

Building an event driven microservice using Kafka and MongoDB with Golang

Salman Shah
Software Engineer | Myntra



Session 2

- Development Tips
- Going Into Production

24/10/2020

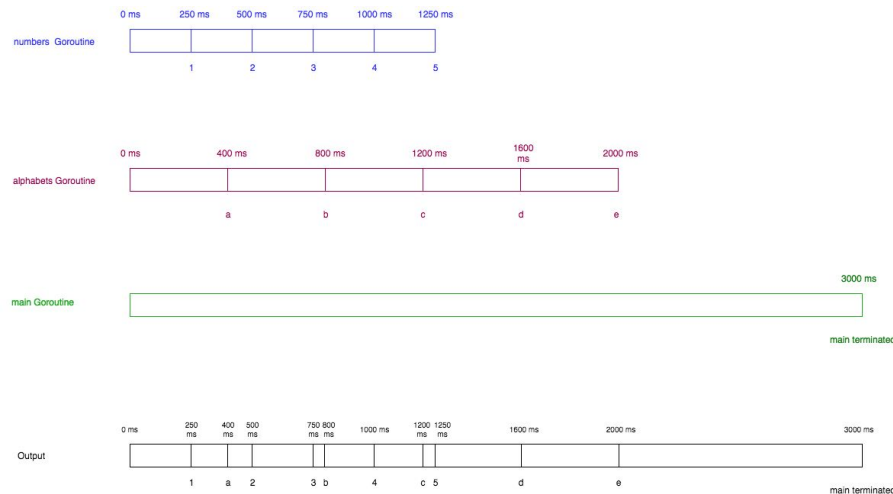


Development Learnings



Understanding goroutines

Functions which occur concurrently and extremely lightweight threads with minimal cost



Credits: GolangBot

24/10/2020



Dev Tip 1: Use Goroutines

- Using Goroutines to parallelize network calls to MongoDB and Kafka as a Producer

```
go f("goroutine")
```

```
go func(msg string) {  
    fmt.Println(msg)  
}("going")
```

Credits: Gobyexample

24/10/2020



Dev Tip 2: Defer, Panic and Recover

Defer allows a function to be called at the end.

Panic is a built-in go feature to stop the program flow on error.

Recover allows the program to regain control of a panicking goroutine

```
func main() {  
    f()  
    fmt.Println("Returned normally from f.")  
}  
  
func f() {  
    defer func() {  
        if r := recover(); r != nil {  
            fmt.Println("Recovered in f", r)  
        }  
    }()  
    fmt.Println("Calling g.")  
    g(0)  
    fmt.Println("Returned normally from g.")  
}
```

Credits: Golang.org

24/10/2020



Dev Tip 3: sync.Once for initialisation

Initialising config
parameters/ reading from
config files is best served
using sync.Once

```
func main() {  
    var once sync.Once  
    onceBody := func() {  
        fmt.Println("Only once")  
    }  
    done := make(chan bool)  
    for i := 0; i < 10; i++ {  
        go func() {  
            once.Do(onceBody)  
            done <- true  
        }()  
    }  
    for i := 0; i < 10; i++ {  
        <-done  
    }  
}
```

Credits: Golang.org

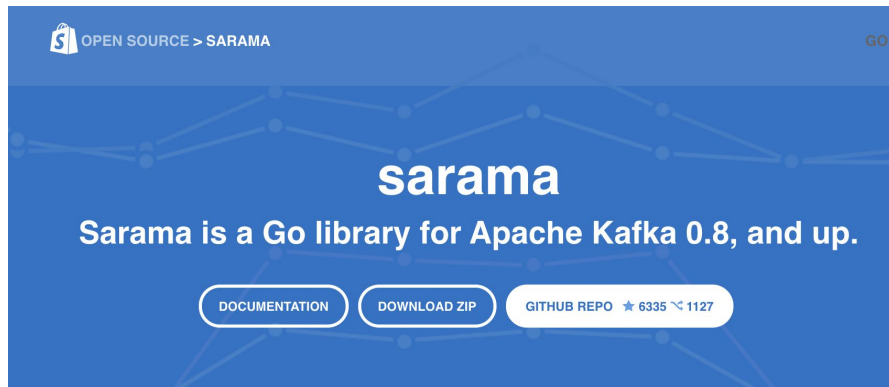
24/10/2020



Go Library - Sarama

Sarama is an MIT-licensed Go client library for Apache Kafka version 0.8 (and later).

<https://github.com/Shopify/sarama>



Sarama is an MIT-licensed Go client library for [Apache Kafka](#) version 0.8 (and later).

API documentation and examples are available via [godoc](#).

Credits: Shopify

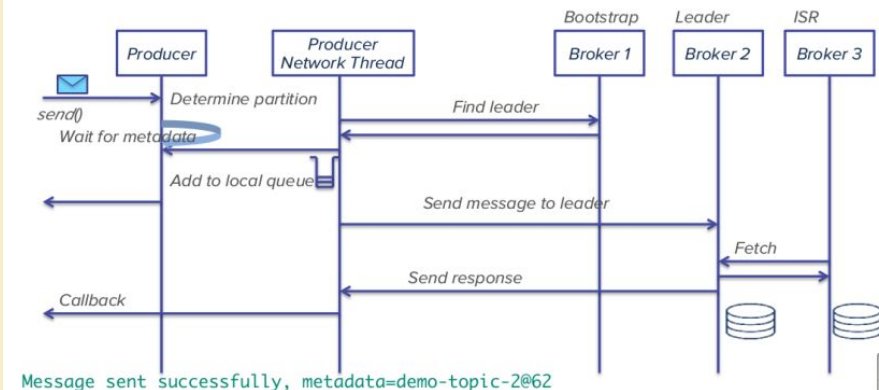
24/10/2020



Dev Tip 4: Prefer Async Producer

Using Async Producer over Sync Producer to reduce latencies

Producer sequence diagram



Credits: Spring One Platform

24/10/2020



Dev Tip 5: Unit Testable

Individual Components
Unit Testable

Usage of testify:

<https://github.com/stretchr/testify>

24/10/2020



Going Into Production

Salman Shah

salman.shah@myntra.com

Twitter: [sbshah97](#)

Github: [salman-bhai](#)

Visit us at myntra.com

Powered by

develer