

Auf einen Blick

1	Einführung	13
2	Die Grundlagen von Go	45
3	Microservices	149
4	Die Qualitätssicherung	309
5	Best Practices – idiomatisches und effektives Go	331
6	Go-Service an die Cloud anbinden	367

Inhalt

Materialien zum Buch	9
Geleitwort des Fachgutachters	11

1 Einführung 13

1.1 Was sind Microservices?	13
1.1.1 Probleme bei monolithischen Architekturen	13
1.1.2 Gemeinsame Eigenschaften von Microservices	16
1.2 Go – Einführung und Geschichte	26
1.3 Warum eignet sich Go so gut für Microservices?	34
1.3.1 Einfache, prägnante Sprache	35
1.3.2 Nebenläufigkeit	35
1.3.3 Effizienz entwickeln	36
1.3.4 Sicherheit	38
1.3.5 Leistungsfähigkeit	39
1.3.6 Standardbibliothek	41
1.3.7 Mitgelieferter Werkzeugkasten	42

2 Die Grundlagen von Go 45

2.1 Installation	46
2.1.1 Go unter Microsoft Windows installieren	47
2.1.2 Go unter Linux installieren	49
2.1.3 Go auf einem Mac installieren	50
2.1.4 Go innerhalb eines Docker Containers nutzen	51
2.1.5 Test der Installation	52
2.1.6 Die erste Go-Anwendung	53
2.1.7 Umgebungseinstellungen	54
2.1.8 Der Go-Workspace – Wie organisiere ich Go-Code?	61
2.1.9 All inclusive – alle Tools im Überblick	67
2.1.10 Die Entwicklungsumgebungen – VSCode und GoLand im Vergleich	82
2.2 Sprachgrundlagen von Golang	89
2.2.1 Formatierung des Quellcodes	90
2.2.2 Packages, Variablen und Funktionen	92

2.2.3	Kontrollstrukturen (Go-Statements)	100
2.2.4	Typen und Interfaces	106
2.2.5	Typen für Collections	114
2.2.6	Mit Fehlern umgehen (Error Handling)	120
2.2.7	Mit Pointer und Pointer Receiver arbeiten	127
2.2.8	Die Go-Dokumentation	130
2.2.9	Quellcode und Packages dokumentieren	131
2.3	Module und Libraries	135
2.3.1	Third-Party Libraries einbinden	138
2.3.2	Eigene Libraries erstellen und teilen	141
2.3.3	Module erstellen, verwenden und versionieren	144
2.3.4	Go-Projekte in Go-Module umwandeln	147
3	Microservices	149
<hr/>		
3.1	Grundlagen	152
3.1.1	Den Architekturstil umsetzen	152
3.1.2	Mit Fehlern leben lernen – Resilient Design	160
3.1.3	Die Vorteile der asynchronen Kommunikation	168
3.1.4	Die Aufteilung in Dienste und Services	175
3.2	Microservices mit Go	176
3.2.1	RESTful HTTP-Services mit Go umsetzen	176
3.2.2	Als Client einen HTTP-Service aufrufen	190
3.2.3	Zwischen Services kommunizieren – das Protokoll gRPC	193
3.2.4	Ereignisbasierte Kommunikation – Messaging als Interaktionsmodell	202
3.2.5	Loggings konfigurieren und auswerten	209
3.2.6	Aspektororientierte Funktionalität mit Middlewares	213
3.2.7	Fremdbibliotheken einsetzen	217
3.3	Go-Services im Microservice-Umfeld	223
3.3.1	Die Fehlerbehandlung auf Client-Seite	223
3.3.2	Anwendungen mit Prometheus-Monitoren	226
3.3.3	HTTPS auf Server- und Client-Seite	238
3.3.4	Services konfigurieren	247
3.3.5	Services absichern und stabilisieren	252
3.4	Datenbankanbindung	256
3.4.1	SQL-Treiber auswählen und installieren	257
3.4.2	Das Package database/sql	263

3.4.3	ORM Mapper für Golang	272
3.4.4	No-SQL-Datenbanken	281
3.5	Die Nebenläufigkeit mit Go – Concurrency	296
3.5.1	Was sind Go-Routines?	297
3.5.2	Wie nutze ich Channels?	299
3.5.3	Typische Vorgehensweisen von Routinen und Channels mit Go	303

4 Die Qualitätssicherung 309

4.1	Das Unit-Test Framework	310
4.1.1	Unit-Tests erstellen	311
4.1.2	Table-Driven-Tests	313
4.1.3	Parallele Ausführung der Unit-Tests	316
4.1.4	Testabdeckung	317
4.2	Benchmarks für Go erstellen	318
4.2.1	Benchmarks implementieren	318
4.2.2	Parallele Ausführung von Benchmarks	319
4.2.3	Optimierungen unterdrücken	320
4.3	Microservices testen und optimieren	320
4.3.1	Unit-Test für einen HTTP-Service	322
4.3.2	Profiling von HTTP-Services	323

5 Best Practices – idiomatisches und effektives Go 331

5.1	Context-Package einsetzen	332
5.1.1	Context-Interface	334
5.1.2	Einen Kontext und davon abgeleitete Kontexte erstellen	335
5.1.3	Methoden mit einem Timeout ausstatten	336
5.1.4	Kontextabhängige Werte setzen	338
5.1.5	Context im HTTP-Handler	339
5.2	Error-Handling	340
5.2.1	Fehlerbehandlung weniger ausführlich gestalten	341
5.2.2	Fehlerverhalten statt Fehlertyp	343
5.3	Projekte sinnvoll organisieren	345
5.3.1	Übergreifendes	345
5.3.2	Verwaltungsstile	349

5.4	Code-Schnipsel und Tipps und Tricks	357
5.4.1	Dateien lesen und schreiben	357
5.4.2	Stucts auf Kommandozeile ausgeben	359
5.4.3	If-Statement mit Initialisierung und Bedingung	360
5.4.4	Single-Method-Interface und Function-Types	361
5.4.5	Logging auf die Schnelle	363
5.4.6	Ausführung von Go-Routinen über Ausführungspool begrenzen	363
5.4.7	Interface-Implementierungen prüfen	364
5.4.8	Prüfung auf leere String-Werte	365

6 Go-Service an die Cloud anbinden 367

6.1	Deployment mit Docker und Kubernetes	368
6.1.1	Docker	368
6.1.2	Docker Compose	374
6.1.3	Kubernetes	378
6.2	Das Go Cloud Development Kit	383
6.2.1	File-Storage	386
6.3	Google Cloud Development	390
6.3.1	Deployment in Kubernetes	391
6.3.2	Cloud Functions	394
6.4	Amazon Web Services Deployment	397
6.4.1	Deployment einer Anwendung über Beanstalk	398
6.4.2	Lambda-Funktionen	399
6.5	Microsoft-Azure-Anbindung	402
6.5.1	Deployment in Kubernetes	402
Index		407