

Inhaltsübersicht

Teil I	Grundlagen	1
1	Application Programming Interfaces – eine Einführung	3
2	Qualitätsmerkmale	13
3	Allgemeines Vorgehen beim API-Design	27
Teil II	Java-APIs	37
4	Ausprägungen	39
5	Grundlagen für Java-APIs	45
6	Fortgeschrittene Techniken für Java-APIs	93
7	Kompatibilität von Java-APIs	125
Teil III	Remote-APIs	147
8	Grundlagen RESTful HTTP	149
9	Techniken für Web-APIs	167
10	SOAP-Webservices	225
11	Messaging	245

Teil IV	Übergreifende Themen	269
12	Dokumentation	271
13	Caching	295
14	Skalierbarkeit	305
15	Erweiterte Architekturthemen	325
16	API-Management	337
	Anhang	349
A	Literaturverzeichnis	351
	Index	357

Inhaltsverzeichnis

Teil I	Grundlagen	1
1	Application Programming Interfaces – eine Einführung	3
1.1	Eine kurze Geschichte der APIs	3
1.2	Web-APIs ab dem Jahr 2000	5
1.3	API-Definition	7
1.4	Vorteile einer API	9
1.5	Nachteile einer API	10
1.6	API als Produkt	11
1.7	Welche Strategien verfolgen Unternehmen mit APIs?	11
1.8	Zusammenfassung	12
2	Qualitätsmerkmale	13
2.1	Allgemeine Qualitätsmerkmale	13
2.2	Benutzbarkeit	14
2.2.1	Konsistent	14
2.2.2	Intuitiv verständlich	15
2.2.3	Dokumentiert	17
2.2.4	Einprägsam und leicht zu lernen	17
2.2.5	Lesbaren Code fördernd	18
2.2.6	Schwer falsch zu benutzen	20
2.2.7	Minimal	21
2.2.8	Stabil	22
2.2.9	Einfach erweiterbar	23
2.3	Konnaszenz	23
2.4	Zusammenfassung	26

3	Allgemeines Vorgehen beim API-Design	27
3.1	Überblick	27
3.2	Heuristiken und Trade-offs	28
3.3	Anforderungen herausarbeiten	29
3.4	Wenn Use Cases nicht ausreichen	29
3.5	Entwurf mit Szenarien und Codebeispielen	30
3.6	Spezifikation erstellen	32
3.7	Reviews und Feedback	33
3.8	Wiederverwendung	34
3.9	Zusammenfassung	35
Teil II	Java-APIs	37
4	Ausprägungen	39
4.1	Implizite Objekt-API	39
4.2	Utility-Bibliothek	42
4.3	Service	42
4.4	Framework	43
4.5	Eine Frage der Priorität	44
4.6	Zusammenfassung	44
5	Grundlagen für Java-APIs	45
5.1	Auswahl passender Namen	45
5.1.1	Klassennamen	46
5.1.2	Methodennamen	46
5.1.3	Parameternamen	49
5.1.4	Ubiquitäre Sprache	50
5.1.5	Fazit	51
5.2	Effektiver Einsatz von Typen	51
5.2.1	Semantischen Vertrag minimieren	52
5.2.2	Semantische Verletzung der Datenkapselung vermeiden	52
5.2.3	Werden Namen überschätzt?	54
5.2.4	Fazit	56
5.3	Techniken für Objektkollaboration	56
5.3.1	Tell, Don't Ask	56
5.3.2	Command/Query Separation	57
5.3.3	Law of Demeter	60
5.3.4	Platzierung von Methoden	61
5.3.5	Fazit	62

5.4	Jigsaw-Module	62
5.5	Minimale Sichtbarkeit	65
5.5.1	Jigsaw-Module	65
5.5.2	Packages	66
5.5.3	Klassen	66
5.5.4	Methoden	66
5.5.5	Felder	66
5.5.6	Fazit	66
5.6	Optionale Hilfsmethoden	67
5.6.1	Komfort	67
5.6.2	Utility-Klassen	67
5.6.3	Fazit	68
5.7	Optionale Rückgabewerte	68
5.7.1	Ad-hoc-Fehlerbehandlung	69
5.7.2	Null-Objekte	70
5.7.3	Verwendung der Klasse <code>java.util.Optional</code>	71
5.7.4	Fazit	72
5.8	Exceptions	72
5.8.1	Ausnahmesituationen	72
5.8.2	Checked Exception versus Unchecked Exception	73
5.8.3	Passende Abstraktionen	74
5.8.4	Dokumentation von Exceptions	75
5.8.5	Vermeidung von Exceptions	76
5.8.6	Fazit	77
5.9	Objekterzeugung	77
5.9.1	Erzeugungsmuster der GoF	78
5.9.2	Statische Factory-Methode	78
5.9.3	Builder mit Fluent Interface	80
5.9.4	Praktische Anwendung der Erzeugungsmuster	81
5.9.5	Fazit	83
5.10	Vererbung	83
5.10.1	Ansätze zum Einsatz von Vererbung	84
5.10.2	Stolperfallen bei Vererbung	85
5.10.3	Bedeutung für API-Design	87
5.10.4	Fazit	88
5.11	Interfaces	88
5.11.1	Typen nachrüsten	89
5.11.2	Unterstützung für nicht triviale Interfaces	89
5.11.3	Markierungsschnittstellen	90
5.11.4	Funktionale Interfaces	90
5.11.5	Fazit	91
5.12	Zusammenfassung	91

