



JÖNKÖPING UNIVERSITY

School of Engineering

WEB APPLICATIONS IN EXPRESS

Peter Larsson-Green

Jönköping University

Autumn 2018

WHAT IS THE MVC PATTERN?

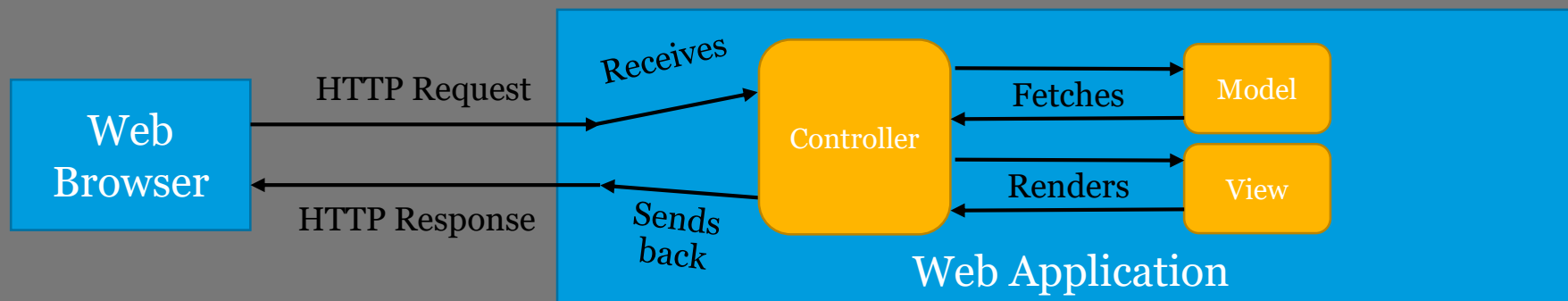
A design pattern common in implementations of GUIs.

Separate your code into three parts:

- **Model:** The part representing the data to be visualized.
- **View:** The part visualizing the Model.
- **Controller:** The part making it all happen.
 1. Creates/fetches the Model.
 2. Instantiates the View and gives it the Model.
 3. Handles user interaction with the View (e.g. clicks on buttons).

THE MVC PATTERN FOR WEB APPS

- The Model (the data) is typically stored in a database.
- The View generates HTML code.
- The Controller:
 1. Receives incoming HTTP requests.
 2. Fetches the Model from the database.
 3. Asks the View to generate the HTML code for the fetched Model.
 4. Sends back an HTTP response with the HTML code the View generated.



MVC IN EXPRESS EXAMPLE

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

```
const app = express()
```

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

```
app.get('/humans/1', function(request, response) {
```

```
  const model = humans[1]
```

```
  response.render('human.hbs', model)
```

```
})
```

```
<h1>{{name}}</h1>
```

```
<p>{{name}} has id {{id}}.</p>
```

/views/human.hbs



Sent to
client.

```
<h1>Bob</h1>
```

```
<p>Bob has id 1.</p>
```

Models

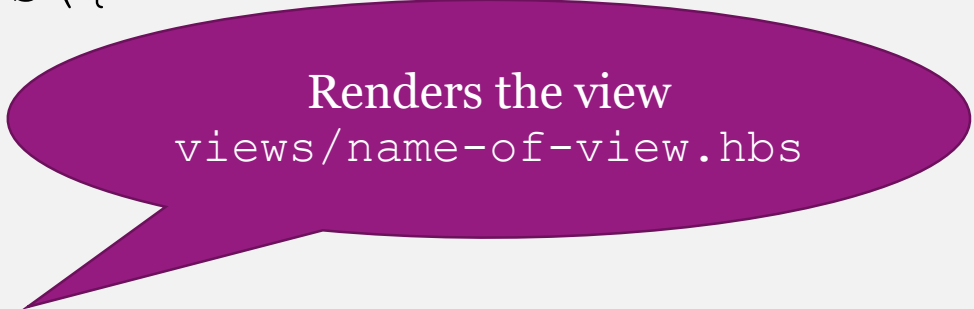
Cont-
roller

View

USING HANDLEBARS IN EXPRESS

```
npm install express-handlebars
```

```
const express = require('express')  
const expressHandlebars = require('express-handlebars')  
  
const app = express()  
  
app.engine('hbs', expressHandlebars({  
  extname: ".hbs"  
}))
```



