Introduction to Backbone.js

8th Light University
Mike Jansen

Goals

An Immediate Digression on Javascript

Backbone.js

Jeremy Ashkenas

Structure

Model

```
var Task = Backbone.Model.extend({
  defaults: {
    'complete': false
  }
});
```

Model

```
var task = new Task({
  description: "Set up lightning rod on
  top of laboratory"
});
task.get("description");


> "set up lightning rod on top of
laboratory"
```

Collection

```
var Tasks = Backbone.Collection.extend({
  model: Todo.models.Task,
  url: "tasks"
});
```

Collection

```
var tasks = new Tasks();
tasks.create(
{
 description: "Dig up some brains at the
 cemetery"
});
```

Pick and Choose

Router

```
var TodoRouter = new
Backbone.Router.extend({
  initialize: function(options) {
   this.tasks = new Tasks(options.tasks)
   this.tasksView = new
   TasksView({collection: this.tasks});
  },
...
```

Router

```
...

routes: {
  "": "loadTasks",
  "tasks/:id/edit": "editTask"
},
```

Router

...

```
  editTask: function(id) {
   var view = new TaskEditView({id: id,
   collection: this.tasks});
   view.render();
  }
});
```

Router

```
Todo.router = new TodoRouter(options);
Backbone.history.start();
```

Router

www.drfrankenstein.com/todo#tasks/34/edit

So…

What about the DOM?

View

"...more convention than they are code"

View

```
var TasksView = Backbone.View.extend({
 tagName: 'div',
 className: 'task-list',

...
```

View

```
events: {
 'keyup #new-task' : 'createOnEnter'
},
```

View

```
createOnEnter : function (event) {
  ...
  this.collection.create({ description:
  inputValue });
  ...
},
```

View

```
render : function () {
 this.collection.each(this.addTask);
 return this;
},
```

View

```
addTask: function (task) {
 var view = new
  Todo.views.TaskView({ model: task});
 var taskEl = view.render().el;
 $('.tasks').prepend(taskEl);
},
```

Summary

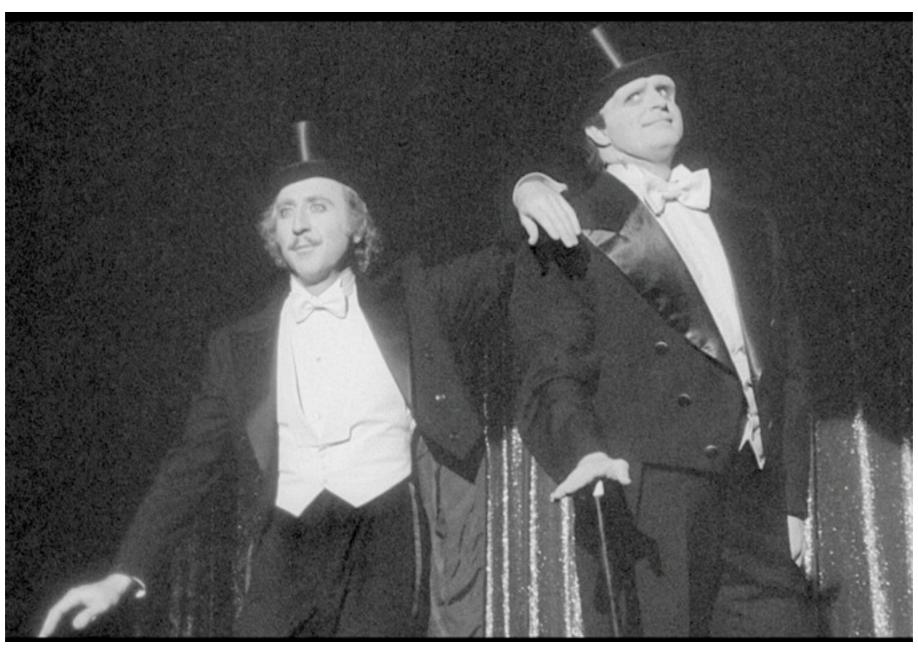# Homework

Enhance this todo app:

https://github.com/mjansen401/todo

-Complete tasks
-Remove tasks
-Add due date
-Clear completed tasks