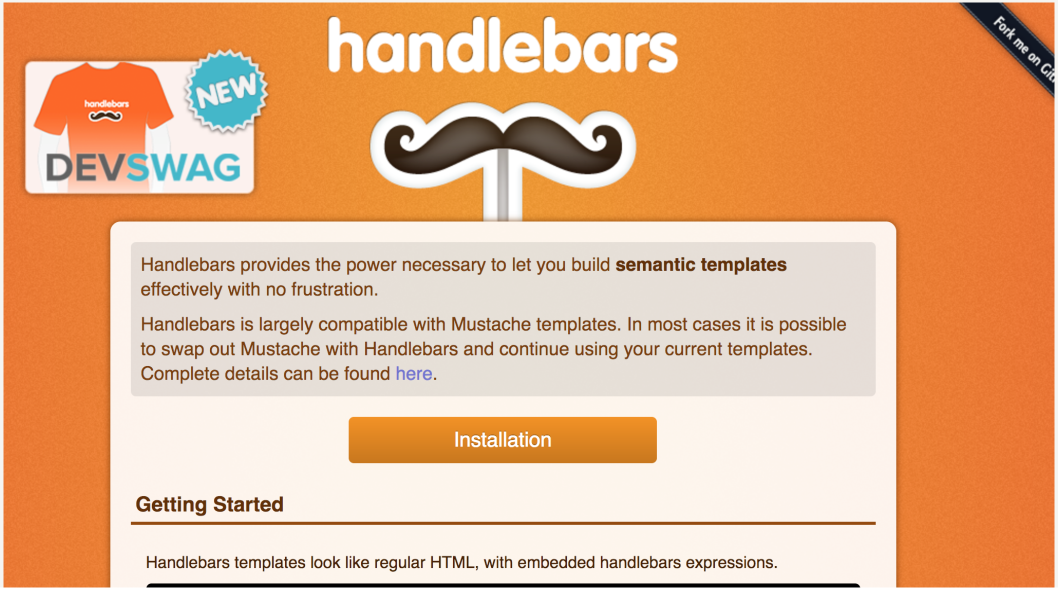
Express View Templates with Handlebars.js

Questions

- What is Handlebars?
- What is a template?
- How does express use a template?
- How do you pass information to a template?
- What is res.render?
- How do you show a variable in a template?
- How do you loop in a template?
- What is `layout.hbs` for?

Handlebars



Handlebars provides the power necessary to let you build **semantic templates** effectively with no frustration.

Handlebars is largely compatible with Mustache templates. In most cases it is possible to swap out Mustache with Handlebars and continue using your current templates. Complete details can be found here.

Getting Started

Handlebars templates look like regular HTML, with embedded handlebars expressions.

Mustache

Logic-less templates.

Available in Ruby, JavaScript, Python, Erlang, node.js, PHP, Perl, Perl6, Objective-C, Java, C#/.NET, Android, C++, CFEngine, Go, Lua, ooc, ActionScript, ColdFusion, Scala, Clojure[Script], Fantom, CoffeeScript, D, Haskell, XQuery, ASP, Io, Dart, Haxe, Delphi, Racket, Rust, OCaml, Swift, Bash, Julia, R, Crystal, Common Lisp, Nim, Smalltalk, Tcl, C, and for ABAP

"logic-less templates"

- for displaying data
- not for your application code

An example template

```html
<h1>Hey Everybody!</h1>
<h2>Templates</h2>
<ul>
  <li>are basically HTML</li>
  <li>but can do more</li>
</ul>
```

Really?

(yes)

Providing data to a template

```
router.get('/:id', function(req, res, next) {
  Member.findOne({
    where: {
      id: req.params.id
    }
  })
  .then((result) => {
    console.log(result);
    res.render('member-profile', {
       member: result
     });
  });
});
```

Grabbing data from "object paths"

```
<h1>{{ member.surname }}, {{ member.firstname }}</h1>
<p>
  Address: {{member.address}}
</p>
<p>
  Zip Code: {{member.zipcode}}
</p>
<p>
  Telephone: {{member.telephone}}
</p>
```

Comments

```
{{! This is just a comment. }}
```

Helpers

Providing a collection (an array)

```
router.get('/', function(req, res, next) {
  Member.findAll()
    .then((result) => {
      res.render('member-list', {
        members: result
      });
    });
});
```

#each

```
{{#each members}}
  <h1>{{surname}}, {{firstname}}</h1>
  <p>
    Address: {{address}}
  </p>
  <p>
    Zip Code: {{zipcode}}
  </p>
  <p>
    Telephone: {{telephone}}
  </p>
{{/each}}
```

Accessing parent paths within a helper block

```
router.get('/', function(req, res, next) {
  Member.findAll()
    .then((result) => {
      res.render('member-list', {
        members: result,
        homeLink: 'https://the.club/'
      });
    });
});
```

"If I'm looping through `members` how do I access `homeLink`?"

