

# Creating Middleware

---



**Michael Van Sickle**

@vansimke



# Introduction



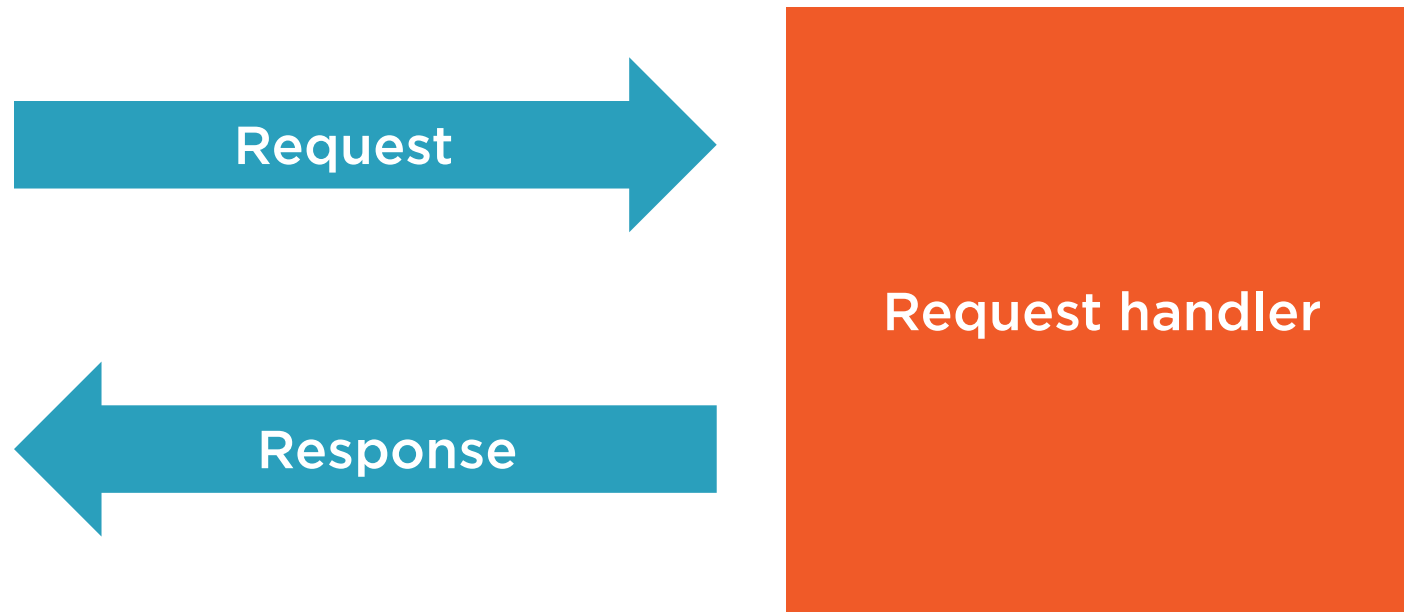
**What is middleware?**

**Creating middleware**

**Request contexts**



# What is Middleware?



# What is Middleware?



# Creating Middleware

```
http.ListenAndServe(addr string, handler Handler) error
```

```
type Handler interface {  
    ServeHTTP(ResponseWriter, *Request)  
}
```



# Creating Middleware

```
type MyMiddleware struct {  
    Next http.Handler  
}  
  
func (m MyMiddleware) ServeHTTP(w http.ResponseWriter, r *http.Request) {  
    // do things before next handler  
    m.Next.ServeHTTP(w, r)  
    // do things after next handler  
}
```



