Routing Requests

Michael Van Sickle

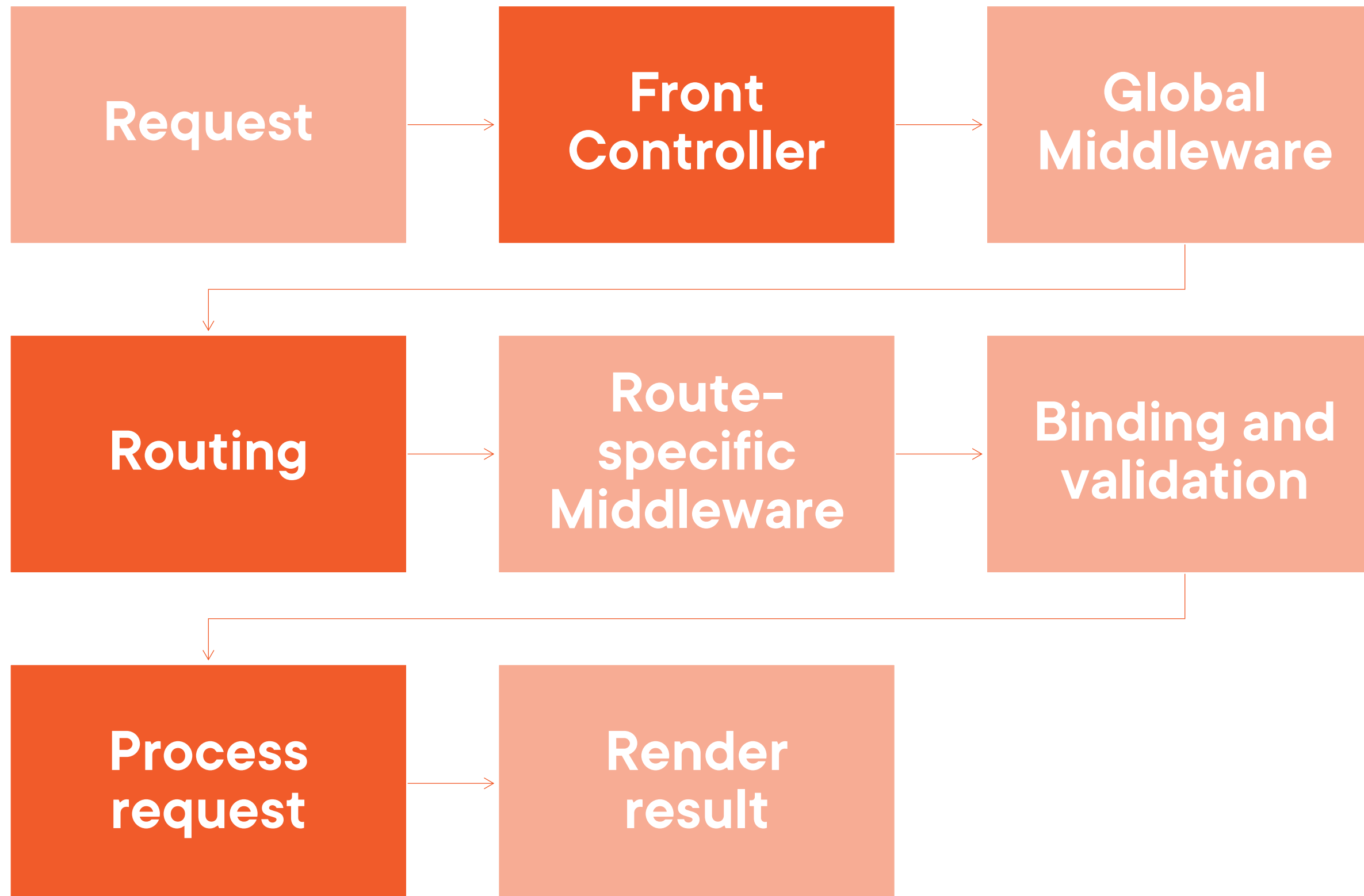@vansimke

HTTP Request Pipeline

Request	→	Front Controller	→	Global Middleware

Routing	→	Route-specific Middleware	→	Binding and validation

Process request	→	Render result

HTTP Request Pipeline

```
Request → Front Controller → Global Middleware
                                    ↓
Routing → Route-specific Middleware → Binding and validation
                                             ↓
Process request → Render result
```
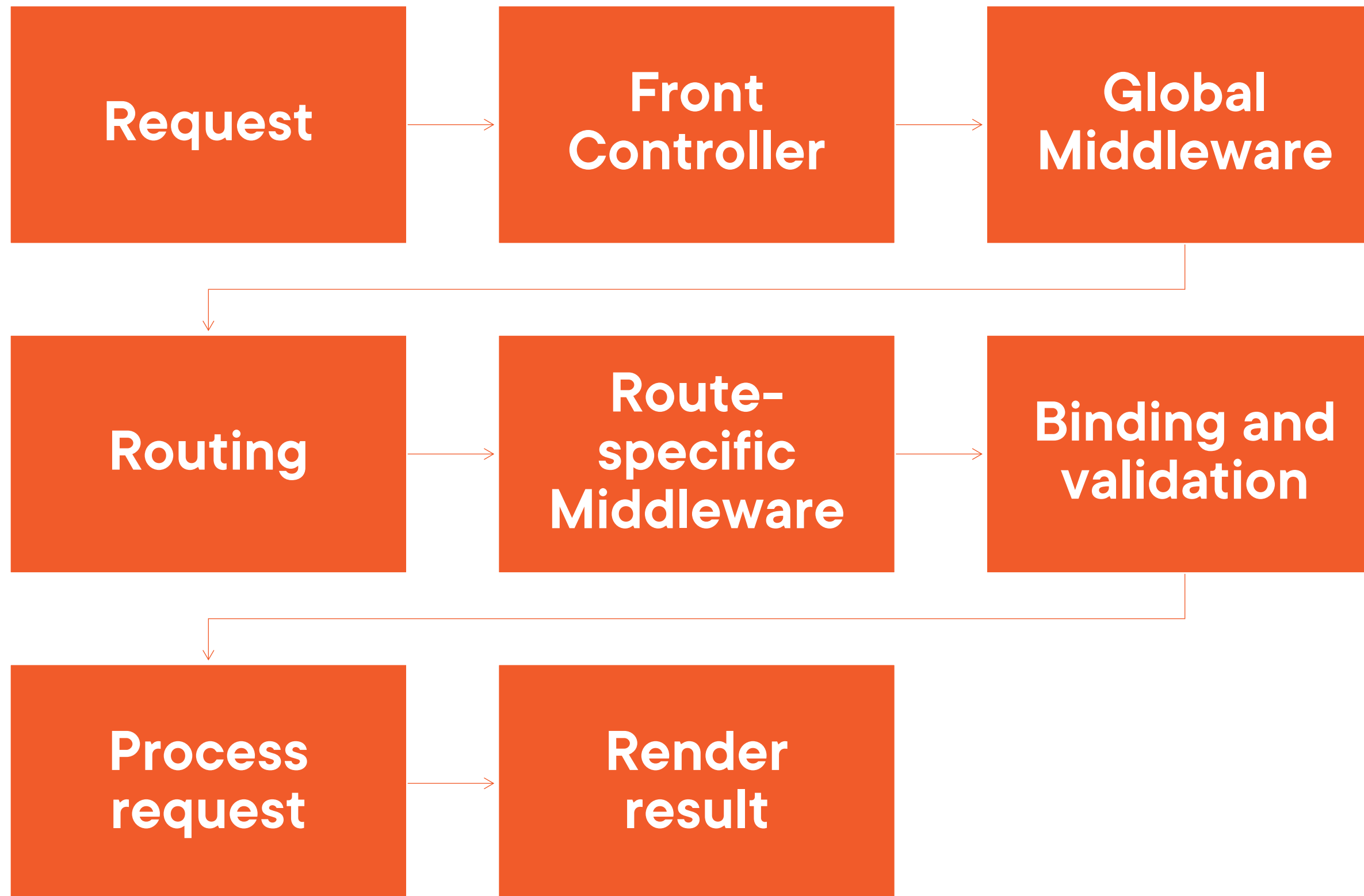
Overview

Static routes

Serving files

HTTP methods

Parameterized routes

Route groups

Static Routes

```
router := gin.Default()        // create a router

router.GET("/hello/world", func(c *gin.Context) {
    c.String("Hello, world!")
})
```

Serving Single Files

```go
router := gin.Default()        // create a router

router.StaticFile({URL path}, {path to file})
router.StaticFile("/favicon.ico", "./public/img/favicon.ico")
```

File Servers

```go
router := gin.Default()          // create a router

router.Static({base URL path}, {path to served directory})
router.Static("/static", "./public")

router.StaticFS({base URL path}, {http.FileSystem to serve})
router.StaticFS("/static", http.Dir("./public"))
```

HTTP Methods

```
// HTTP/2 message headers
:method = GET
:scheme = https
:path = /users
host = vacationtrack.info
<snip>
```

HTTP Methods

```
// HTTP/2 message headers
:method = GET
:scheme = https