Renders the view
views/name-of-view.hbs

```
// Use response.render("name-of-view.hbs") in your callbacks.
```

LAYOUTS IN EXPRESS-HANDLEBARS

```
<!DOCTYPE html>
<html>
  <head>
    <title>Website</title>
  </head>
  <body>
    <h1>About</h1>
    <p>About us...</p>
  </body>
</html>
```

/views/about.hbs

```
<!DOCTYPE html>
<html>
  <head>
    <title>Website</title>
  </head>
  <body>
    <h1>Contact</h1>
    <p>Contact us...</p>
  </body>
</html>
```

/views/contact.hbs

LAYOUTS IN EXPRESS-HANDLEBARS

```
<h1>About</h1>
```

```
<p>About us...</p>
```

```
/views/about.hbs
```

```
<h1>Contact</h1>
```

```
<p>Contact us...</p>
```

```
/views/contact.hbs
```

```
app.engine('hbs', expressHandlebars({
  extname: ".hbs",
  defaultLayout: "blue", // views/layouts/blue.hbs
}))
app.get("/about", function(request, response) {
  response.render("about.hbs", {})
})
```

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Webite</title>
```

```
  </head>
```

```
  <body>
```

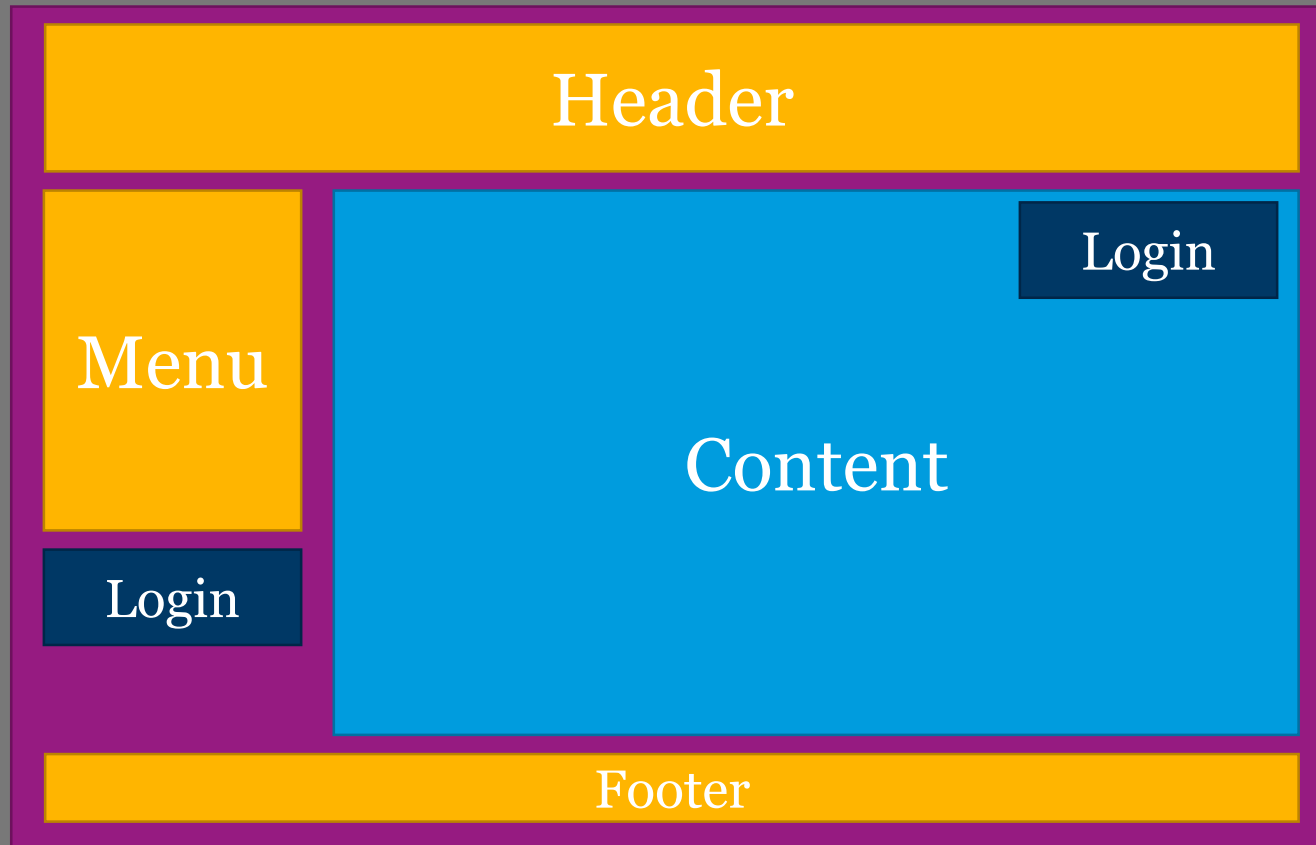
```
    {{{body}}}
```

