

# 5 - Frameworks and Middleware

## Echo Framework

### Frameworks Pros and Cons

- Reasons
  - Limited development resources
  - Limited pool of ideas
- Cons:
  - Unknown code quality
  - Licensing can be a hassle
- Pros:
  - Larger feature set and expertise
  - Easier to get support, fixes, and new developers

### Golang Web Frameworks

- Gorilla - large and not very performant
- httprouter - a very fast router
- Gin - quick and extensible
- Echo - light and the quickest out there

### First Steps in Echo

```
1 $ go get github.com/labstack/echo/...
```

src/echo/main.go

```
1 package main
2
3 import (
4     "net/http"
5
6     "github.com/labstack/echo"
7 )
8
9 func root(c echo.Context) error {
10     return c.String(http.StatusOK, "Running API v1")
11 }
12
13 func main() {
14     e := echo.New()
15 }
```

```
16     e.GET("/", root)
17
18     e.Start(":8080")
19 }
```

## Recreating Our API in Echo

### Middleware

- Middleware is a piece of functionality between the request and the response
- A sample middleware chain:



