# **Templating**

Liam McLennan

@liammclennan

# Outline

- **Client-side templating**
- **Underscore.js templates**
- **Handlebars templates**
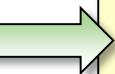- **Pre-compiling templates**

# Client-side Templating

- **Dynamically build markup from a template and some data**

- **All but the simplest UI elements will require templating**

- **Templating happens in the view's render method**

# Underscore templates

- **Backbone requires underscore, underscore can render templates**

- **There are three valid types of code blocks:**

- **<% … %>**

  □ Execute arbitrary code

- **<%= … %>**

  □ Evaluate an expression and render the result inline

- **<%- …%>**

  □ Evaluate an expression and render the html escaped result inline

# Underscore templates (cont.)

```
var V = Backbone.View.extend({
  render: function () {
    var data = { lat: -27, lon: 153 };
    this.$el.html(
      _.template('<%= lat %> <%= lon %>', data }));
    return this;
  }
});
```

The underscore.js API has since changed.
The correct use of the template function is now:

```
var template = _.template("<%= lat %> <%= lon %>")(data);
```

# Underscore templates (cont.)

```
<script id="latlon-template" type="text/html">
  <p>Latitude: <%= lat %></p>
  <p>Longitude: <%= lon %></p>
  <% _([1,2,3]).each(function (number) { %>
    <p>
      <%= number %>
    </p>
  <% }); %>
</script>
```

# Script Tag Templates

- **Templates can be stored inside script tags with a non-JavaScript 'type' attribute**

```html
<script id="latlon-template" type="text/html">
  <p>Latitude: <%= lat %></p>
  <p>Longitude: <%= lon %></p>
</script>
```

- **Accessed on demand with jQuery**

```javascript
var template = $('#latlon-template').html();
```

# Handlebars

- **Templating engine based on mustache**
- **Philosophically opposed to code in templates**
- **Code blocks are delimited by {{ .. }}**

```
<p>Latitude: {{lat}}</p>
<p>Longitude: {{lon}}</p>
{{#each numbers}}
  <p>
    {{this}}
  </p>
{{/each}}
```

substitution

looping

# Handlebars (cont.)

- **Rendering is a two stage process**
  - Compile
  - Execute

*Compile*

```
var source = '<p>Latitude: {{lat}}</p>';
var compiled = Handlebars.compile(source);
```

*Execute*

```
var rendered = compiled({lat: -27});
// rendered === <p>Latitude: -27</p>
```

# Pre-compiling Templates

- **Compiling a template == converting to a function**

*Underscore compilation*

```
var source = '<p>Latitude: <%= lat %></p>';
var compiled = _.template(source);
```

- **Pre-compile templates for performance**
- **Script compilation as a build step**

# Pre-compiling Templates (cont.)

- **Handlebars includes support for pre-compiling file-based templates**

Handlebars pre-compilation (console)

```
handlebars <input> -f <output>
```

# Summary

- **Templating is a core dependency for client-side applications**

- **Underscore.js includes templating and is required by Backbone.js**

- **Handlebars supports semantic templates**
    - Has a powerful pre-compilation tool