

REST und HTTP

Einsatz der Architektur des Web für Integrationsszenarien

von
Stefan Tilkov

akt. u. erw. Auflage

REST und HTTP – Tilkov

schnell und portofrei erhältlich bei beck-shop.de DIE FACHBUCHHANDLUNG

Thematische Gliederung:

[Webprogrammierung](#)

dpunkt.verlag 2011

Verlag C.H. Beck im Internet:

www.beck.de

ISBN 978 3 89864 732 8

Inhaltsverzeichnis

1	Einleitung	1
1.1	Warum REST?	1
1.1.1	Lose Kopplung	2
1.1.2	Interoperabilität	3
1.1.3	Wiederverwendung	3
1.1.4	Performance und Skalierbarkeit	4
1.2	Zielgruppe und Voraussetzungen	4
1.3	Zur Struktur des Buches	5
2	Einführung in REST	9
2.1	Eine kurze Geschichte von REST	9
2.2	Grundprinzipien	11
2.3	Zusammenfassung	17
3	Fallstudie: OrderManager	19
3.1	Fachlicher Hintergrund	19
3.2	Ressourcen	20
3.2.1	Bestellungen	21
3.2.2	Bestellungen in unterschiedlichen Zuständen	27
3.2.3	Stornierungen	28
3.3	Repräsentationen	31
3.4	Zusammenfassung	31
4	Ressourcen	33
4.1	Ressourcen und Repräsentationen	33
4.2	Ressourcendesign	34
4.2.1	Primärressourcen	35
4.2.2	Subressourcen	36

4.2.3	Listen	36
4.2.4	Projektionen	36
4.2.5	Aggregationen	37
4.2.6	Aktivitäten	37
4.2.7	Konzeptressourcen	37
4.2.8	Evolutionäre Weiterentwicklung und YAGNI	37
4.3	Ressourcenidentifikation und URIs	38
4.3.1	URI, IRI, URL, URN, XRI?	38
4.3.2	Anatomie einer HTTP-URI	40
4.3.3	URI-Templates	43
4.4	URI-Design	43
4.4.1	URI-Entwurfsgrundsätze	44
4.4.2	REST aus Versehen	47
4.4.3	Stabile URIs	48
4.5	Zusammenfassung	49
5	Verben	51
5.1	Standardverben von HTTP 1.1	51
5.1.1	GET	51
5.1.2	HEAD	53
5.1.3	PUT	54
5.1.4	POST	54
5.1.5	DELETE	55
5.1.6	OPTIONS	56
5.1.7	TRACE und CONNECT	56
5.2	HTTP-Verben in der Praxis	56
5.3	Tricks für PUT und DELETE	57
5.3.1	HTML-Formulare	57
5.3.2	Firewalls und eingeschränkte Clients	59
5.4	Definition eigener Methoden	61
5.4.1	WebDAV	61
5.4.2	Partial Updates und PATCH	63
5.4.3	Multi-Request-Verarbeitung und BATCH	64
5.5	Zusammenfassung	66

6	Hypermedia	67
6.1	Hypermedia im Browser	67
6.2	HATEOAS und das »Human Web«	71
6.3	Hypermedia in der Anwendung-zu-Anwendung-Kommunikation	73
6.4	Ressourcenverknüpfung	74
6.5	Einstiegspunkte	74
6.6	Aktionsrelationen	76
6.7	Das <link>-Element	77
6.8	Standardisierung von Link-Relationen	80
6.9	Zusammenfassung	81
7	Repräsentationsformate	83
7.1	Formate, Medientypen und Content Negotiation	83
7.2	XML	84
7.3	HTML/XHTML	86
7.4	Textformate	88
	7.4.1 Plaintext	89
	7.4.2 URI-Listen	89
7.5	CSV	90
7.6	JSON	91
7.7	RSS und Atom	92
7.8	Binäre Formate	94
7.9	Microformats	94
7.10	RDF	96
7.11	Zusammenfassung	97
8	Fallstudie: AtomPub	99
8.1	Historie	99
8.2	Discovery und Metadaten	100
8.3	Ressourcentypen	103
8.4	REST und Atom/AtomPub	105
8.5	Zusammenfassung	106

