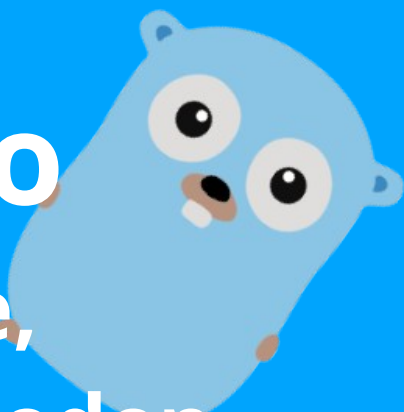


# Way To Go

## Wie ich lernte, (wieder) gern zu coden



Andreas Schneider, Tech Lead, Staffbase

# Motivation

- Microservices sind toll
- Cloud-Hosting muss ressourcensparend sein
- Spring-Boot bootet mir zu langsam
- Die JVM ist zu fett

Enter: Go



Gopher von Takuya Ueda, CC 3.0

```
package main

import "fmt"

func main() {
    fmt.Println("Hallo Publikum!")
}
```

# Was ist Go?

## Google brauchte eine bessere Programmiersprache

- einfach zu verstehen
- effiziente Ressourcennutzung
- schnelle Compilation
- soll von Haus aus „moderne“ Probleme lösen

# Was ist Go?

- 2007 von Robert Griesemer, Rob Pike und Ken Thompson ins Leben gerufen
- Go erfindet nicht das Rad neu, sondern vereinfacht es.
- Sprache nutzt Konzepte aus C und Oberon
- Garbage Collected

## Was kann Go (u.a.)?

- HTTP/2 Server
- JSON, XML, CSV
- Template Engine
- Flags
- Test, Benchmark und Profiling
- Ausgezeichnete Nebenläufigkeit
- Cross Compilation
- SMTP, FTP, ...
- Datenbankzugriff
- Kryptographie
- ...

## Das nutzt doch keiner, oder?

<https://github.com/golang/go/wiki/GoUsers>

- Google (danke, Sherlock)
- Docker
- Sixt
- Uber
- Twitch
- Soundcloud
- Netflix
- New York Times
- BBC
- ...

# Live Coding Time



## Bin überzeugt. Wie leg' ich los?

- Wo bekomme ich Go?  
[golang.org](https://golang.org)
- Zeig mir Features!  
[tour.golang.org](https://tour.golang.org)
- Beispiele für gängige Probleme  
[gobyexample.com](https://gobyexample.com)
- Gewusst wie. Vorbildhafter Go-Code  
[golang.org/doc/effective\\_go.html](https://golang.org/doc/effective_go.html)
- Gib ... mir ... Libraries!  
[awesome-go.com](https://awesome-go.com)

# Irgendwelche Bibliotheks-Empfehlungen?

- Logger: [Uber's zap](#)
- HTTP Router: [Chi](#)
- Noch mehr Command-Line Parsing (mit Sub-Commands und Environment Variablen): [Kong](#)
- Test-Helper: [Testify](#)

## Tipps für den Einstieg

- Startet immer einfach (KISS). Keine Abstraktion, bis sie wirklich nötig ist. Keine „Package-Struktur“, bevor man sie nicht wirklich braucht.
- Packages in Go entsprechen Libraries in C bzw Modules in Java. Ja, Packages können viele viele Dateien und „Klassen“ beinhalten.
- OOP funktioniert in Go anders als in Sprachen mit Vererbung. Versucht nicht, Pattern aus anderen Sprachen mitzubringen.
- Schreibt Code nicht „clever“. Je einfacher der Code, umso besser ist er wartbar.
- Schreibt Go nicht ab, nur weil es nicht viele Features hat. Lernt die Einfachheit zu lieben!

**Vielen Dank für Eure  
Aufmerksamkeit!**

