Swing sieht keiner der Plattformen ähnlich, auf denen es läuft. SWT ist eine hervorragende Alternative zu Swing, da es native Widgets zur Verfügung stellt, aber es kann noch nicht in einem Applet verwendet und somit auch nicht in einem Webbrowser ausgeführt werden. Es ist nur nützlich zur Erstellung von Anwendungen wie Eclipse – und wie der Flex Builder 3, der auf der Eclipse Rich Client Platform (RCP) aufsetzt.

Doch nun werde ich Ihnen ein Geheimnis verraten, das in diesem Buch zum ersten Mal gelüftet wird: Flex und Rails arbeiten erstaunlich gut zusammen!

Mit Flex 3 und Rails 2 lassen sich RIAs erstellen, die Web 3.0 mehr ähneln als die „achso-tollen“-Web 2.0-Sites, die sich heutzutage alle gegenseitig kopieren. Dieses Buch wird Ihnen zeigen, wie auch Sie dies bewerkstelligen können.

In dieser Iteration gebe ich einen Überblick über Flex und Rails, ihre Geschichte und ihre Kombinationsmöglichkeiten.

Iterationen

In diesem Buch heißen die Kapitel Iterationen. Diesen Begriff habe ich gewählt, weil wir im Laufe des Buchs iterativ eine Anwendung erstellen, und nicht etwa, weil dies etwas mit Flex oder Rails zu tun hat. (Es sei denn dies, dass Flex und Rails beide für den iterativen Programmierstil sehr gut geeignet sind.) Jede Iteration bringt die Anwendung einen Schritt weiter. Sie können mit dem Buch am Anfang jeder Iteration den Code vom Ende der jeweils vorhergehenden Iteration weiter verwenden. Nur dieses erste Kapitel enthält keinen Code, weil es lediglich eine Einführung ist. Ich nenne es nur deshalb „Iteration 1“ statt „Einführung“, damit Sie es lesen, denn viele Leser überspringen Einführungen, um gleich zum Kapitel 1 zu kommen. Ich liebe es, wenn ein Plan funktioniert ...

1.1 Die Features und Stärken von Flex 3 und Rails 2

Jetzt sind Sie bestimmt schon ganz gespannt, also wollen wir tief Luft holen und beide Plattformen kurz vorstellen. Dieser Abschnitt kann nur einen sehr groben Überblick über Rails und Flex geben und dann zeigen, wie sich beides kombinieren lässt. Keine Sorge, wenn Sie hier etwas nicht verstehen, die Erklärung wird später nachgereicht.

1.1.1 Überblick über Flex 3

Mit Flex 3 erstellen Sie Code in MXML (XML-Dateien mit der Erweiterung `.mxml`, wobei das M für Macromedia steht) und ActionScript (Textdateien mit der Erweiterung `.as`) und kompilieren diese in eine SWF-Datei, die im Flash-Player läuft. Diese SWF-Datei wird von einer HTML-Datei referenziert, sodass ein Benutzer, der die HTML-Datei in einen modernen Webbrowser lädt, den Flash-Film abspielt. Vorher wird der Benutzer aufgefordert, Flash 9 herunterzuladen, wenn es auf seinem Computer noch nicht installiert ist. Die SWF kann mit der Webseite, in die sie eingebunden ist, und mit dem Server, von dem sie gesendet wurde, interagieren.

Selbst wenn Sie noch nie in Ihrem Leben einen Flash-Film erstellt haben, sich selbst nicht als Designer bezeichnen würden und die Timeline selbst dann nicht erkennen würden, wenn sie direkt vor Ihrer Nase läge, können Sie mit Flex attraktive Anwendungen erstellen, die im Flash-Player laufen. Flex-Entwicklung kann jeder in der Programmierung von Web- oder Desktop-Oberflächen (zum Beispiel mit Windows Forms oder Java Swing) durchschnittlich erfahrene Entwickler erlernen.

1.1.2 Überblick über Rails 2

Abbildung 1.1 zeigt, dass Rails eine standardmäßige Drei-Schichten-Architektur (Präsentationsschicht, Modellschicht, Datenschicht) sowie das Model-View-Controller (MVC)-Muster zur Verfügung stellt. Wie im Diagramm zu sehen ist, kümmert sich Rails um alles, was zwischen dem Webserver und der Datenbank abläuft.