9	Sitzungen und Skalierbarkeit	107
9.1	Cookies	108
9.2	Ressourcen- und Clientstatus	110
9.3	Skalierbarkeit und »Shared Nothing«-Architektur	112
9.4	Zusammenfassung	114
10	Caching	115
10.1	Expirationsmodell	115
10.2	Validierungsmodell	117
10.3	Cache-Topologien	119
10.4	Caching und Header	122
	10.4.1 Response-Header	122
	10.4.2 Request-Header	123
10.5	Schwache ETags	123
10.6	Invalidierung	124
10.7	Caching und personalisierte Inhalte	125
10.8	Caching im Internet	125
10.9	Zusammenfassung	125
11	Sicherheit	127
11.1	SSL und HTTPS	127
11.2	Authentisierung, Authentifizierung, Autorisierung	128
11.3	HTTP-Authentifizierung	129
11.4	HTTP Basic Authentication	130
11.5	Der 80 %-Fall: HTTPS + Basic-Auth	131
11.6	HTTP Digest Authentication	133
11.7	Browser-Integration und Cookies	134
11.8	HMAC	136
11.9	OpenID	137
11.10	OAuth	138
11.11	Autorisierung	140
11.12	Nachrichtenverschlüsselung und Signatur	140
11.13	Zusammenfassung	141

12	Dokumentation	143
12.1	Selbstbeschreibende Nachrichten	144
12.2	Hypermedia	145
12.3	HTML als Standardformat	145
12.4	Beschreibungsformate	146
12.4.1	WSDL	146
12.4.2	WADL	147
12.4.3	RDDL	149
12.5	Zusammenfassung	152
13	Erweiterte Anwendungsfälle	153
13.1	Asynchrone Verarbeitung	153
13.1.1	Notifikation per HTTP-»Callback«	155
13.1.2	Polling	156
13.2	Zuverlässigkeit	157
13.2.1	PUT statt POST	160
13.2.2	POST-PUT-Kombination	161
13.2.3	Reliable POST	162
13.3	Transaktionen	163
13.3.1	Atomare (Datenbank-)Transaktionen	163
13.3.2	Verteilte Transaktionen	164
13.3.3	Fachliche Transaktionen	165
13.4	Parallelzugriff und konditionale Verarbeitung	166
13.5	Versionierung	167
13.5.1	Zusätzliche Ressourcen	167
13.5.2	Erweiterbare Datenformate	167
13.5.3	Versionsabhängige Repräsentationen	168
13.6	Zusammenfassung	168
14	Fallstudie: OrderManager, Iteration 2	169
14.1	OrderEntry	170
14.1.1	Servicedokumentation	170
14.1.2	Medientypen	173
14.1.3	Bestellpositionen	173
14.2	Fulfilment	174
14.2.1	Notifikation über neue Bestellungen	174
14.2.2	Bestellübernahme	177

14.2.3	Produktionsaufträge	180
14.2.4	Versandfristen	181
14.2.5	Lieferdatum	181
14.3	Reporting	182
14.4	Zusammenfassung	183
15	Architektur und Umsetzung	185
15.1	Architekturebenen	185
15.2	Anwendungsarchitektur	186
15.2.1	Systemgrenzen und Ressourcen	187
15.2.2	Medientypen und Kontrakte	188
15.2.3	Identität und Adressierbarkeit	190
15.3	Softwarearchitektur	191
15.3.1	Schichten	191
15.3.2	Domänenmodell	193
15.3.3	Nutzung von Diensten	194
15.4	Systemarchitektur	196
15.4.1	Netztopologie	196
15.4.2	Caching	196
15.4.3	Firewalls	198
15.5	Frontend-Architektur	198
15.5.1	Benutzerschnittstellen und RESTful-HTTP-Backends ..	199
15.5.2	Sinn und Unsinn von Portalen	202
15.6	Web-API-Architektur	203
15.7	Zusammenfassung	205
16	»Enterprise REST«: SOA auf Basis von RESTful HTTP	207
16.1	SOA-Definitionen	207
16.2	Business/IT-Alignment	210
16.3	Governance	211
16.3.1	Daten- und Schnittstellenbeschreibungen	212
16.3.2	Registry/Repository-Lösungen	213
16.3.3	Discovery	213
16.4	Orchestrierung und Choreografie	214
16.5	Enterprise Service Bus (ESB)	215
16.6	WSDL, SOAP & WS-*: WS-Architektur	215
16.7	Zusammenfassung	219

Anhang

A	HTTP-Statuscodes	223
B	Fortgeschrittene HTTP-Mechanismen	227
B.1	Persistente Verbindungen	227
B.2	Request-Pipelining	227
B.3	Range Requests	228
B.4	Chunked Encoding	228
C	Werkzeuge und Bibliotheken	231
C.1	Kommandozeilen-Clients	231
C.2	HTTP-Server	232
C.3	Caches	233
C.4	Programmierungsumgebungen	234
C.4.1	Java	234
C.4.2	Microsoft .NET	237
C.4.3	Ruby und Ruby on Rails	238
C.4.4	Python, Perl & Co	238