

Inhaltsverzeichnis

1	Einleitung	1
1.1	Für wen dieses Buch bestimmt ist	2
1.2	Organisation des Buches	3
1.3	Web-Site zum Buch	5
1.4	Feedback	5
2	Überblick und Installation	7
2.1	Was ist Ruby on Rails?	7
2.2	Bestandteile von Rails	10
2.2.1	Komponenten und Zusammenspiel	10
2.2.2	Action Pack	11
2.2.3	Active Record	11
2.2.4	Action Mailer	12
2.2.5	Ajax	12
2.2.6	Unit Tests	12
2.3	Installation	12
2.3.1	Windows	12
2.3.2	Mac	13
2.3.3	Linux	13
2.3.4	Aktualisierung	13
2.3.5	Datenbank	13
2.3.6	Glückwunsch! Willkommen an Bord!	13
3	Hands-on Rails	15
3.1	Entwicklungsphilosophie	16
3.2	Domain-Modell	16
3.3	OnTrack Product Backlog	17
3.4	Aufsetzen der Infrastruktur	18
3.5	Projekte erfassen, bearbeiten und löschen	20

3.5.1	Modell erzeugen	20
3.5.2	Datenbankmigration	22
3.5.3	Controller	26
3.6	Iterationen hinzufügen	27
3.7	Zwischenstand	34
3.8	Iterationen anzeigen	34
3.9	Iterationen bearbeiten und löschen	36
3.10	Tasks hinzufügen	39
3.11	Tasks anzeigen, bearbeiten und löschen	42
3.12	Struktur in die Seiten bringen	44
3.13	Validierung	45
3.14	Benutzerverwaltung	49
3.15	Login	51
3.16	Tasks zuweisen	54
3.17	Endstand und Ausblick	55
4	Active Record	57
4.1	Active Record-Klassen	57
4.1.1	Mehr über Modellattribute	59
4.1.2	Mehr über Primärschlüssel	60
4.1.3	Zusammengesetzte Primärschlüssel	61
4.1.4	Mehr über Tabellennamen	61
4.2	Active Record direkt verwenden	63
4.3	Die Rails-Konsole	63
4.4	Objekte erzeugen, laden, aktualisieren und löschen	64
4.4.1	Erzeugung	65
4.4.2	Objekte laden	65
4.4.3	Objekte aktualisieren	66
4.4.4	Objekte löschen	68
4.5	Mehr über Finder	68
4.5.1	Suchbedingungen: conditions	68
4.5.2	SQL-Injection vermeiden	70
4.5.3	Ordnung schaffen: order	71
4.5.4	Limitieren: limit	71
4.5.5	Seitenweise: limit und offset	71
4.5.6	Weitere Parameter: joins und include	72
4.6	Dynamische Finder	73

4.7	Kann ich weiterhin SQL verwenden?	74
4.8	Metadaten – Daten über Daten	75
4.9	Assoziationen	76
4.9.1	Grundsätzliches	76
4.9.2	1:1-Beziehungen: <code>has_one</code> – <code>belongs_to</code>	78
4.9.3	1:N-Beziehungen: <code>has_many</code> – <code>belongs_to</code>	84
4.9.4	N:M-Beziehungen: <code>has_and_belongs_to_many</code>	90
4.9.5	Polymorphe Assoziationen: <code>has_many</code> – <code>belongs_to</code>	94
4.9.6	<code>has_many :through</code>	97
4.10	Aggregation	99
4.11	Vererbung	100
4.12	Transaktionen	103
4.13	Von Bäumen und Listen	105
4.13.1	<code>acts_as_list</code>	106
4.13.2	<code>acts_as_tree</code>	108
4.14	Validierung	110
4.14.1	Validierungs-Klassenmethoden	111
4.15	Callbacks	115
4.15.1	Überschreiben von Callback-Methoden	116
4.15.2	Callback-Makros	116
4.15.3	Observer	118
4.16	Konkurrierende Zugriffe und Locking	119
4.16.1	Optimistisches Locking	119
5	Action Controller	121
5.1	Controller-Grundlagen	121
5.1.1	Actions	122
5.1.2	Responses	123
5.2	Datenaustausch	126
5.2.1	Vom Controller zum View	126
5.2.2	Vom View zum Controller	127
5.2.3	Aus der Action in den View und zurück	129
5.3	Redirects	130
5.4	GET vs. POST	131
5.5	Unterschiedliche Response-Formate: <code>respond_to</code>	133
5.6	Zugriff auf Datensätze einschränken	135
5.7	Ausnahme fangen mit <code>rescue_from</code>	136