Der typische Ablauf ist wie folgt:

1. Ein Benutzer besucht einen bestimmten URL mit seinem Browser (sendet einen HTTP-Request).
2. Der Request wird über das Internet dem Webserver übermittelt, auf dem Rails ausgeführt wird (zum Beispiel WEBrick, lighttpd, Mongrel oder Apache).
3. Dieser Webserver übergibt den Request an den Routing-Code in Rails, das heißt an `ActionController::Routing::Routes`. Diese Routen sind in `config/routes.rb` definiert. Die Standardroute macht aus den HTTP-Requests Methodenaufrufe auf Controllern.

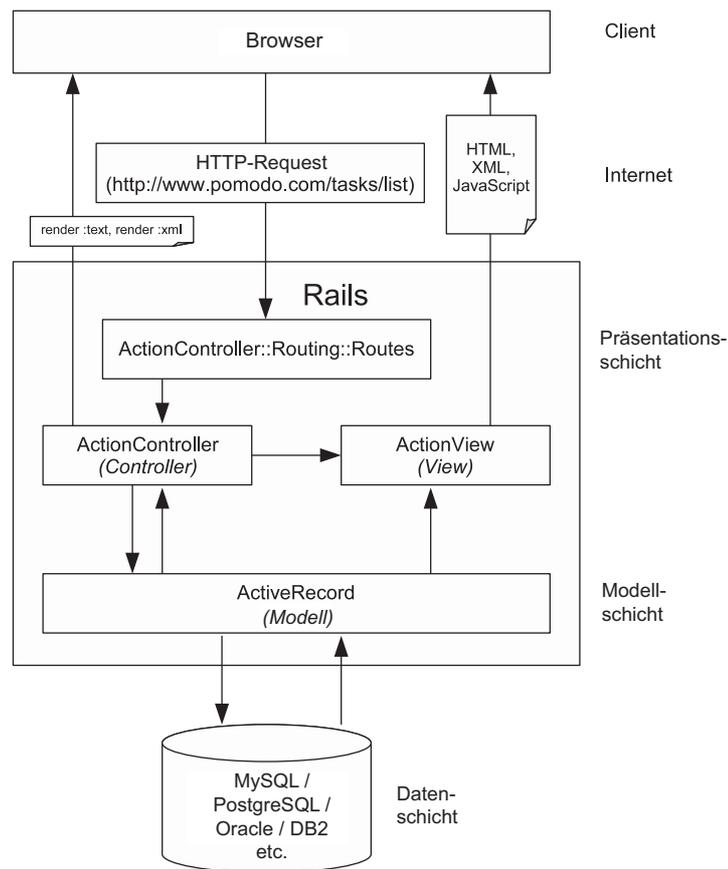


Abbildung 1.1 Rails stellt eine standardmäßige Drei-Schichten-Architektur (Präsentationsschicht, Modellschicht, Datenschicht) sowie das Model-View-Controller (MVC)-Muster zur Verfügung.

- Vom Controller (beispielsweise `TasksController`) wird eine Methode (beispielsweise `index`) aufgerufen. Diese kommuniziert mit diversen ActiveRecord-Modellen (die in einer Datenbank Ihrer Wahl gespeichert und von ihr abgerufen werden). Sodann hat die Controller-Methode folgende Möglichkeiten:
 - Sie kann einige Instanzvariablen setzen und ein View-Template (etwa eine speziell benannte `.html.erb`-Datei) verwenden lassen, um HTML, XML oder JavaScript zu erstellen, das dann an den Browser gesendet wird. Dies ist die Aufgabe von Action View. Action View und Action Controller bilden zusammen das Action Pack.
 - Sie kann den View-Mechanismus umgehen und die Wiedergabe direkt durch einen Aufruf der Rendering-Methode ermöglichen. Diese Methode kann einfachen Text (`render :text => "foo"`), XML (`render :xml => @task`) und so weiter produzieren.

1.1.3 Überblick über die Kombination von Flex 3 und Rails 2

Abbildung 1.2² zeigt, wie Flex und Rails zusammen benutzt werden können.

