

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
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
<b>2</b>	<b>Einführung in REST</b>	<b>9</b>
2.1	Eine kurze Geschichte von REST .....	9
2.2	Grundprinzipien .....	11
2.3	Zusammenfassung .....	19
<b>3</b>	<b>Fallstudie: OrderManager</b>	<b>21</b>
3.1	Fachlicher Hintergrund .....	21
3.2	Ressourcen .....	23
3.2.1	Bestellungen .....	23
3.2.2	Bestellungen in unterschiedlichen Zuständen .....	30
3.2.3	Stornierungen .....	30
3.3	Repräsentationen .....	32
3.4	Zusammenfassung .....	33
<b>4</b>	<b>Ressourcen</b>	<b>35</b>
4.1	Ressourcen und Repräsentationen .....	35
4.2	Ressourcendesign .....	36
4.2.1	Primärressourcen .....	37
4.2.2	Subressourcen .....	38
4.2.3	Listen .....	38

4.2.4	Projektionen .....	39
4.2.5	Aggregationen .....	39
4.2.6	Aktivitäten .....	39
4.2.7	Konzept- und Informationsressourcen .....	39
4.2.8	Evolutionäre Weiterentwicklung und YAGNI .....	40
4.3	Ressourcenidentifikation und URIs .....	40
4.3.1	URI, IRI, URL, URN, XRI? .....	41
4.3.2	Anatomie einer HTTP-URI .....	42
4.3.3	URI-Templates .....	46
4.4	URI-Design .....	47
4.4.1	URI-Entwurfsgrundsätze .....	47
4.4.2	REST aus Versehen .....	50
4.4.3	Stabile URIs .....	51
4.5	Zusammenfassung .....	52
<b>5</b>	<b>Verben</b>	<b>53</b>
5.1	Standardverben von HTTP 1.1 .....	53
5.1.1	GET .....	53
5.1.2	HEAD .....	55
5.1.3	PUT .....	56
5.1.4	POST .....	57
5.1.5	DELETE .....	57
5.1.6	OPTIONS .....	58
5.1.7	TRACE und CONNECT .....	58
5.2	HTTP-Verben in der Praxis .....	58
5.3	Tricks für PUT und DELETE .....	59
5.3.1	HTML-Formulare .....	59
5.3.2	Firewalls und eingeschränkte Clients .....	61
5.4	Definition eigener Methoden .....	63
5.4.1	WebDAV .....	63
5.4.2	Partial Updates und PATCH .....	65
5.4.3	Multi-Request-Verarbeitung .....	66
5.5	LINK und UNLINK .....	68
5.6	Zusammenfassung .....	69
<b>6</b>	<b>Hypermedia</b>	<b>71</b>
6.1	Hypermedia im Browser .....	71
6.2	HATEOAS und das »Human Web« .....	75
6.3	Hypermedia in der Anwendung-zu-Anwendung-Kommunikation ..	77
6.4	Ressourcenverknüpfung .....	78
6.5	Einstiegspunkte .....	78
6.6	Aktionsrelationen .....	80

6.7	Darstellung von Links und das <link>-Element . . . . .	81
6.8	Standardisierung von Link-Relationen . . . . .	84
6.9	Zusammenfassung . . . . .	85
<b>7</b>	<b>Repräsentationsformate . . . . .</b>	<b>87</b>
7.1	Formate, Medientypen und Content Negotiation . . . . .	87
7.2	JSON . . . . .	88
7.2.1	HAL . . . . .	89
7.2.2	Collection+JSON . . . . .	92
7.2.3	SIREN . . . . .	94
7.2.4	Fazit . . . . .	96
7.3	XML . . . . .	97
7.4	HTML/XHTML . . . . .	98
7.5	Textformate . . . . .	101
7.5.1	Plaintext . . . . .	101
7.5.2	URI-Listen . . . . .	102
7.6	CSV . . . . .	102
7.7	RSS und Atom . . . . .	103
7.8	Binäre Formate . . . . .	105
7.9	Microformats . . . . .	106
7.10	RDF . . . . .	107
7.11	Zusammenfassung . . . . .	109
<b>8</b>	<b>Fallstudie: AtomPub . . . . .</b>	<b>111</b>
8.1	Historie . . . . .	111
8.2	Discovery und Metadaten . . . . .	112
8.3	Ressourcentypen . . . . .	115
8.4	REST und Atom/AtomPub . . . . .	117
8.5	Zusammenfassung . . . . .	117
<b>9</b>	<b>Sitzungen und Skalierbarkeit . . . . .</b>	<b>119</b>
9.1	Cookies . . . . .	120
9.2	Ressourcen- und Clientstatus . . . . .	122
9.3	Skalierbarkeit und »Shared Nothing«-Architektur . . . . .	124
9.4	Zusammenfassung . . . . .	126
<b>10</b>	<b>Caching . . . . .</b>	<b>127</b>
10.1	Expiresmodell . . . . .	127
10.2	Validierungsmodell . . . . .	129
10.3	Cache-Topologien . . . . .	131