5.8	Sessions	137
5.8.1	Session-Daten löschen	139
5.9	Der Flash-Speicher	140
5.9.1	Weitere Flash-Methoden	141
5.10	Filter	141
5.10.1	Around-Filter	142
5.10.2	Bedingungen	143
5.10.3	Filterklassen und Inline-Filter	144
5.10.4	Filtervererbung	144
5.10.5	Filterketten	145
5.11	HTTP-Authentifikation	146
5.12	Routing und URL-Generierung	147
5.12.1	Routing	147
5.12.2	Anpassung des Routings	149
5.12.3	Root Route	150
5.12.4	Named Routes	150
5.12.5	URL-Generierung	152
5.13	Layout	153
5.13.1	Automatische Layoutzuweisung	154
5.13.2	Explizite Layoutzuweisung	154
5.13.3	Dynamische Bestimmung des Layouts	155
5.13.4	Action-spezifische Layouts	155
5.14	Datei-Download	156
5.14.1	Die Methode send_data	156
5.14.2	Die Methode send_file	157
5.14.3	Unsichere Dateidownloads verhindern	157
5.15	Datei-Upload	158
5.16	Rails-Konsole für Controller nutzen	158
6	Action View	161
6.1	HTML-Templates	161
6.2	Helper-Module	163
6.3	Action View Helper	165
6.3.1	Formulare	165
6.3.2	Schutz vor Cross-Site Request Forgery	167
6.3.3	Formular-Helper mit Bezug zu Modellen	168
6.3.4	Formular-Helper ohne Bezug zu Modellen	170

6.3.5	HTML-Tags	171
6.3.6	Texte und Zahlen	171
6.3.7	Datum und Zeit	172
6.3.8	Auswahlboxen	173
6.3.9	Verweise und URLs	175
6.3.10	Ressourcen einbinden	177
6.3.11	JavaScript	178
6.3.12	Code speichern und wiederverwenden	178
6.3.13	Debugging	179
6.3.14	HTML-Code filtern	179
6.4	Layouts	180
6.4.1	Layout aus Views beeinflussen	180
6.5	Partial Views	181
6.6	Anzeige Fehlermeldungen	185
6.7	XML-Templates	186
6.8	RJS-Templates	187
6.9	Caching	187
6.9.1	Seiten-Caching	188
6.9.2	Action-Caching	188
6.9.3	Fragment-Caching	190
6.9.4	Was Sie nicht cachen sollten	193
6.9.5	Caching testen	193
6.9.6	Action Cache Plugin	193
7	RESTful Rails	195
7.1	Was ist REST?	196
7.2	Warum REST?	197
7.3	Was ist neu?	197
7.4	Vorbereitung	198
7.5	Ressource Scaffolding	198
7.6	Das Modell	199
7.7	Der Controller	199
7.7.1	REST-URLs	200
7.7.2	REST-Actions verwenden <code>respond_to</code>	201
7.7.3	Accept-Feld im HTTP-Header	202
7.7.4	Formatangabe via URL	203
7.8	REST-URLs in Views	204

7.8.1	New und Edit	205
7.8.2	Path-Methoden in Formularen: Create und Update	206
7.8.3	Destroy	208
7.9	URL-Methoden im Controller	209
7.10	REST-Routing	210
7.10.1	Konventionen	210
7.10.2	Customizing	210
7.11	Verschachtelte Ressourcen	211
7.11.1	Anpassung des Controllers	213
7.11.2	Neue Helper für verschachtelte Ressourcen	214
7.11.3	Zufügen neuer Iterationen	216
7.11.4	Bearbeiten existierender Iterationen	218
7.12	Eigene Actions	219
7.12.1	Sind wir noch DRY?	221
7.13	Eigene Formate	221
7.14	RESTful AJAX	222
7.15	Testen	224
7.16	RESTful Clients: ActiveResource	225
7.17	Abschließend	226
8	Internationalisierung	227
8.1	Internationalisierung oder Lokalisierung?	228
8.2	Lokalisierung und Codes	228
8.2.1	Language Tag	228
8.2.2	Locale	228
8.3	Checkliste	229
8.4	Internationalisierung vorbereiten	230
8.5	Ruby Gettext-Package	231
8.5.1	Installation	232
8.5.2	Texte übersetzen	232
8.5.3	Die Schritte im Überblick	237
8.5.4	Anwendung und Gettext zusammenbringen	237
8.5.5	Lokalisierte Templates	238
8.5.6	Controller	238
8.5.7	Modelle	239
8.5.8	Dynamische Texte und Pluralisierung	241
8.5.9	Datum, Zahlen und Währungen	243