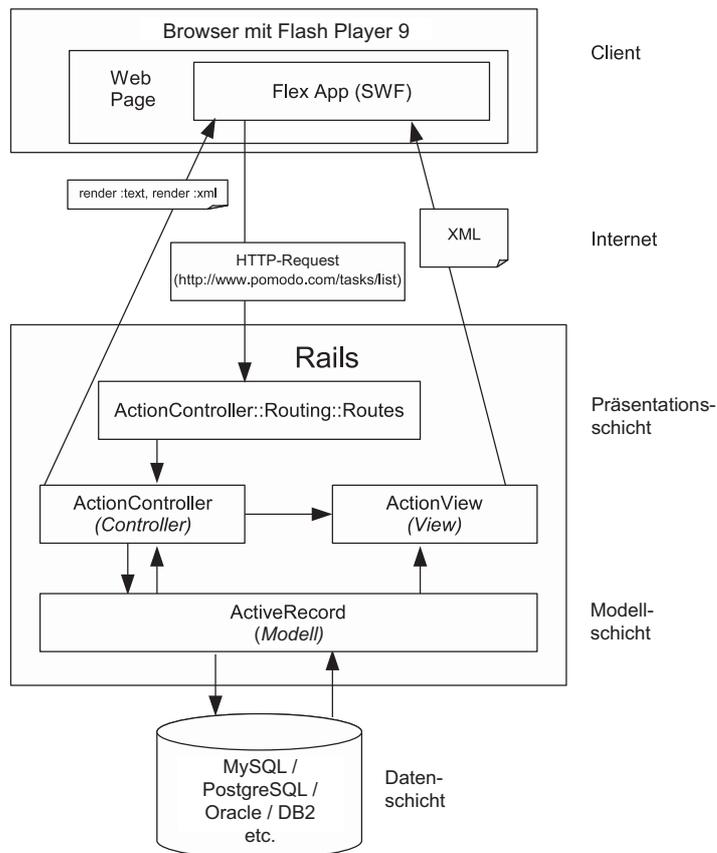


Abbildung 1.2 Wie man Flex und Rails zusammen verwendet.

² Das Diagramm "Fig 7. The complete MVC Architecture for a Thin-Client Web Application" unter <http://www.uidesign.net/Articles/Papers/UsingMVCPatterninWebInter.html> hat das Design meiner Blockdiagramme inspiriert.



Register

Numerale

200 OK 114
3. Profit!!! 190
37signals 408
37signals 13
38noises 103

A

Abmeldung, Unterstützung 398
Accordion 77, 103
AccountBox, Benutzer löschen 406
AccountCreateBox 74, 77, 80, 142
 adding client-side validation 256
 anpassen 355
 anpassen an Cairngorm 313
 anpassen an User-Konstruktor 381
 EmailValidator 246
 Validierung von Flex 3 und Rails integrieren 251
AccountCreateEvent 82
AcknowledgeMessage 418
ActionMailer 59
Active Record, Definition 130
Active Support 133
 dasherize 133
ActiveRecord
 Base 390
 Errors 242
 Migration 60
 Serialization 127
Adobe 267
Adobe Flash Player 8 *siehe Flash 9*
Adobe Integrated Runtime (AIR) 447
Adobe Labs 267
AdvancedDataGrid 103
Agile WebDevelopment with Rails 17
 siehe AWDwR
Alert 91
Amazon S3 382
AMF 327
Anmeldebildschirm 236
Anmeldung
 Testen in HTML 67
 verlangen 151
Anmeldung 57 *siehe Benutzeranmeldung*
anonymes Objekt
 ActionScript 3 84
 erstellen 84
Apollo 447
Application 69
application.rb, hinzufügen zu_xml_full method 243
application/xml 322
application/x-www-form-urlencoded 84, 322, 84, 322
ApplicationController 64, 156
 Schutz gegen CSRF-Attacken 165
Aptana RadRails, Projekt erstellen 33
Architektur 6
Arctic Monkeys 371
Arkin, Assaf 476
Armstrong, Peter XXVI

Array, Konvertierungen 127
 ArrayCollection 340
Arrington, Michael 76
 Assistent-generierter Code 111
 Assoziation hinzufügen zum Modell 130
 AsyncToken 283
 attachment_fu 476
 attr_accessible 62
 Sicherheit 383
 attr_protected 384
 Aufgaben
 ändern und löschen 208
 auflisten in Flex 174
 erstellen in Flex 180
 filtern 392
 Namen in TaskListBox bearbeiten 225
 authenticated? 133
 AuthenticatedSystem 64, 156
 AuthenticatedTestHelper 64
 Authentifizierung 57
 AWDwR 17

B

backgroundGradientColors 69
 Bailey, Chris 386
 Beast 476
Beethoven, Ludwig van 198
 before_create 376
Bekor, Chen 270
 belongs_to 130, 132
 Benutzer
 erstellen 57
 löschen 403
 Benutzeranmeldung 57
 hinzufügen zu Flex 68
 hinzufügen in Rails 57
 restful_authentication 58
 Benutzerkonten anlegen, Testen in HTML 66
Bezos, Jeff 76
 Bindable 70, 76
 Bindung 70
Black, David 374
Bland II, John C. 461

Blub 136
 BPM 469
 Browser-Navigation, Integration in Flex
 ausschalten 56
 bubbles-Property 90
Buck, Jamis 389
 Builder 175
 verstehen 244
 XmlMarkup 243, 244
 Business, Services 432
 Button 71
 Buzzword 371

