Google Developers

Production-Ready “Hello World!”

Daniela Petruzalek

Principal Software Engineer @ JP Morgan Chase

Google Developer Expert - Go & GCP

@danicat83

Experts

About me...

Brazilian, living in the UK since 2019

19 years of career: SWE, DBA, Sales Eng, Data Eng, Consulting, SWE...

Two cats: Paprika & Shoyu

Hobbies: video games, anime, kendo, Japanese, piano, K-doramas, running

ADHD, mostly an introvert

Agenda

Problem statement

A minimal hello world server

What is production-ready?

Katas

Coding Katas

Gilded Rose Kata (solution by Sandi Metz):

https://www.youtube.com/watch?v=8bZh5LMaSmE

https://github.com/gamontal/awesome-katas

Problem statement

You are in charge of developing an application called "pingpong". This application is a web server that listens to requests and if it receives a GET request on the "/ping" path it should respond the message "pong".

The code should be production ready: include all the things that you consider important in production code, but you are **only allowed** to use packages from the **standard library**.

Minimal a.k.a. “Make it Work” version

```go
package main

import "net/http"

func main() {

  http.HandleFunc("/ping",

    func(w http.ResponseWriter, req *http.Request) {

      w.Write([]byte(`{"message":"pong"}`))

    })

  http.ListenAndServe(":8080", nil)

}
```

The Software Evolution Process

Break things

Make it work

Make it better

Repeat

What is production-ready?

Software that satisfies the business needs of performance, correctness and maintainability.

What is production-ready?

Performance: availability, response time / latency

Correctness: does what it is supposed to do

Maintainability: easy to diagnose and modify, specially on a bad day

How to ensure performance?

Use appropriate algorithms and data structures

Maximize the use of your infrastructure: replicas, sharding, concurrency...

“Channels orchestrate, mutexes serialize”

Caching, memoization, etc.

How to ensure correctness?

Are we building the right thing?

Are non-functionals (e.g. security) covered?

Unit testing

Integration, E2E testing

Regression tests

How to ensure maintainability?

Is the code easy to read?

Is the code easy to change? Patterns, hardcoded values...

Do you have enough info to diagnose problems? e.g. logs, metrics

Can you diagnose things in production?

How do you evaluate this version?

```go
package main

import "net/http"

func main() {

  http.HandleFunc("/ping",

    func(w http.ResponseWriter, req *http.Request) {

      w.Write([]byte(`{"message":"pong"}`))

    })

  http.ListenAndServe(":8080", nil)

}
```

Making it production ready

- Make it modular
- Make it configurable
- Make it testable
- Add diagnostic info
- Add instrumentation

Code for this talk

https://github.com/danicat/pingpong

daniela.petruzalek@gmail.com

@danicat

@danicat83

https://github.com/danicat/public-speaking