Routing Requests

Michael Van Sickle

@vansimke

Introduction

Role of a controller

Static routing

Parametric routing

Third-party routers

Role of a Controller

```
func main() {
    templates := populateTemplates()
    http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
        requestedFile := r.URL.Path[1:]
        template := templates[requestedFile+".html"]
        var context interface{}
        switch requestedFile {
        case "shop":
            context = viewmodel.NewShop()
        default:
            context = viewmodel.NewHome()
        }
        if template != nil {
            err := template.Execute(w, context)
            if err != nil {
                log.Println(err)
            }
        } else {
            w.WriteHeader(404)
        }
    })
    http.Handle("/img/", http.FileServer(http.Dir("public")))
    http.Handle("/css/", http.FileServer(http.Dir("public")))
    http.ListenAndServe(":8000", nil)
}
```

Role of a Controller

```
func main() {
    templates := populateTemplates()
    http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
        requestedFile := r.URL.Path[1:]
        template := templates[requestedFile+".html"]
        var context interface{}
        switch requestedFile {
        case "shop":
            context = viewmodel.NewShop()
        default:
            context = viewmodel.NewHome()
        }
        if template != nil {
            err := template.Execute(w, context)
            if err != nil {
                log.Println(err)
            }
        } else {
            w.WriteHeader(404)
        }
    })
    http.Handle("/img/", http.FileServer(http.Dir("public")))
    http.Handle("/css/", http.FileServer(http.Dir("public")))
    http.ListenAndServe(":8000", nil)
}
```

Setup

Role of a Controller

```go
func main() {
    templates := populateTemplates()
    http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
        requestedFile := r.URL.Path[1:]
        template := templates[requestedFile+".html"]
        var context interface{}
        switch requestedFile {
        case "shop":
            context = viewmodel.NewShop()
        default:
            context = viewmodel.NewHome()
        }
        if template != nil {
            err := template.Execute(w, context)
            if err != nil {
                log.Println(err)
            }
        } else {
            w.WriteHeader(404)
        }
    })
    http.Handle("/img/", http.FileServer(http.Dir("public")))
    http.Handle("/css/", http.FileServer(http.Dir("public")))
    http.ListenAndServe(":8000", nil)
}
```

Setup

Serve pages

Role of a Controller

```
func main() {
    templates := populateTemplates()
    http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
        requestedFile := r.URL.Path[1:]
        template := templates[requestedFile+".html"]
        var context interface{}
        switch requestedFile {
        case "shop":
            context = viewmodel.NewShop()
        default:
            context = viewmodel.NewHome()
        }
        if template != nil {
            err := template.Execute(w, context)
            if err != nil {
                log.Println(err)
            }
        } else {
            w.WriteHeader(404)
        }
    })
    http.Handle("/img/", http.FileServer(http.Dir("public")))
    http.Handle("/css/", http.FileServer(http.Dir("public")))
    http.ListenAndServe(":8000", nil)
}
```

Setup

Serve pages

Static resources

Role of a Controller

```
http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
    requestedFile := r.URL.Path[1:]
    template := templates[requestedFile+".html"]
    var context interface{}
    switch requestedFile {
    case "shop":
        context = viewmodel.NewShop()
    default:
        context = viewmodel.NewHome()
    }
    if template != nil {
        err := template.Execute(w, context)
        if err != nil {
            log.Println(err)
        }
    } else {
        w.WriteHeader(404)
    }
})
```

Role of a Controller

General

```
http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
    requestedFile := r.URL.Path[1:]
    template := templates[requestedFile+".html"]
    var context interface{}
    switch requestedFile {
    case "shop":
        context = viewmodel.NewShop()
    default:
        context = viewmodel.NewHome()
    }
    if template != nil {
        err := template.Execute(w, context)
        if err != nil {
            log.Println(err)
        }
    } else {
        w.WriteHeader(404)
    }
})
```

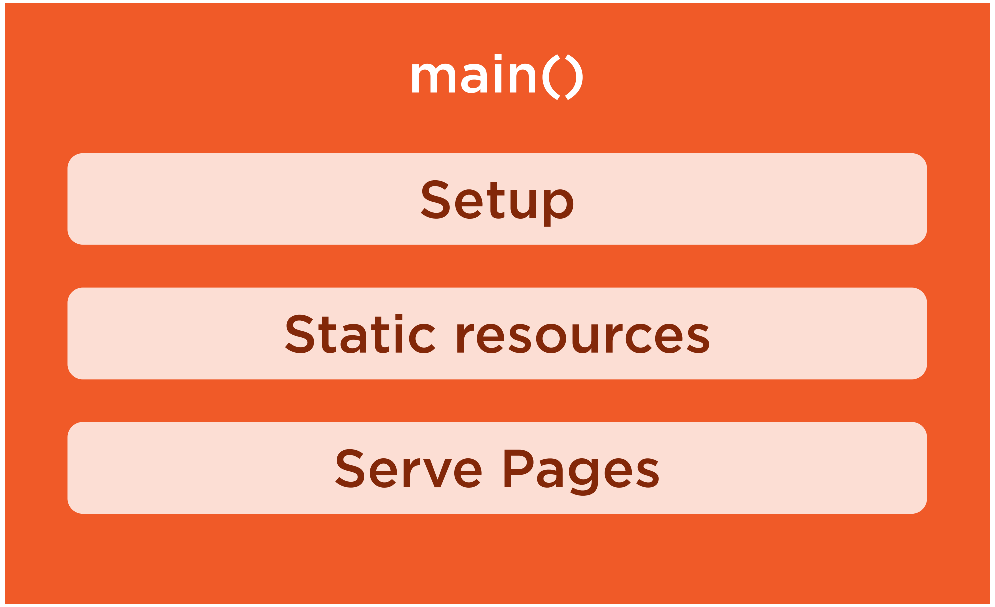
Role of a Controller

```
http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
    requestedFile := r.URL.Path[1:]
    template := templates[requestedFile+".html"]
    var context interface{}
    switch requestedFile {
    case "shop":
        context = viewmodel.NewShop()
    default:
        context = viewmodel.NewHome()
    }
    if template != nil {
        err := template.Execute(w, context)
        if err != nil {
            log.Println(err)
        }
    } else {
        w.WriteHeader(404)
    }
})
```

General

Page specific

Role of a Controller



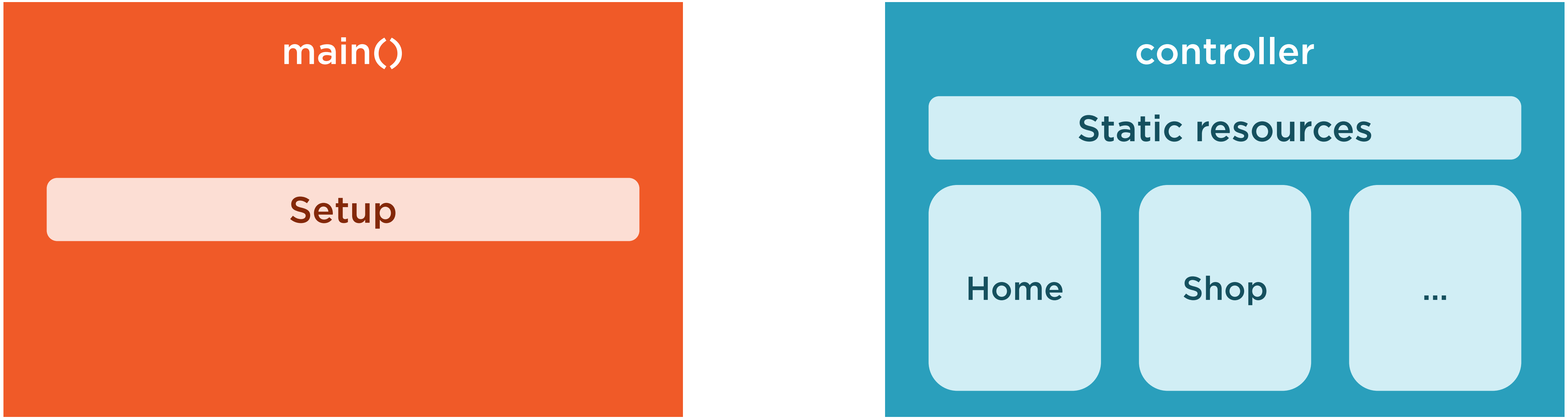
main()

Setup

Static resources

Serve Pages

Role of a Controller



main()

Setup

controller

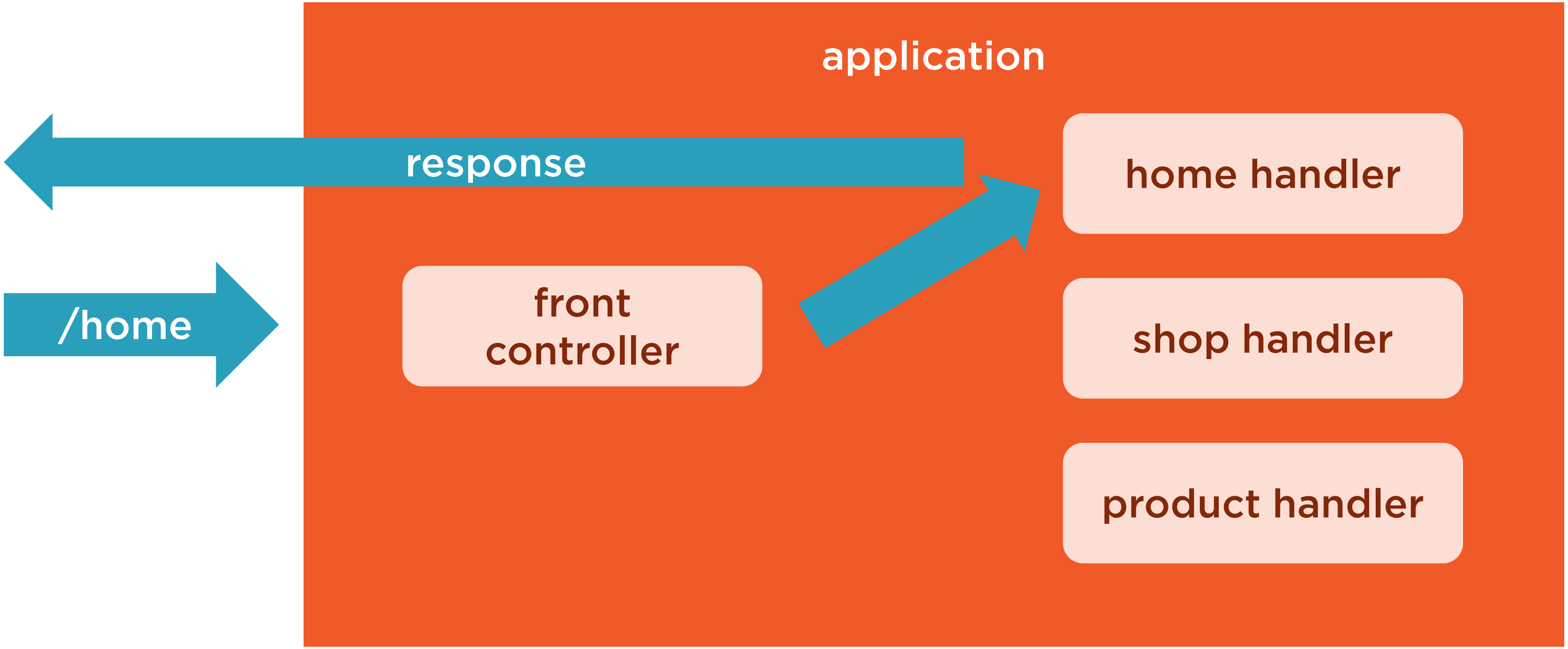
Static resources

Home

Shop

...