C

Cairngorm
 Befehle 398
 Dokumentation 267
 Event-Folge 274
 Generator 378
 Geschichte 266
 herunterladen 268
 Hintergrund und Setup 266
 Java-Ähnlichkeit 267
 ModifiedCairngormStore 270
 Notwendigkeit 267
 pomodo 265, 271
 Quellen, in Flex Builder importieren 268
 Standardverzeichnisse 273
 Vorteile 318
 Wertobjekt 328
 Cairngorm Enterprise 268
 CairngormStore 268
 CairngormUtils 280
Carlson, Bryan 414
 CDATA 76
 CheckBox 107
 erstellen 222
 checkIsOnline 466
 Clark, Mike 476
 com.pomodo.command.* erstellen 283
 com.pomodo.components.* ändern 301
 com.pomodo.control.* erstellen 278
 com.pomodo.events löschen 300

- ComboBox 103, 107
 - dataProvider, None-Objekt 200
 - in TaskCreateBox 191
 - itemRenderers 210
 - Prompt 199
 - CommandShell implementieren 397
 - config, routes.rb 62
 - ControlBar 109
 - create method 135
 - CreateLocations 144
 - CreateLocationCommand 285
 - create-Methode 180
 - CreateProjects 144
 - CreateProjectCommand 284
 - CreateSessionCommand 286
 - anpassen 347
 - Refactoring der Event-Auslösung 460
 - CreateTasks 143
 - CreateTaskCommand 283
 - CreateUsers 60
 - CreateUserCommand 287
 - anpassen 347
 - Refactoring der Event-Auslösung 460
 - creationPolicy 457
 - CRUD 113
 - CSRF attack 64
 - CSRF-Attacken, Schutz vor 165
 - curl 414
 - current_user 79, 156
 - current_user.tasks 156
 - Curtis, Ben 476
- D**
- Daigle, Ryan* 371
 - dasherize 123
 - DataGrid 103
 - Zellen mit itemRenderer anpassen 210
 - Date 119
 - Datenschutz, Verstoß, E-Mail-Validierung 253
 - Datentransferobjekt *siehe Transferobjekt*
 - Dawson, Alastair* 477
 - DebugMessage 119
 - DebugPanel 119
 - default route 67
 - dependent implementieren 385
 - destroy method 135
 - destroy_xml 208
 - DestroyLocationCommand 293
 - DestroyProjectCommand 292
 - DestroyTaskCommand 291
 - anpassen 348
 - Umstellung auf RubyAMF 443
 - DestroyUserCommand, Benutzer löschen 404
 - Destruktive Aktionen, POST 151
 - Developing Rich Clients with Macromedia Flex 267
 - development.log 245
 - DHH* 10, 175
 - Discovering a world of Resources on Rails 58, 116
 - displayAsPassword 75
 - Dont Repeat Yourself (DRY) 181
 - Flex-Validierung und 239
 - Drei-Schichten-Architektur 6
- E**
- E4X (ECMAScript für XML) 81, 122
 - Eccles, Stuart* XV, 216
 - edit method 135
 - EmailValidator 246
 - encrypt_password 133
 - endpoint 415
 - ERb (Embedded Ruby) 136
 - Erledigt-Kontrollkästchen erstellen 228
 - <error> 241
 - error_messages_for 238
 - <errors> 241
 - Event 82
 - bubbles 90
 - EventNames 278
 - Anpassen für Abmeldung 398
 - Benutzer löschen 404
 - Konstanten hinzufügen 380
 - Events auslösen, Refactoring 457

F

Fade 402
 fdb-Debugger 96, 118
 Fehlermeldungen, eigene, XML 240
Fielding, Roy 115
 Fight Club 136
 filterFunction 392
 Fireworks 103
 Fixnum 133
 Fixtures 92

- erweitern 166
- notes.yml 375

 flash[:notice], notice 161
 Flash 9 8
 flash.events.Event 83
 Flash-UI, Kritik wegen fester Größe

- unbegründet 103

 Flex

- API-Dokumentation 72
- Aufgaben auflisten 174
- Aufgaben erstellen 180
- benötigte Version XXIV
- Benutzeranmeldung hinzufügen 68
- Compiler-Argumente 450
- Hello World 27, 35, 44
- Installation 19
- Projekt erstellen 25
- Projekte und Orte erstellen 184
- Überblick 5
- und Rails 7, 10
- Validatoren 255
- Validierung 239
- Versionen 11

 Flex 3

- Dokumentation 72
- Validierung 246
- Validierung, Integration mit Rails 248, 255

