

Inhalt

1	Einleitung.....	1
1.1	Alle bauen APIs.....	1
1.2	Web-Frameworks	2
1.3	Alle wollen Go	2
1.4	Warum REST?	3
1.5	Aufbau des Buches	4
1.6	Disclaimer	4
1.7	Gebrauchsanleitung.....	5
1.8	Code-Beispiele	5
2	Go	7
2.1	Installation.....	7
2.2	Der Go-Workspace.....	8
2.3	Test der Installation	9
2.4	Programmstruktur.....	9
2.5	Packages	10
2.6	Funktionen	11
2.7	Variablen	11
2.8	Typen	12
2.9	Schleifen	14
2.10	Verzweigungen	15
2.11	Methoden	17
2.12	Pointer.....	18
2.13	Interfaces	19
2.14	Arrays und Slices	21
	2.14.1 Arrays	21
	2.14.2 Slices	21
	2.14.3 Polymorphe Arrays.....	23

2.15	Maps	23
2.16	Verzögerungen, Panic und Recover	24
2.17	Was sonst noch?	26
3	REST	27
3.1	Überblick	27
3.2	Ressourcen	29
3.3	Uniform Ressource Identifier	29
3.4	HTTP-Methoden	30
3.4.1	GET	30
3.4.2	POST	31
3.4.3	PUT	31
3.4.4	DELETE	31
3.4.5	PATCH	31
3.4.6	HEAD	31
3.4.7	OPTIONS	31
3.5	Repräsentationen	32
3.6	Statuscodes	32
3.6.1	2xx Erfolg	33
3.6.2	3xx Umleitung	33
3.6.3	4xx Client-Fehler	33
3.6.4	5xx Server-Fehler	33
3.7	Hypermedia	34
3.8	REST als Architekturstil	35
3.8.1	Client-Server	35
3.8.2	Zustandslosigkeit	35
3.8.3	Cacheability	36
3.8.4	Uniform Interface	36
3.8.5	Layered System	36
3.8.6	Code on Demand	37
3.9	Zusammenfassung	37
4	HTTP und JSON	39
4.1	Das Package net/http	39
4.1.1	Request Handling	39
4.1.2	Level-2-Services	43
4.2	Das Package encoding/json	45
4.2.1	JSON-Encoding	46
4.2.2	JSON-Decoding	48
4.3	HTTP und JSON	49
4.4	Abschluss	50

5	Restvoice	51
5.1	Vision	51
5.2	Use Cases	52
5.3	Domainmodell	53
5.4	Ressourcen	55
5.5	Kern-Use-Cases	55
5.5.1	Rechnung erstellen	57
5.5.2	Buchung zufügen	60
5.5.3	Buchung löschen	63
5.5.4	Rechnung abschließen	64
5.5.5	Rechnung zustellen	65
5.6	Kritik am Entwurf	66
5.6.1	Sicherheit	66
5.6.2	Testbarkeit	67
5.6.3	Kopplung	67
5.6.4	RESTfulness	68
5.7	Warum Go?	68
5.8	Abschluss	69
6	Design	71
6.1	Layering	72
6.2	Domain-driven Design	72
6.3	Hexagonale Architektur	73
6.3.1	Beispiel	75
6.3.2	Der Use Case ELIZA	75
6.3.3	Ports und Adapter	76
6.4	Use Case „Rechnung erstellen“	77
6.4.1	HTTP ⇒ Invoice	78
6.4.2	Geschäftslogik	79
6.4.3	Invoice ⇒ HTTP	80
6.4.4	Bootstrapping	80
6.5	Use Case „Tätigkeit buchen“	82
6.6	Use Case „Rechnung abschließen“	83
6.7	Use Case „Rechnung anfordern“	85
6.8	Was haben wir erreicht?	86
6.8.1	Entkopplung	87
6.8.2	Wartbarkeit	87
6.8.3	Testbarkeit	88

7	Testing	89
7.1	Unit Tests	89
7.2	Unit Tests in Go	90
7.3	HTTP-Tests in Go	92
7.4	Was testen?	93
7.5	Beispiel: Rechnung abschließen	94
7.5.1	Testszenario	95
7.5.2	Method under Test	95
7.5.3	Unit Test	97
7.5.4	Integrationstest	97
7.6	Fake- und Mock-Objekte	99
7.6.1	Fake-Objekte	99
7.6.2	Mock-Objekte	101
7.7	Unit Test mit Fake-Objekt	102
7.8	HTTP-Test	102
7.9	API-Test	103
7.10	Testseparation	105
7.10.1	Build Tags: -tags	105
7.10.2	Short Modus: -short	106
7.10.3	Einzelne Tests ausführen: -run	106
7.11	Test Coverage	106
7.12	Abschluss	108
8	Hypermedia als Motor	109
8.1	HATEOAS	110
8.2	Hypermedia Rechnungsstellung	110
8.2.1	Zustandsübergänge	111
8.2.2	Wie sage ich es meinem Client?	112
8.3	Hypertext Application Language	113
8.4	HAL in Go	116
8.4.1	Geschäftslogik	116
8.4.2	Media Type „application/hal+json“	117
8.5	Hypermedia-Clients	119
8.6	Resource Expansion	121
8.7	CRUD vs. Hypermedia	124
8.8	Abschluss	125

9	Sicherheit	127
9.1	Client-Authentifizierung	127
9.1.1	HTTP Basic	128
9.1.2	HTTP Digest	132
9.1.3	JSON Web Token	134
9.2	Server-Authentifizierung	139
9.3	Autorisierung	141
9.4	Verschlüsselung	143
9.5	Abschluss	144
10	Skalierbarkeit	147
10.1	Horizontale Skalierung	147
10.2	Zustandslosigkeit	148
10.2.1	Sitzungsstatus	148
10.2.2	Clientstatus	148
10.2.3	Ressourcenstatus	149
10.3	Restvoice skalieren	150
10.4	Vertikale Skalierung	150
10.4.1	Go-Routinen	151
10.4.2	Restvoice vertikal skalieren	151
10.5	Abschluss	153
11	Caching	155
11.1	Arten von Caches	155
11.1.1	Lokaler Cache	155
11.1.2	Proxy Cache	156
11.1.3	Reverse Proxy Cache	156
11.2	HTTP Caching	156
11.2.1	GET	156
11.2.2	Cachebare Responses	156
11.2.3	Cache-Validierung	157
11.2.4	Cache-Invalidierung	159
11.3	Restvoice Caching	159
11.4	Activity Caching	159
11.4.1	Cacheable Responses	159
11.4.2	Cache-Validierung	161
11.4.3	Invalidierung	163
11.4.4	Alternative ETags	164
11.5	Abschluss	164

12 Wie geht es weiter?	165
12.1 Danke!	165
12.2 Feedback	166
Literatur	167
Stichwortverzeichnis	169