Static Routing



Handler with most specific match handles request

Parametric Routing

```
/home

/shop


/products/lemons

/products/oranges
```

Static routes – one handler per page

Parametric Routing

```
/home

/shop
```

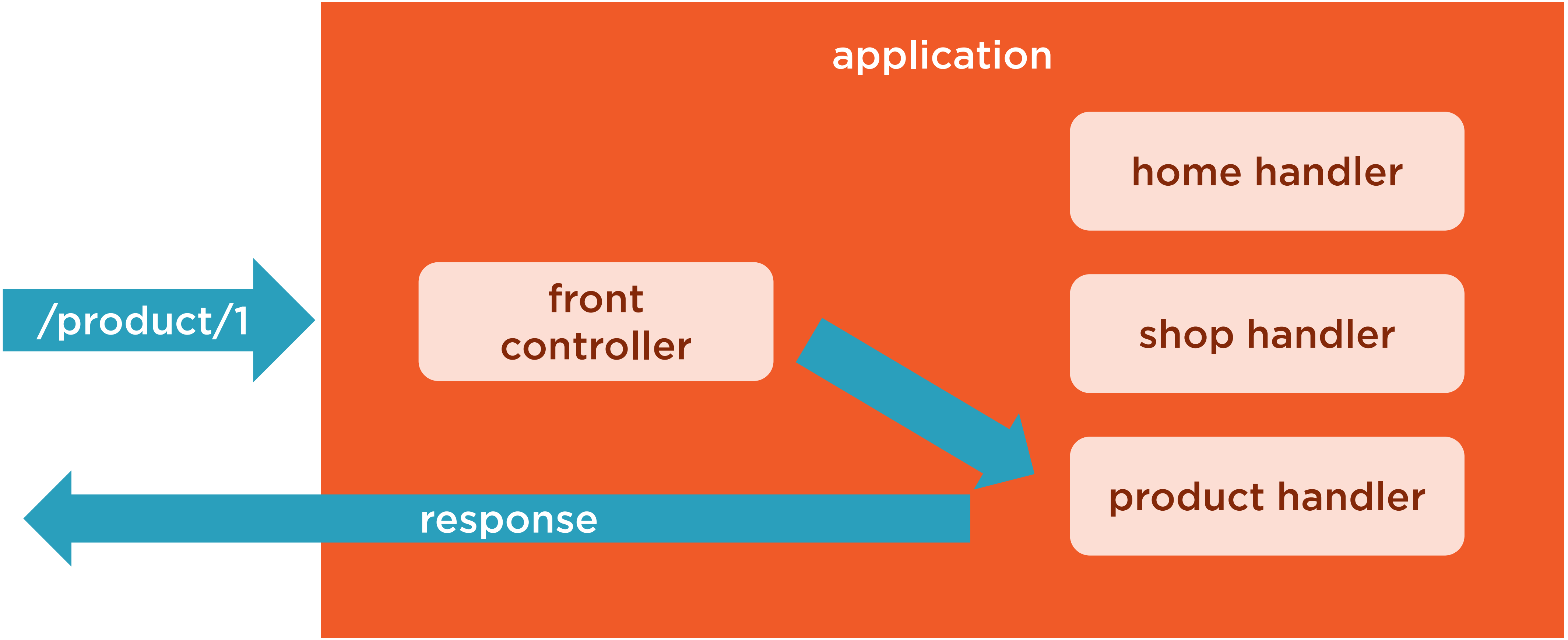
Static routes – one handler per page

```
/products/lemons

/products/oranges
```

Parametric routes – family of pages

Parametric Routing



Handler responsible for working with parameters

Third-party Routers

gorilla/mux

https://github.com/gorilla/mux

Highly configurable

httprouter

https://github.com/julienschmidt/httprouter

Extremely fast

Summary

Role of a controller

Static routing

Parametric routing

Third-party routers