 Flex Builder

- Compiler-Einstellungen 400
- Debugger 95, 118
- Design-Modus 102
- konfigurieren für pomodo 95

Visuelles Design 104
 Flex Builder 3 25

- AIR 447

 Flex SDK, fdb-Debugger 96
 Flex und Rails

- Geschichte 9
- Hello World 32, 42, 49

 FlexUnit 475
 Form 107

- Layout-Tool 74

 format.amf 412
Fowler, Martin 130
 Fremdschlüssel-Constraint 132
 FrontController 279
 full_messages 243

G

geschachtelte Ressource 373
 Geschäftsdelegate erstellen 296
 Geschichte Flex und Rails 9
 GET 70
 getRemoteURL 454
 Getting Things Done (GTD) 12, 101

- Semantik 385

Gilbert, W. S. 469
 Git 95
Goldberg, Rube 266
 GPL 413
Grosenbach, Geoffrey 58, 64, 117
 GTD 12
 GUI

- Anforderungen 101
- Code 103
- Design 102

H

Hansson, David Heinemeier 113
 has_many 130, 132
 Hash

- Konvertierungen 126
- Ruby 84

 HBox 105
Heinemeier Hansson, David 10 *siehe DHH*

- HelloController 69
 - .html.erb 179
 - HTTP 209
 - Methoden 70
 - Statuscodes 114
 - vorgebliche und tatsächliche Methode 116
 - HTTPService 69, 411
 - contentType 321
 - Fallstricke 321
 - human_attribute_name 243
- I**
- ICollectionView 340
 - ICommand 284
 - id, Primärschlüssel 61
 - Idempotenz 113
 - IList 340
 - Image 77
 - Import, Projekte in Workspace 54
 - include 178
 - index 135
 - Inflections 133
 - Installation 16
 - Interface vs. Klasse 340
 - itemRenderer
 - Anpassung von DataGrid 210
 - benutzerdefinierter, für CheckBox 222
 - data-Property 219
 - Iterationen 4
 - IUndoableCommand 326
 - IViewCursor 216
- J**
- Java Swing 326
 - JBuilder, GUI-Builder 102
 - JSON 263, 327
 - JSplitPane 109
 - JUnit 475
- K**
- Kernighan, Brian* 15
 - Knauss, Greg* 173
 - Kontrollkästchen 222 *siehe CheckBox*
 - Konvention geht vor Konfiguration 61
 - Kopieren-Einfügen-Ändern *siehe MXML*
- L**
- Label 77
 - labelWidth 78
 - Library Path 272
 - Lighthouse 476
 - LinkButton 408
 - Linux, Flex SDK installieren 20
 - ListCollectionView 340
 - filtern 392
 - Listendaten verlagern, Refactoring 191
 - ListLocationsCommand 295
 - Anpassen 349
 - refactoring to RubyAMF 444
 - ListProjectsCommand 294
 - anpassen 349
 - Umstellung auf RubyAMF 443
 - ListTasksCommand 293
 - Anpassen 350
 - Umstellung auf RubyAMF 443
 - Live Cycle Data Services 11, 270
 - LoadURLCommand 398
 - Location
 - anpassen an GTD-Semantik 386
 - anpassen für Wertobjekte 435
 - erstellen 128, 333
 - Sicherheit verbessern 384
 - LocationCreateBox 188
 - anpassen 355
 - anpassen an Cairngorm 304
 - erstellen 107
 - locationCreate-Event 189
 - LocationDelegate 298
 - anpassen 344
 - Umstellung auf RubyAMF 438
 - LocationEvent 187
 - Unterstützung für Update und Delete 228
 - locations.yml 166
 - LocationsController
 - reparieren 161
 - Umstellen auf RubyAMF 426

- update und destroy 209
- LocationsControllerTest 169
- LocationsListBox
 - Anpassen 366
 - Anpassen an Cairngorm 311
 - creating and listing locations 185
 - erstellen 110
 - Löschen-Schaltflächen und Erledigt-Kontrollkästchen 231
 - Verlagerung der Listendaten durch Refactoring 195
- LocationVO 433
- Login 57 *siehe Benutzeranmeldung*
- login_required 151
- LoginBox 75, 77, 83, 142
 - Anpassen an Cairngorm 315
- LoginEvent, erstellen 84
- Löschen-Schaltfläche erstellen 220, 228

