

**Rails + React(.js)**

# MVC

**Where is Rails lacking?**

1	Wookiee	10
2	Kosmo	14

## Task #1: Update match scores without a full page refresh

```
var match = {  
  id: 1,  
  player1_score: 20,  
  player2_score: 14  
}
```

```
$('#match1 .player1 .score').html(match.player1_score);  
$('#match1 .player2 .score').html(match.player2_score);
```

1	Wookiee	20
2	Kosmo	14

## Task #2: Also update player names and images

```
var match = {
  id: 1,
  player1_score: 20,
  player2_score: 14,
  player1: {
    name: "Wookiee Cat",
    imagePath: "/assets/images/wookiee.png"
  },
  player2: {
    name: "Kosmo",
    imagePath: null
  }
}
```

# Not looking so hot

```
$('#match1 .player1 .score').html(match.player1_score);
$('#match1 .player2 .score').html(match.player2_score);

$('#match1 .player1 .name').html(match.player1.name);
$('#match1 .player2 .name').html(match.player2.name);

var $p1Image = $('#match1 .player1 .image');
var $p2Image = $('#match1 .player2 .image');


if (match.player1.imagePath) {
    $p1Image.attr('src', match.player1.imagePath);
    $p1Image.show();
} else {
    $p1Image.hide();
}

if (match.player2.imagePath) {
    $p2Image.attr('src', match.player2.imagePath);
    $p2Image.show();
} else {
    $p2Image.hide();
}
```

# JS templates to the rescue

```
<script id="match-template" type="text/x-handlebars-template">
  <div class="match">
    <div class="player player1">
      {{#if match.player1.imagePath}}
        <img class="image" src={{match.player1.imagePath}} />
      {{/if}}
      <div class="name">{{match.player1.name}}</div>
      <div class="score">{{match.player1_score}}</div>
    </div>
    <div class="player player2">
      <!-- ... -->
    </div>
  </div>
</script>
```

```
var source = $("#match-template").html();
var template = Handlebars.compile(source);
var newHtml = template({match: match});
```

1	 Wookiee Cat	20
2	Kosmo	14

## Task #3: Highlight a player when clicked

```
$(".match .player").on("click", function() {  
    $(this).addClass("selected");  
});
```

**But what happens when the template is re-rendered?**

# Application State != UI State

```
var p1Selected = false;
var p2Selected = false;

$(".match .player1").on("click", function() {
  p1Selected = !p1Selected;
  render();
});

$(".match .player2").on("click", function() {
  p2Selected = !p2Selected;
  render();
});

function render(matchData) {
  matchData.p1Class = (p1Selected ? 'selected' : '');
  matchData.p2Class = (p2Selected ? 'selected' : '');
  return template(matchData);
}
```



**Templates - separation of concerns?**

# React.js

Build components, not templates.

```
var Player = require('./Player');

var Match = React.createClass({
  propTypes: {
    match: React.PropTypes.object.isRequired
  },

  render: function() {
    return (
      <div className="match">
        <Player
          player={this.props.match.player1}
          score={this.props.match.player1_score} />
        <Player
          player={this.props.match.player2}
          score={this.props.match.player2_score} />
      </div>
    )
  }
});
```



```
var Player = React.createClass({
  propTypes: {
    player: React.PropTypes.object.isRequired,
    score: React.PropTypes.number.isRequired
  },

  getDefaultState: function() {
    return {
      selected: false
    }
  },

  toggleSelected: function() {
    this.setState({selected: !this.state.selected});
  },

  render: function() {
    return (
      <div className="player" onClick={this.toggleSelected}>
        <div className="name">{this.props.player.name}</div>
        <div className="score">{this.props.score}</div>
      </div>
    )
  }
});
```

**Render after all state changes**

# Virtual DOM

# react-rails gem

```
react_component "Match", {match: {...}}
```

```
react_component "Match", {match: {...}}, prerender: true
```



# React Native

```
// iOS

var React = require('react-native');
var { TabBarIOS, NavigatorIOS } = React;

var App = React.createClass({
  render: function() {
    return (
      <TabBarIOS>
        <TabBarIOS.Item title="React Native" selected={true}>
          <NavigatorIOS initialRoute={{ title: 'React Native' }} />
        </TabBarIOS.Item>
      </TabBarIOS>
    );
  },
});
```



# Quick Demo

**Thank you!**

**@davecornelius**

**@challonge**

**@bracketsforgood**

**BFG work day - November 1st, 1pm**