:path = /users
host = vacationtrack.info
<snip>
```

```
router := gin.Default()
router.GET({URL Pattern},              ◄ GET

    func(c *gin.Context){

        ...

    },
)
router.POST(...)                       ◄ POST

router.PUT(...)                        ◄ PUT

router.DELETE(...)                     ◄ DELETE

router.PATCH(...)                      ◄ PATCH

router.HEAD(...)                       ◄ HEAD

router.OPTIONS(...)                    ◄ OPTIONS

router.Handle(httpMethod, ...)         ◄ Arbitrary method

router.Any(...)                        ◄ Any HTTP Method
```

Parameterized Routes

```
// https://vacationtrack.info/employees/ajane

router.GET("/employees/:username", func(c *gin.Context) {
    username := c.Param("username")              // ajane
})

// https://vacationtrack.info/employees/ajane/timeoff/scheduled

router.GET("/employees/:username/*rest", func(c *gin.Context) {
    username := c.Param("username")              // ajane
    rest := c.Param("rest")                      // timeoff/scheduled

    wholeroute := c.FullPath()                   // /ajane/timeoff/scheduled
})
```

Route Groups

```
https://vacationtrack.info/admin/users
https://vacationtrack.info/admin/roles
https://vacationtrack.info/admin/policies


router := gin.Default()
g := router.Group("/admin", ...gin.HandlerFunc)
g.GET("/users", ...)
g.GET("/roles", ...)
g.GET("/policies", ...)
```

Summary

Static routes

Serving files

HTTP methods

Parameterized routes

Route groups