M

- MacCaw, Alex* 383, 476, 383
- Macromedia 267
- Macromedia Flex *siehe* Flex 10
- MainBox 183, 189, 183, 189
 - Anpassen 352
 - Anpassen an Cairngorm 316
 - Anpassen für Abmeldung 399
 - Benutzer löschen 407
 - ComboBox-itemRenderer in TasksListBox 211
 - erstellen 86
 - GUI code 104
 - modifizieren für neue Events 233
 - None-Objekt 200
 - Notizen hinzufügen 381
 - Projekte und Orte anzeigen 198
 - Refactoring der Event-Auslösung 458
 - Verlagerung der Listendaten durch Refactoring 191
- Malik, Om* 76
- map.resources 138, 373
- map.root 138
- Marketing 400

- McLeod, Alistair* 266
- Mehrfachvererbung, Nachteile 156
- Mephisto 476
- Merb 476
- Messerjongleur 383
- Metadata 82, 86
- Metal 266
- method, Parameter für ServiceUtils 297
- method_missing 245
- Methoden
 - class vs. instance 132
 - Zugriff einschränken 383
- Migrationen 144
 - bearbeiten und ausführen 143
- Mikroarchitektur 267
- MIT-Lizenz 411
- Mixin 156
- Modellklassen erstellen 328
- ModelLocator 275, 352, 275, 352
- Model-View-Controller (MVC) 5, 265
- ModifiedCairngormStore 270
- MVC 5
- mx, Komponente 218
- MXML 10
 - benutzerdefinierte Komponente 219
 - vs. html.erb 179
- MXML 5
- MySQL, Installation 18

N

- Napkin Look & Feel 266
- new method 135
- new.rhtml 66, 237, 66, 237
- Next Action 383
 - Definition 103
 - implementieren 385
- None-Objekt 205
 - ComboBox dataProvider 200
- Note 373, 376, 373, 376
 - Anpassen für Wertobjekte 435
 - Sicherheit verbessern 385
- NoteDelegate 378
 - Umstellung auf RubyAMF 439

- Notely 371
 - erstellen 111
 - Online/Offline-Unterstützung 463
 - Registerkarte anpassen 381
- notes.yml 375
- NotesController 374
 - Umstellen auf RubyAMF 428
- NotesHelper 375
- NoteVO 433
- Notizen bearbeiten in TasksListBox 225
- Nullobjekt-Muster 330
- Number 82
- O**
- objektrelationale Abbildung 130
- OfflineMainBox 461
- Olson, Rick* 58, 476, 58
- onlineCheckURL 462
- ORM *siehe objektrelationale Abbildung*
- Orte
 - ändern und löschen 208
 - Auswahl speichern 202
 - erstellen und auflisten in Flex 184
- OS X
 - Flex installieren 19
 - Flex SDK installieren 20
 - Ruby installieren 17
 - TextMate installieren 22
 - und Flex Builder 3 25
- outerDocument 219
- Outlook 101
- P**
- Package, Backward Domain Names-Konvention 73
- paddingLeft 89
- ParameterMappings.scaffolding 420
- PasswordConfirmationValidator 255
- PeepCode 58
- PeepCode RESTful Rails 116
- Photoshop 103
- Polymorphismus 340
- pomodo, Umstellung auf AIR 447
- Pomodo 12, 120
 - Online/Offline-Unterstützung 462
 - StatusBox hinzufügen 455
- Pomodo.mxml 69, 75
 - Anpassen an Cairngorm 319
 - Marketing 400
 - RubyAMF-Hello World 415
 - Umstellung auf RubyAMF 431
- PomodoController 279
 - Anpassen für Abmeldung 399
 - Befehle hinzufügen 380
 - Benutzer löschen 404
- PomodoModelLocator
 - anpassen 336
 - Eingabemöglichkeit für Stamm-URL 453
 - erstellen 275
 - Note-Klasse hinzufügen 377
 - Online/Offline-Unterstützung 462
 - Refactoring der Event-Auslösung 459
 - Umstellung auf RubyAMF 440
- POST 70
- Pragmatic Programmer, The 181
- prevent_access 160
- Principle of Least Surprise 123
- Programming Ruby 245
- Project
 - Anpassen an GTD-Semantik 385
 - Anpassen für Wertobjekte 434
 - erstellen 331
 - Sicherheit verbessern 384
- ProjectCreateBox 187
 - Anpassen 354
 - Anpassen an Cairngorm 303
 - erstellen 107
- projectCreate-Event 189
- ProjectDelegate 297
 - Anpassen 343
 - Umstellung auf RubyAMF 437
- ProjectEvent 186
 - Unterstützung für Update und Delete 228
- Project-Ressource erstellen 128
- projects.yml 166
- ProjectsController

