



JÖNKÖPING UNIVERSITY

*School of Engineering*

---

# EXPRESS BASICS

**Peter Larsson-Green**

Jönköping University

Autumn 2018

# WEB APPS IN NODE.JS

Why not use the `http` module?

- Requires a lot of work 😞

We can start using a framework instead.

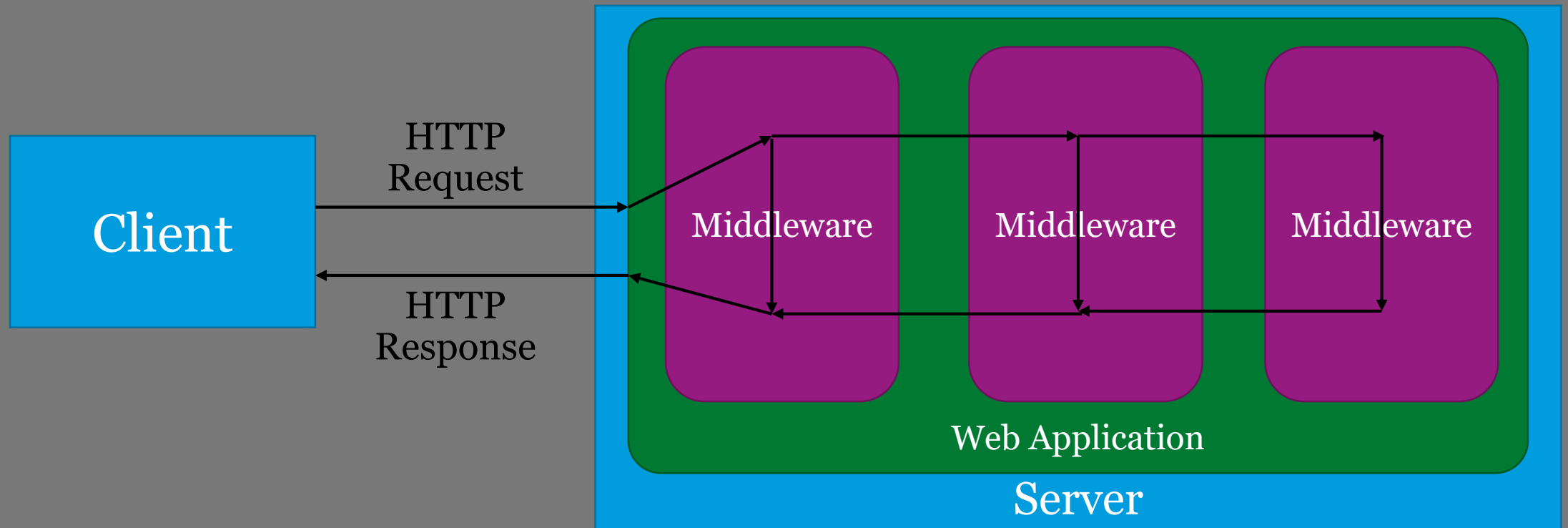
- Learning a framework takes time 😞
- Using a framework saves time 😊
  - Need to write less code (less time spent on writing code)
  - Less things can go wrong
  - Less time spent on testing
  - ...

# EXPRESS

A framework for web applications in Node.js.

- Website: <https://expressjs.com>
- Is distributed as an npm package.
  - `npm install express`

# MIDDLEWARE ARCHITECTURE



# EXPRESS MIDDLEWARES

```
const express = require('express')
const app = express()
app.use(function(request, response, next) {
  if (weHandleTheRequestHere) {
    response.send("This is the body of the HTTP response")
  } else {
    next()
  }
})
app.listen(8000)
```

The Application object: <http://expressjs.com/en/4x/api.html#app>



# EXPRESS MIDDLEWARES

```
const express = require('express')
const app = express()
app.use(function(request, response, next){
  if(request.method == "GET" && request.uri == "/hello"){
    response.send("Hello!")
  }else{
    next()
  }
})
app.listen(8000)
```

The Application object: <http://expressjs.com/en/4x/api.html#app>

# EXPRESS MIDDLEWARES

```
const express = require('express')
const app = express()
app.get('/hello', function(request, response) {
  response.send("Hello!")
})
app.get('/hi', function(request, response) {
  response.send("Hi!")
})
app.listen(8000)
```



# EXPRESS REQUESTS

request in Express  $\supset$  request in Node.js

```
app.get('/hello', function(request, response) {  
  const clientIp = request.ip  
  const queryStringObject = request.query  
})
```

```
GET /the-uri?what=this&that=8 HTTP/1.1  
...
```

```
request.query = {what: "this", that: "8"}
```

The Request object: <https://expressjs.com/en/api.html#req>

# ROUTING PARAMETERS

Useful for dynamic URIs.

- GET /humans/0 → Info about Alice.
- GET /humans/1 → Info about Bob
- ...

```
const humans = [{
  id: 0,
  name: "Alice"
}, {
  id: 1,
  name: "Bob"
}]
```

```
app.get('/humans/:id', function(request, response) {
  const id = request.params.id // E.g. '1'.
})
```

# EXPRESS RESPONSES

response in Express  $\supset$  response in Node.js

```
app.get('/hello', function(request, response) {  
  response.sendFile("the-file.html")  
  response.download("the-file.html")  
})
```

The Response object: <https://expressjs.com/en/api.html#res>

# EXPRESS MIDDLEWARES

```
const express = require('express')
const app = express()

app.use(function(request, response, next) {
  request.stupidExample = 15
  next()
})

app.get('/hello', function(request, response) {
  // We can use request.stupidExample here!
})
```

# SIMPLE AS THAT!

What remains to learn?

- Different middleware functions you can use.
  - Commonly used middleware functions:  
<https://expressjs.com/en/resources/middleware.html>
  - Many other npm packages contains middleware functions.
- More advanced features in Express (i.e. `response.render()`).