6.148

Templating and Handlebars

Monde Duinkharjav
A quick recap

Tuesday Webpage components: HTML, CSS, JS
Wednesday Sending a webpage: Node.js + Express.js, AJAX
This morning Storing information: MongoDB
A quick recap

Tomorrow: no class; Mystery Hunt

Saturday and Sunday: Milestone 1: Project Pitches
Milestone 1: Project Pitches

- Pitch one idea to staff and sponsors
- Present wireframes of what your web app would look like
- We will give you feedback on your idea
- Suggest useful APIs that the web app could use
- Warn about potential technical difficulties you may face in development
- Give ideas on potential features to consider etc.
Milestone 1: Project Pitches

Project Pitches will be in 32-044 from 1-5PM Saturday and Sunday

It will be 15 minutes long.

Plan to arrive 15 minutes before your team’s slot.

Sign ups will be up by tonight.
Next week

Monday, no lecture. MLK holiday

Guest lectures on useful libraries, services and APIs

Your web apps may benefit a lot from some of them

One lecture and one workshop on Authentication and Security

You probably want to start working on your project early this week

Hackathon Friday night: opportunity to implement a key function for the webapp
Templating
Why handlebars?

1. Code is more structured, easy to read, natural and iterable
2. Reduce code written by you
Profile

Name: Monde

Role: Lecturer
Dynamic HTML

<html>
  <body>
    <h1>Profile</h1>
    <p>Name: Monde</p>
    <p>Role: Lecturer</p>
  </body>
</html>
Handlebars

<html>
<body>
<h1>Profile</h1>
<p>Name: {{ name }}</p>
<p>Role: {{ role }}</p>
</body>
</html>
Handlebars

```html
<html>
  <body>
    <h1>Profile</h1>
    <p>Name: {{ name }}</p>
    <p>Role: {{ role }}</p>
  </body>
</html>

{
  name: "Monde",
  role: "Lecturer"
}
```
If we didn’t use handlebars

```javascript
var user = { name: "Monde", role: "lecturer" }

var html_string = "<body> <h1> Profile </h1>";
html_string += "<p> Name :" + user.name + "</p>";
html_string += "<p> Role:" + user.role + "</p>";
html_string += "</body>";

$(html).html(html_string);
```
Handlebar conditionals

{{#if isLoggedIn }}
  <a href=''>Profile</a>
  <a href=''>Log out</a>
{{else}}
  <a href=''>Log in</a>
{{/if}}
Handlebar conditionals

{{#if isLoggedIn }}
  <a href="">Profile</a>
  <a href="">Log out</a>
{{else}}
  <a href="">Log in</a>
{{/if}}

{ isLoggedIn: true }
Handlebar conditionals

{{#if isLoggedIn }}
  <a href="">Profile</a>
  <a href="">Log out</a>
{{/if}}

{{else}}
  <a href="">Log in</a>
{{/if}}

{ isLoggedIn: true }
Handlebar forEach

{{#each recentComments}}
  <h1>{{ user }}</h1>
  <p>{{ body }}</p>
{{/each}}
Handlebar forEach

```handlebars
{{#each recentComments}}
  <h1>{{ user }}</h1>
  <p>{{ body }}</p>
{{/each}}
```

```javascript
{ recentComments:[
  { user: "Monde", body: "foo"},
  { user: "David", body: "bar"}
] }
```
Handlebar forEach

```handlebars
{{#each recentComments}}
  <h1>{{ user }}</h1>
  <p>{{ body }}</p>
{{/each}}

{ recentComments:
  [ { user: "Monde", body: "foo"},
    { user: "David", body: "bar"} ]
}
```
Breaking down your web app
Breaking down your web app
Breaking it down even further

You can create individual components that would appear on across different templates
Breaking it down even further

You can create individual components that would appear on across different templates
Breaking it down even further

You can create individual components that would appear on across different templates
Breaking it down even further

You can create individual components that would appear on across different templates
Making Components (or widgets)

File 1: navbar.hbs
Making Components (or widgets)

File 1: navbar.hbs

File 2: lectures.hbs

{{{ > navbar }}}
Making Components (or widgets)

File 1: navbar.hbs

File 2: lectures.hbs

File 3: schedule.hbs
Making Components (or widgets)

File 1: navbar.hbs

File 2: lectures.hbs

{{{ > navbar }}}

File 3: schedule.hbs

{{{ > navbar }}}
Providing the .hbs file the data

/* GET home page */
router.get('/', function (req, res, next) {
    res.render('index', { title: 'Sign Up'});
});

Here, index refers to the index.hbs file in views/

and the following JS object is the data that is used to render the final .html
Workshop!

go.6148.io/handlebars-workshop
Resources to learn

JS 101: CodeAcademy (JS)
Terminal 101: CodeAcademy (Terminal)

HTML, CSS, JS documentation: W3Schools

More documentation: jQuery, MongoDB, Handlebars

Debugging: Printing to the console, Chrome Inspector, Stack Overflow, Google

A brief crashcourse on using git: go.6148.io/learngit

Useful libraries, frameworks to look into:

CSS: Bootstrap, Semantic UI, Foundation

Front end: Angular.js, React.js, Require.js

Back end: Socket.io

Useful dev tools:

Sublime Package Manager, Postman, Nodemon
Personal Googling Tips

- Blog posts are easier to read and follow along
- Reading documentation, although discouraging, is often much better than digging through stack overflow questions
- Googling “blah doesn’t work” will not give you useful results