- Anpassen an GTD-Semantik 391
 - reparieren 161
 - Umstellung auf RubyAMF 424
 - update und destroy 209
 - ProjectsControllerTest 170
 - ProjectsListBox
 - Anpassen 363
 - Anpassen an Cairngorm 309
 - creating 109
 - Löschen-Schaltflächen und Erledigt-Kontrollkästchen 228
 - Projekte erstellen und auflisten 184
 - Verlagerung der Listendaten durch Refactoring 194
 - ProjectVO 433
 - Projekte
 - ändern und löschen 208
 - Auswahl speichern 202
 - erstellen und auflisten in Flex 184
 - protect_from_forgery 64
 - protected 133
- R**
- Rails 3
 - Ablauf 5
 - Architektur 6
 - benötigte Version XXV
 - Benutzeranmeldung hinzufügen 57
 - Hello World 29, 39, 47
 - Installation 17
 - MIT-Lizenz 412
 - Projekt erstellen 23
 - rel_2-0-0_RC1 56
 - Überblick 5
 - und Flex 7
 - Validierung 239, 240
 - Validierung, Integration mit Flex 3 248, 255
 - Version einfrieren 56
 - vs. Flex 10
 - XML 10
 - Rails Engines 58
 - Rails on AIR 469
 - Rails Routing (Digital Shortcut) 374
 - Rails to Italy 326, 418
 - rake 94, 169
 - rake db:fixtures:load 145
 - rake db:migrate 144
 - rake routes 139, 374
 - Raymond, Scott* 113, 115
 - RecordNotFound 160
 - Refactoring
 - Listendaten verlagern 191
 - RubyAMF 418
 - Samurai-Art 328
 - RemoteClass 432
 - RemoteObject 411, 415
 - remoteURL 462
 - render, :amf 411, 412
 - request 82
 - required, visueller Hinweis auf FormItem 75
 - rescue 160
 - respond_to 113
 - Ressourcenorientierte Architektur 114
 - REST (Representational State Transfer) 57
 - Einführung 114
 - RESTful, Gründe für 116
 - RESTful Authentication 57
 - restful_authentication 53
 - installieren und ausführen 58
 - RESTful-Controller 135
 - RESTful-Entwurf, Nomen und Verben 116
 - RESTful-Route hinzufügen 62
 - RESTful-Verfahren, Gründe für 118
 - RESTful Web Services 114
 - ResultEvent 84
 - RIA 4
 - Rich Internet Applications 4, 175 *siehe RIA*
 - Richardson, Leonard* 114
 - RJS 136
 - ROA 114
 - routes.rb 137, 373
 - Routing 69
 - Format einstellen 137
 - RSpec 475
 - Ruby, Installation 16
 - Ruby for Rails 133

- Ruby on Rails 3, 9
- Ruby, Sam* 114
- RubyAMF 327, 411
 - flexiblerails-Branch 412
 - Hello, World 413
 - MIT-Lizenz schließt WebORB aus 413
 - params[0] 423
 - RESTful-Controller-Integration 412
 - Typkonvertierungen 421
 - Umstellung auf 418
- rubyamf_config.rb 418

- S**
- Samurai-Coder 328
- save_with_gtd_rules! 386
- sayhello 69
- scaffold-Befehl 128
- scaffold_resource-Befehl 128
- Scaffolding 128
- schema_info 61
- Schreibblockade 136
- Script, mx: 76
- security through obscurity 150
- selectedChild 86
- self 132
- ServerErrors 249
- ServerErrorValidator 249, 250
- ServiceLocator 430, 432
- services, Compiler-Argument 416
- Services.mxml 411, 430
- services-config.xml 415
- ServiceUtils 281, 322
 - anpassen 340
 - Anpassen von URLs 455
 - send 296
- SessionDelegate 300
 - Anpassen 346
- sessions 156
- SessionsController 63, 68, 78, 152
 - Anpassen für Abmeldung 400
 - making RESTful 141
- SessionsControllerTest 64
- Shaw, Zed* 418
- show 135
- ShowNoteCommand 379
 - Online/Offline-Unterstützung 464
 - Umstellung auf RubyAMF 444
- Sicherheit 150
 - attr_accessible 383
- Silicon Valley* 469
- Singleton 430
- Skinny Controller, Fat Model 389
- skip_before_filter 152
- slice(0) 340
- Softwareinstallation 15
- Spacer 77
- SplashBox 88
 - anpassen an Cairngorm 318
 - anpassen an größere AccountCreateBox 259
- StatusBox 454
 - Online/Offline-Unterstützung 465
- String 71
- Subclipse 95
- Subversion 95
- sudo 59
- summaryPanel 227
- Swing 103, 469
- Symbol 62
- Syntaktischer Zucker 136