#each

```
{{#each members}}
  <h1>{{surname}}, {{firstname}}</h1>
  <p>
    Address: {{address}}
  </p>
  <p>
    Zip Code: {{zipcode}}
  </p>
  <p>
    Telephone: {{telephone}}
  </p>
  <p>
    <a href="{{../homeLink}}">Return to the.club</a>
  </p>
{{/each}}
```

`../` refers to the parent object

#if

```
<h1>{{ member.surname }}, {{ member.firstname }}</h1>
{{#if member.address}}
<p>
  Address: {{member.address}}
</p>
{{/if}}
{{#if member.zipcode}}
<p>
  Zip Code: {{member.zipcode}}
</p>
{{/if}}
{{#if member.telephone}}
<p>
  Telephone: {{member.telephone}}
</p>
{{/if}}
```

#unless

```
<h1>{{ member.surname }}, {{ member.firstname }}</h1>
{{#unless member.bookings}}
  <p>Make your first booking!</p>
{{/unless}}
```

#with

```
{{#with member}}
  <h1>{{ surname }}, {{ firstname }}</h1>
  <p>
    Address: {{address}}
  </p>
  <p>
    Zip Code: {{zipcode}}
  </p>
  <p>
    Telephone: {{telephone}}
  </p>
{{/with}}
```

Layouts

Default layout: layout.hbs

```
<!DOCTYPE html>
<html>
  <head>
    <title>{{title}}</title>
    <link rel="stylesheet" href="/stylesheets/style.css" />
  </head>
  <body>
    {{{body}}}
  </body>
</html>
```

You can customize the HTML skeleton

home.hbs

```
<!DOCTYPE html>
<html>
  <head>
    <title>{{title}}</title>
    <link rel="stylesheet" href="/stylesheets/style.css" />
  </head>
  <body>
    <h1>yeah boyee</h1>
    {{{body}}}
  </body>
</html>
```

Using a layout file

```
router.get('/', function(req, res, next) {
  res.render('index', {
    layout: 'home',
    title: 'Express',
  });
});

`
```

Partials

express-handlebars

More powerful than the builtin one

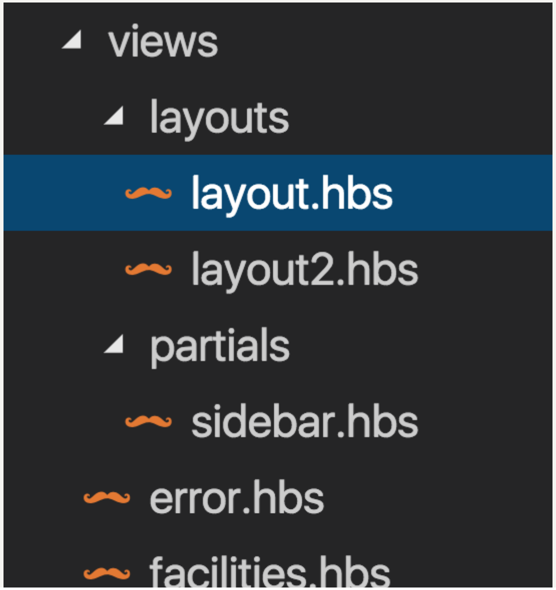
```
npm install --save express-handlebars
```

Using express-handlebars

```
var expressHbs = require('express-handlebars');

//app.set('views', path.join(__dirname, 'views'));
app.engine('.hbs', expressHbs({defaultLayout: 'layout', extname: '.hbs'}));
app.set('view engine', '.hbs');
```

Updated views directory structure



```
◢ views
   ◢ layouts
        layout.hbs
        layout2.hbs
   ◢ partials
        sidebar.hbs
     error.hbs
     facilities.hbs
```

Sidebar.hbs

```
<div>
  <ul>
    <li>
      <a href="#">thing1</a>
    </li>
    <li>
      <a href="#">thing2</a>
    </li>
    <li>
      <a href="#">thing3</a>
    </li>
  </ul>
</div>
```

Home.hbs

```
<!DOCTYPE html>
<html>
<head>
  <title>{{title}}</title>
  <link rel='stylesheet' href='/stylesheets/style.css' />
</head>
<body>
  <h1>yeah boyee</h1>
  {{{body}}}
  {{> sidebar }}
</body>
</html>
```

Partials are included with `{{> (name-of-partial)}}`

Questions

- What is Handlebars?
- What is a template?
- How does express use a template?
- How do you pass information to a template?
- What is res.render?
- How do you show a variable in a template?
- How do you loop in a template?
- What is `layout.hbs` for?

Express View Templates with Handlebars.js