8.5.10	Wochen- und Monatsnamen	245
8.5.11	Sortierung	247
8.5.12	Zusammenfassung	248
9	Action Mailer	249
9.1	E-Mail-Versand	249
9.1.1	E-Mail erstellen	249
9.1.2	E-Mail-Objekt erzeugen	253
9.1.3	E-Mail versenden	254
9.1.4	Testen	256
9.1.5	Tipp zum zuverlässigen E-Mail-Versand	257
9.1.6	Multipart E-Mails	258
9.2	E-Mail-Empfang	260
9.2.1	E-Mail empfangen	261
9.2.2	Empfang testen	262
10	Ajax	263
10.1	Ajax und Rails	265
10.2	Hello, Ajax World	266
10.3	Ajax-Formulare	267
10.3.1	Task-Schnellerfassung	267
10.3.2	Ajax-Formulare und JavaScript	270
10.4	Feldbeobachter	270
10.4.1	Beispiel: Tasks filtern	272
10.5	Callback-Methoden	274
10.6	Drag and Drop	275
10.6.1	Einkaufen per Drag and Drop	275
10.6.2	Sortieren per Drag and Drop	278
10.7	Autocompletion	281
10.8	RJS Templates	282
10.8.1	Hinzufügen von Tasks visuell optimieren	283
10.8.2	Weitere Techniken	286
10.8.3	Debugging und Testen	289
10.9	Zusammenfassung und Ausblick	290
11	Produktion	291
11.1	Umgebungen in Rails	291
11.1.1	Umgebung definieren	291

11.1.2	Entwicklung	292
11.1.3	Test	292
11.1.4	Produktion	292
11.1.5	Umgebung hinzufügen	293
11.2	Webserver und Datenbank	294
11.2.1	Datenbank	294
11.2.2	Webserver Cluster	294
11.2.3	Apache und Mongrel	295
11.2.4	Alternativen	298
11.3	Skalierung	299
11.4	Session-Speicher	300
11.5	Tipps zur Performanz	301
11.6	Sicherheit	302
11.7	Auslieferung	305
11.8	Wartung	305
11.9	Überwachung	306
12	Deployment mit Capistrano (und Subversion)	309
12.1	Quickstart: Capistrano in 10 Minuten	310
12.1.1	Voraussetzungen	310
12.1.2	Installation von Capistrano	312
12.1.3	Anwendung Capistrano-ready machen	312
12.1.4	Konfiguration	313
12.1.5	Setup des entfernten Verzeichnisses	315
12.1.6	Erstes Deployment	315
12.1.7	Fallstricke	318
12.1.8	Nachfolgende Deployments	319
12.2	Datenbanksetup und Migration	319
12.3	Rollback eines Release	320
12.3.1	Rollback mit Datenbankmigration	320
12.4	Tasks	320
12.4.1	Ausführen von Tasks	321
12.4.2	Mehr über run und sudo	323
12.4.3	Weitere Task-Helper: delete, put und render	324
12.4.4	Transaktionen und Rollbacks	324
12.4.5	Überschreiben von Standardtasks	325
12.4.6	Tasks erweitern: Before- und After-Tasks	325

12.5 Variablen	326
12.6 FastCGI-Utilities	327
12.6.1 Spawner	327
12.6.2 Reaper	328
12.7 Gemeinsame Dateien – das Shared-Verzeichnis	328
12.7.1 Eine „persistente“ Datenbank-Konfiguration	328
12.8 Capistrano-Referenz	330
12.9 Capistrano-Konfiguration: deploy.rb	331
13 Testgetriebene Entwicklung mit Ruby und Test::Unit	333
13.1 Unit Tests – eine Definition	333
13.2 Ein Beispiel	334
13.3 Warum testen?	335
13.4 Test::Unit	336
13.4.1 Strukturierung von Unit Tests	337
13.4.2 Wohin mit den Tests?	339
13.4.3 Ausführen der Tests	339
13.4.4 Unabhängigkeit von Tests	340
13.5 Testgetriebene Softwareentwicklung	340
13.5.1 TODO-Listen	342
13.5.2 Beispiel	342
13.6 Retrospektive	344
14 Testgetriebene Entwicklung mit Ruby on Rails	345
14.1 Generierte Testklassen	345
14.2 Testdatenbank	346
14.3 Testausführung mit Rake	347
14.4 Eine Programmierepisode	347
14.4.1 Entwicklung einer Modellklasse	348
14.4.2 Entwicklung des Controllers	352
14.4.3 Programmierung des Views	353
14.4.4 Geänderte Anforderungen	355
14.4.5 Retrospektive	361
14.5 Unit Tests – Testen von Modellklassen	362
14.5.1 Struktur und Elemente von Modelltests	362
14.5.2 Testmethoden	363
14.5.3 Testdaten – Fixtures	364
14.5.4 Transaktionale Fixtures	367

14.5.5 Testrezepte für Modelle	367
14.6 Funktionale Tests	373
14.6.1 Testmethoden	373
14.6.2 Kontrollfluss-Assertions	374
14.6.3 Routing-Assertions	376
14.6.4 Datencontainer-Assertions	377
14.6.5 Template-Assertions	377
14.6.6 Testrezepte für Controller und Views	379
14.7 Integrationstests	383
14.7.1 Test-DSLs	385
14.7.2 Sessions	386
14.8 Testen von E-Mails	387
14.9 Mock-Objekte	387
14.10 Zusammenfassung	388
14.11 Assertions – Übersicht	389
Anhang	391
Literaturverzeichnis	399
Stichwortverzeichnis	401