10.4	Caching und Header . . . . .	134
10.4.1	Response-Header . . . . .	134
10.4.2	Request-Header . . . . .	135
10.5	Schwache ETags . . . . .	135
10.6	Invalidierung . . . . .	136
10.7	Caching und personalisierte Inhalte . . . . .	137
10.8	Caching im Internet . . . . .	137
10.9	Zusammenfassung . . . . .	138
<b>11</b>	<b>Sicherheit</b>	<b>139</b>
11.1	SSL und HTTPS . . . . .	139
11.2	Authentisierung, Authentifizierung, Autorisierung . . . . .	140
11.3	HTTP-Authentifizierung . . . . .	141
11.4	HTTP Basic Authentication . . . . .	142
11.5	Der 80%-Fall: HTTPS + Basic-Auth . . . . .	143
11.6	HTTP Digest Authentication . . . . .	145
11.7	Browser-Integration und Cookies . . . . .	147
11.8	HMAC . . . . .	149
11.9	OpenID . . . . .	150
11.10	OAuth . . . . .	151
11.10.1	OAuth 1.0 . . . . .	152
11.10.2	OAuth 2.0 . . . . .	152
11.11	Autorisierung . . . . .	154
11.12	Nachrichtenverschlüsselung und Signatur . . . . .	155
11.13	Zusammenfassung . . . . .	156
<b>12</b>	<b>Dokumentation</b>	<b>157</b>
12.1	Selbstbeschreibende Nachrichten . . . . .	158
12.2	Hypermedia . . . . .	159
12.3	HTML als Standardformat . . . . .	159
12.4	Beschreibungsformate . . . . .	160
12.4.1	WSDL . . . . .	160
12.4.2	WADL . . . . .	161
12.4.3	Swagger, RAML und API Blueprint . . . . .	164
12.4.4	RDDL . . . . .	168
12.5	Zusammenfassung . . . . .	171
<b>13</b>	<b>Erweiterte Anwendungsfälle</b>	<b>173</b>
13.1	Asynchrone Verarbeitung . . . . .	173
13.1.1	Notifikation per HTTP-»Callback« . . . . .	175
13.1.2	Polling . . . . .	176