# Uses for Middleware

Logging

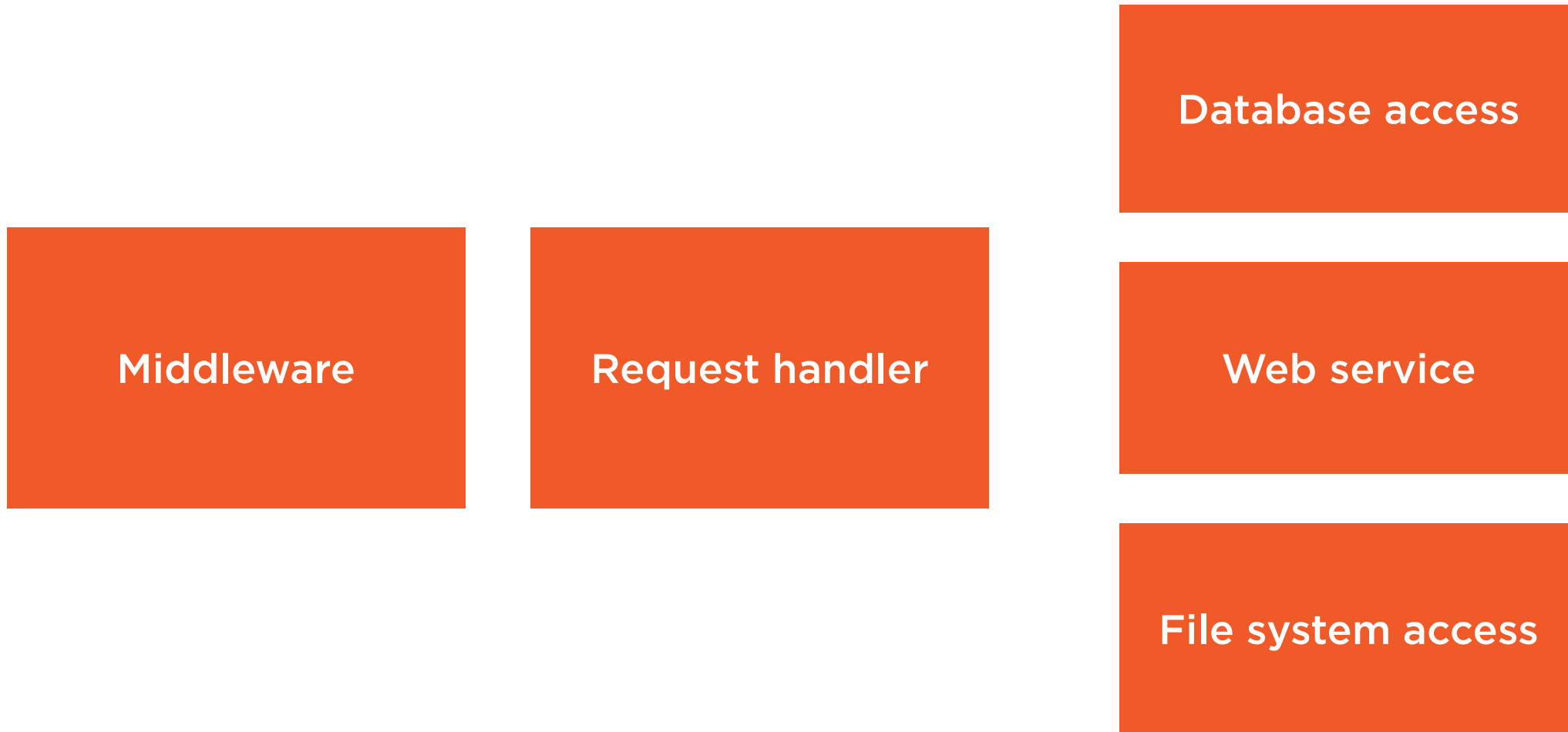
Security

Request timeouts

Response compression



# Using Request Contexts





# Using Request Contexts

```
func (*Request) Context() context.Context
```

```
func (*Request) WithContext(ctx context.Context) context.Context
```



# Context API

```
type Context interface {  
    Deadline() (deadline time.Time, ok bool)  
  
    Done() <-chan struct{}  
  
    Err() error  
  
    Value(key interface{}) interface{}  
}
```



# Context API

`WithCancel()`

`WithDeadline()`

`WithTimeout()`

`WithValue()`



# Summary



**What is middleware?**

**Creating middleware**

**Request contexts**