- T**
- TabNavigator 103, 109
- Ta-da List 13
- Task
 - anpassen an GTD-Semantik 387
 - anpassen für Wertobjekte 433
 - erstellen 329–331
 - Sicherheit verbessern 384
- TASK_CREATE 182
- TaskCreateBox 200
 - anpassen 353
 - anpassen an Cairngorm 301
 - ComboBox-Prompt hinzufügen 199
 - erstellen 106
 - Erstellen von Aufgaben 182
 - Projekte und Orte anzeigen 196

- Speichern der Auswahl 203
 - taskCreate-Event 183
 - TaskDelegate 296
 - anpassen 342
 - Umstellung auf RubyAMF 436
 - TaskEvent 181, 182
 - Task-Ressource erstellen 128
 - tasks.yml 146, 167
 - TasksController 134, 155
 - Anpassen an GTD-Semantik 389
 - für Zuordnungsproxies modifizieren 158
 - Umstellen auf RubyAMF 421
 - update und destroy 208
 - TasksControllerTest 170
 - TasksListBox 217, 220
 - anpassen 356
 - anpassen an Cairngorm 305
 - Aufgaben auflisten 175
 - Aufgaben filtern 393
 - ComboBox-itemRenderers in TasksListBox 214
 - Completed CheckBoxes hinzufügen 223
 - creating 107
 - Namen und Notizen von Aufgaben bearbeiten 225
 - Verlagerung der Listendaten durch Refactoring 194
 - TaskVO 432
 - test/spec 475
 - test/unit 475
 - Tests
 - Ad-hoc, mit HTML-Views 146
 - simulierte 65
 - überprüfen 94
 - Testdaten laden 92
 - Text 77
 - TextArea 109
 - TextInput 70, 109, 70, 109
 - TextMate 451
 - Projekt erstellen 43
 - Thomas, Dave* 418
 - to_xml 123
 - fixing 125–128
 - to_xml_full 241, 249
 - To-Do-Liste 12
 - Transferobjekt 328
 - Tree 103
 - TypeError 457
- ## U
- unsavedNoteContent 462, 463
 - Unterstützung Online/Offline 461
 - update 135
 - update_attributes 383
 - update_xml 208
 - UpdateLocationCommand 290
 - anpassen 350
 - updateLocationIdMap 214
 - UpdateNoteCommand 380
 - UpdateProjectCommand 289
 - anpassen 351
 - Aufgaben auflisten 392
 - UpdateTaskCommand 288
 - anpassen 351
 - Projekte auflisten 391
 - Umstellung auf RubyAMF 444
 - User 131, 157, 238, 373, 376
 - Anpassen an GTD-Semantik 387
 - erstellen 335
 - ID-Feld hinzufügen 377
 - @user.errors.to_xml 241
 - UserDelegate 299
 - Anpassen 346
 - Benutzer löschen 405
 - UsersController 63, 66, 79, 153, 236
 - adding logger.info statement 245
 - Benutzer löschen 403
 - making RESTful 140
 - Validierung von Flex 3 und Rails integrieren 248
 - UsersControllerTest 64
 - UserTest 64
 - Using Flex Builder 102

V

validate 239
validate_on_create 239
validate_on_update 239
validateAll 258
validates_confirmation_of 239
validates_length_of 239
validates_uniqueness_of 239
Validator, Flex 255
Validierung
 Flex 3 246
 Framework in UI 235
 in Rails and Flex 239
 Rails 240
 Rails, integrieren in Flex 3 248
Validierungs-Hilfsklassen 238
VBox 74, 105
VDividedBox 109
ViewStack 86
visible-Property 109
Visual Basic, GUI-Builder 102
Visual C#, GUI-Builder 102
Vixiom Axioms 477

W

Web 2.0 12, 173
WebORB 327, 477
Webster, Steven 266, 327
Welcome Aboard 67
Wertobjekt 328, 418
Wertobjekte erstellen 432

WindowedApplication 452
Windows
 Aptana, RadRails installieren 22
 Flex installieren 19
 und Flex Builder 3 25
 Flex SDK installieren 20, 33
 Ruby installieren 16
Windows Forms 327
WipeUp 402
Wischusen, Derek 477
workflowState 287, 288, 320
WS-Death Star 113

X

XML
 Ergebnis in XML umwandeln 82
 in Flex 105
 Namensraum 69
 nativer Typ in ActionScript 3 82
XMLListCollection 340
XMLUtils 222

Y

YAML 9, 327, 9, 327
Yegge, Steve 101, 136

Z

Zen 101
Zugriffssteuerung 153
Zygmuntowicz, Ezra 476