13.2	Zuverlässigkeit . . . . .	177
13.2.1	PUT statt POST . . . . .	180
13.2.2	POST-PUT-Kombination . . . . .	182
13.2.3	Reliable POST . . . . .	183
13.3	Transaktionen . . . . .	184
13.3.1	Atomare (Datenbank-)Transaktionen . . . . .	184
13.3.2	Verteilte Transaktionen . . . . .	185
13.3.3	Fachliche Transaktionen . . . . .	185
13.4	Parallelzugriff und konditionale Verarbeitung . . . . .	186
13.5	Versionierung . . . . .	187
13.5.1	Zusätzliche Ressourcen . . . . .	188
13.5.2	Erweiterbare Datenformate . . . . .	188
13.5.3	Versionsabhängige Repräsentationen . . . . .	189
13.6	Zusammenfassung . . . . .	189
<b>14</b>	<b>Fallstudie: OrderManager, Iteration 2</b>	<b>191</b>
14.1	OrderEntry . . . . .	192
14.1.1	Medientypen . . . . .	192
14.1.2	Servicedokumentation . . . . .	195
14.1.3	Bestellpositionen . . . . .	199
14.1.4	Zustandsänderungen . . . . .	200
14.2	Fulfilment . . . . .	205
14.2.1	Notifikation über neue Bestellungen . . . . .	205
14.2.2	Bestellübernahme . . . . .	207
14.2.3	Produktionsaufträge . . . . .	212
14.2.4	Versandfristen . . . . .	213
14.2.5	Lieferdatum . . . . .	213
14.3	Reporting . . . . .	215
14.4	Zusammenfassung . . . . .	217
<b>15</b>	<b>Architektur und Umsetzung</b>	<b>219</b>
15.1	Architekturebenen . . . . .	219
15.2	Domänenarchitektur . . . . .	220
15.2.1	Systemgrenzen und Ressourcen . . . . .	221
15.2.2	Medientypen und Kontrakte . . . . .	222
15.2.3	Identität und Adressierbarkeit . . . . .	224
15.3	Softwarearchitektur . . . . .	225
15.3.1	Schichten . . . . .	226
15.3.2	Domänenmodell . . . . .	227
15.3.3	Nutzung von Diensten . . . . .	228

15.4	Systemarchitektur . . . . .	230
15.4.1	Netztopologie . . . . .	230
15.4.2	Caching . . . . .	231
15.4.3	Firewalls . . . . .	232
15.5	Frontend-Architektur . . . . .	233
15.5.1	Benutzerschnittstellen und RESTful-HTTP-Backends . . . . .	234
15.5.2	Sinn und Unsinn von Portalen . . . . .	236
15.6	Web-API-Architektur . . . . .	237
15.7	Zusammenfassung . . . . .	240
<b>16</b>	<b>»Enterprise REST«: SOA auf Basis von RESTful HTTP</b>	<b>241</b>
16.1	SOA-Definitionen . . . . .	241
16.2	Business/IT-Alignment . . . . .	244
16.3	Governance . . . . .	245
16.3.1	Daten- und Schnittstellenbeschreibungen . . . . .	246
16.3.2	Registry/Repository-Lösungen . . . . .	247
16.3.3	Discovery . . . . .	247
16.4	Orchestrierung und Choreografie . . . . .	248
16.5	Enterprise Service Bus (ESB) . . . . .	249
16.6	WSDL, SOAP & WS-*: WS-Architektur . . . . .	249
16.7	Zusammenfassung . . . . .	253
<b>17</b>	<b>Weboberflächen mit ROCA</b>	<b>255</b>
17.1	REST: Nicht nur für Webservices . . . . .	255
17.2	Clientaspekte von ROCA . . . . .	258
17.2.1	Semantisches HTML . . . . .	259
17.2.2	CSS . . . . .	260
17.2.3	Die Rolle von JavaScript . . . . .	261
17.2.4	Unobtrusive JavaScript und Progressive Enhancement . . . . .	263
17.3	ROCA vs. Single Page Apps . . . . .	264
17.4	Zusammenfassung . . . . .	265

---

**Anhang**

---

<b>A</b>	<b>HTTP-Statuscodes</b>	<b>269</b>
<b>B</b>	<b>Fortgeschrittene HTTP-Mechanismen</b>	<b>275</b>
B.1	Persistente Verbindungen .....	275
B.2	Request-Pipelining .....	275
B.3	Range Requests .....	276
B.4	Chunked Encoding .....	276
<b>C</b>	<b>Werkzeuge und Bibliotheken</b>	<b>279</b>
C.1	Kommandozeilen-Clients .....	279
C.2	HTTP-Server .....	280
C.3	Caches .....	281
C.4	Programmierumgebungen .....	282
C.4.1	Java/JVM-Sprachen .....	282
C.4.2	Microsoft .NET .....	285
C.4.3	Ruby .....	286
C.4.4	Python, Perl, Node.js & Co .....	286
<b>D</b>	<b>HTTP/2 und SPDY</b>	<b>289</b>
D.1	Geschichte von HTTP .....	289
D.2	SPDY .....	290
D.3	HTTP/2 .....	291
	<b>Referenzen</b>	<b>293</b>
	<b>Index</b>	<b>305</b>