```
  </body>
```

```
</html>
```

```
/views/layouts/blue.hbs
```


PARTIALS IN EXPRESS-HANDLEBARS



PARTIALS IN EXPRESS-HANDLEBARS

```
<h1>Home</h1>
<p>Login using the form below.</p>
{{> login}}
```

/views/home.hbs

```
<h1>Login</h1>
<form>
  ...
</form>
```

/views/partials/login.hbs

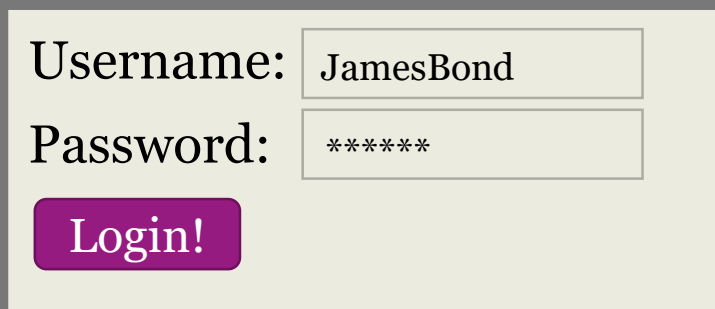


```
<h1>Home</h1>
<p>Login using the form below.</p>
<h1>Login</h1>
<form>
  ...
</form>
```

HANDLING HTML FORMS

```
<form method="GET" action="http://www.mi6.com/login">  
  Username: <input type="text" name="un"><br>  
  Password: <input type="password" name="pw"><br>  
  <input type="submit" value="Login!">  
</form>
```

Browser renders.



Username:

Password:

GET /login?un=JamesBond&pw=missMP HTTP/1.1
Host: www.mi6.com
...

User submits.

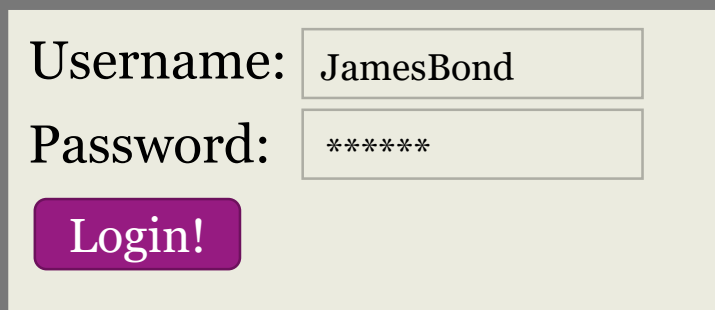
In Express, simply use:

- `request.query.un`
- `request.query.pw`

HANDLING HTML FORMS

```
<form method="POST" action="http://www.mi6.com/login">  
  Username: <input type="text" name="un"><br>  
  Password: <input type="password" name="pw"><br>  
  <input type="submit" value="Login!">  
</form>
```

Browser renders.



User submits.

```
POST /login HTTP/1.1  
Host: www.mi6.com  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 22  
...  
un=JamesBond&pw=missMP
```

In Express, use
the `body-parser`
middleware.

THE BODY-PARSER MIDDLEWARE

Installation: `npm install body-parser`

```
const express = require('express')
const app = express()
const bodyParser = require('body-parser')
app.use(bodyParser.urlencoded({extended: false}))

app.post('/login', function(request, response) {
  const username = request.body.un
  const password = request.body.pw
})
```

```
POST /login HTTP/1.1
Content-Type: application/x-www-
                form-urlencoded
...
un=JamesBond&pw=missMP
```

THE STATIC MIDDLEWARE

```
const express = require('express')  
const app = express()  
app.use(express.static("public"))
```

```
<link rel="stylesheet" href="/layout.css">
```

```

```

```
▲ PROJECT  
  ▲ public  
    # layout.css  
    🖼️ logo.png  
    JS app.js
```