6	Fortgeschrittene Techniken für Java-APIs	93
6.1	Fluent Interface	93
6.1.1	DSL-Grammatik	94
6.1.2	Schachteln versus Verketteten	97
6.1.3	Fluent Interface von jOOQ	97
6.1.4	Ist der Aufwand gerechtfertigt?	98
6.1.5	Fazit	98
6.2	Template-Methoden	98
6.2.1	API versus SPI	99
6.2.2	Erweiterbare Parameter	101
6.2.3	Fazit	101
6.3	Callbacks	101
6.3.1	Synchrone Callbacks	102
6.3.2	Asynchrone Callbacks	103
6.3.3	Fazit	104
6.4	Annotationen	105
6.4.1	Auswertung zum Kompilierzeitpunkt	105
6.4.2	Auswertung zur Laufzeit	107
6.4.3	Fazit	108
6.5	Wrapper-Interfaces	109
6.5.1	Proxy	109
6.5.2	Adapter	111
6.5.3	Fassade	111
6.5.4	Fazit	114
6.6	Immutability	114
6.6.1	Wiederverwendung	115
6.6.2	Thread-Sicherheit	116
6.6.3	Einfachheit	116
6.6.4	Umsetzung	117
6.6.5	Automatische Überprüfung mit dem Mutability Detector	118
6.6.6	Codegenerierung mit Immutables	119
6.6.7	Fazit	119
6.7	Thread-sichere APIs	120
6.7.1	Vorteile	120
6.7.2	Nachteile	120
6.7.3	Was bedeutet Thread-Sicherheit?	121
6.7.4	Fazit	124
6.8	Zusammenfassung	124

7	Kompatibilität von Java-APIs	125
7.1	Kompatibilitätsstufen	125
7.1.1	Code-Kompatibilität	125
7.1.2	Binäre Kompatibilität	126
7.1.3	Funktionale Kompatibilität	127
7.2	Verwandtschaftsbeziehungen	129
7.3	Design by Contract	130
7.4	Codeänderungen	132
7.4.1	Package-Änderungen	133
7.4.2	Interface-Änderungen	134
7.4.3	Klassenänderungen	135
7.4.4	Spezialisierung von Rückgabetypen	136
7.4.5	Generalisierung von Parametertypen	137
7.4.6	Generics	137
7.4.7	Ausnahmen	138
7.4.8	Statische Methoden und Konstanten	138
7.5	Praktische Techniken für API-Änderungen	139
7.6	Test Compatibility Kit	143
7.7	Zusammenfassung	145
Teil III	Remote-APIs	147
8	Grundlagen RESTful HTTP	149
8.1	REST versus HTTP	149
8.2	REST-Grundprinzipien	150
8.3	Ressourcen – die zentralen Bausteine	155
8.4	HTTP-Methoden	157
8.5	HATEOAS	162
8.6	Zusammenfassung	165
9	Techniken für Web-APIs	167
9.1	Anwendungsbeispiel: Onlineshop	167
9.2	URI-Design	177
9.3	Medientypen	181
9.4	Fehlerbehandlung	195
9.5	Versionierung	200
9.5.1	Daten- und Sprachversionierung	200
9.5.2	Kompatibilität und Perspektive	201
9.5.3	Versionsidentifikation	203

9.6	Sicherheitsmechanismen	207
9.7	Partielle Rückgaben	210
9.8	GraphQL	214
9.9	odata	220
9.10	Zusammenfassung	224
10	SOAP-Webservices	225
10.1	SOAP-Grundlagen	225
10.2	WSDL-Grundlagen	228
10.3	Entwurfsansätze und -muster	231
10.4	Versionierung	238
10.5	SOAP versus REST	242
10.6	Zusammenfassung	243
11	Messaging	245
11.1	Routenplanung für Lkw-Transporte (Teil 1)	246
11.2	Message Broker	247
11.3	Produkte	250
11.4	Standards und Protokolle	254
11.5	Routenplanung für Lkw-Transporte (Teil 2)	257
11.6	Transaktionen und garantierte Nachrichtenzustellung	259
11.7	Asynchrone Verarbeitung und REST	262
11.8	Push Notifications	264
11.9	Zusammenfassung	267
Teil IV	Übergreifende Themen	269
12	Dokumentation	271
12.1	Motivation	271
12.2	Zielgruppen unterscheiden	272
12.3	Allgemeiner Aufbau	272
12.4	Beispiele	274
12.5	Dokumentation von Java-APIs	276
12.6	Dokumentation von Web-APIs	284
12.7	Zusammenfassung	294

13	Caching	295
13.1	Anwendungsfälle	295
13.2	Performance-Vorteil	296
13.3	Verdrängungsstrategien	296
13.4	Cache-Strategien für Schreibzugriffe	297
13.5	Cache-Topologien für Webanwendungen	298
13.6	HTTP-Caching	299
13.7	Zusammenfassung	304
14	Skalierbarkeit	305
14.1	Anwendungsfall	305
14.2	Grundlagen	306
14.3	Load Balancing	309
14.4	Statuslose Kommunikation	313
14.5	Skalierung von Datenbanken	315
14.6	Skalierung von Messaging-Systemen	320
14.7	Architekturvarianten	322
14.8	Zusammenfassung	324
15	Erweiterte Architekturthemen	325
15.1	Consumer-Driven Contracts	325
15.2	Backends for Frontends	328
15.3	Vernachlässigte Frontend-Architektur	331
15.4	Netflix-APIs	332
15.5	Zusammenfassung	335
16	API-Management	337
16.1	Überblick	337
16.2	Funktionen einer API-Management-Plattform	338
16.3	API-Management-Architektur	340
16.4	Open-Source-Gateways	345
16.5	Zusammenfassung	348
Anhang		349
A	Literaturverzeichnis	351
	Index	357