App Orgnanization

- LEVEL 6 -

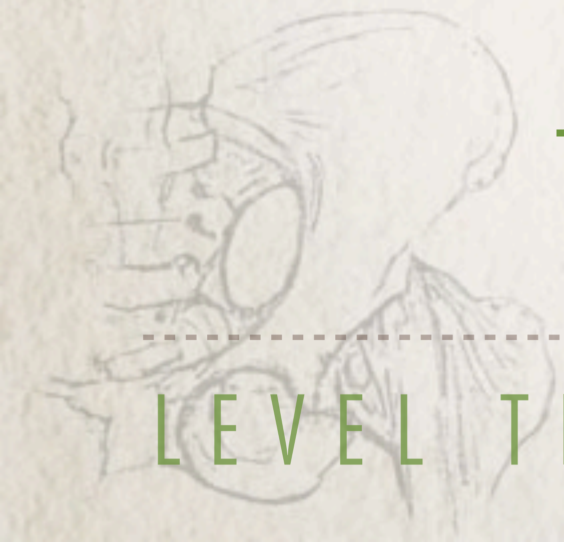
Class naming

Everything in the Global Scope

```
var TodoItem = Backbone.Model.extend({});
var TodoItemView = Backbone.View.extend({});
var TodoItems = Backbone.Collection.extend({});
var TodoItemsView = Backbone.View.extend({});
var TodoRouter = Backbone.Router.extend({});
```

✗

- Leads to naming collisions
- Need to put “what kind of object” it is in the name e.g. “TodoItemView”
- Maintainability doesn’t scale with large applications



Use a Global Object for Namespace

```
var App = {
  Models: {},
  Views: {},
  Collections: {}
}
```

Create a single global object where everything is stored

```
App.Models.TodoItem = Backbone.Model.extend({});
App.Views.TodoItem = Backbone.View.extend({});
App.Collections.TodoItems = Backbone.Collection.extend({});
App.Views.TodoItems = Backbone.View.extend({});
App.TodoRouter = Backbone.Router.extend({});
```

Store one-off objects on App

Reference classes with the namespace

```
var todoItem = new App.Models.TodoItem({...})
```

Handle Links Outside of Backbone Views

```html
<ul>
  <li><a href="/completed">Show Completed</a></li>
  <li><a href="/support">Support</a></li>
</ul>
```

Using plain jQuery

```javascript
$('a').click(function(e){
  e.preventDefault();
  Backbone.history.navigate(e.target.pathname, {trigger: true});
});
```

! What if we wanted backbone to handle this?

Handle Links Outside of Backbone Views

```
var App = {
  Models: {},
  Views: {},
  Collections: {},
  start: function(){
    $('a').click(function(e){
      e.preventDefault();
      Backbone.history.navigate(e.target.pathname, {trigger: true});
    });

    Backbone.history.start({pushState: true});
  }
}
```

```
$(function(){ App.start(); })
```

! **Outside of Backbone Views**

Handle Links Outside of Backbone Views

```
var App = Backbone.View.extend({
  Models: {},
  Views: {},
  Collections: {},
  events: {
    'click a': function(e){
      e.preventDefault();
      Backbone.history.navigate(e.target.pathname, {trigger: true});
    }
  },
  start: function(){
    Backbone.history.start({pushState: true});
  }
})
```

App is now a View Class

Use View event handling for clicks

```
var app = new App({el: document.body});
```

Using body will capture everything

Handle Links Outside of Backbone Views

```
var App = new (Backbone.View.extend({
  ...
  events: {
    'click a': function(e){
      e.preventDefault();
      Backbone.history.navigate(e.target.pathname, {trigger: true});
    }
  },
  start: function(){
    Backbone.history.start({pushState: true});
  }
}))({el: document.body});
```

← Create instance without creating a class

```
$(function(){ App.start(); })
```

Skip links

```
<ul>
  <li><a href="/completed">Show Completed</a></li>
  <li><a href="/support">Support</a></li>
</ul>
```

Event Handler

```
events: {
  'click a': function(e){
    e.preventDefault();
    Backbone.history.navigate(e.target.pathname, {trigger: true});
  }
}
```

Captures all link clicks

Skip links

```
<ul>
  <li><a href="/completed" data-internal="true">Show Completed</a></li>
  <li><a href="/support">Support</a></li>
</ul>
```

Event Handler

```
events: {
  'click a[data-internal]': function(e){
    e.preventDefault();
    Backbone.history.navigate(e.target.pathname, {trigger: true});
  }
}
```

Captures only the links we want it to

Build Initial HTML

```
var App = new (Backbone.View.extend({

  ...

  template: _.template('<h1>ToDo List!</h1>' +
    '<div id="app"></div>'),

  render: function(){
    this.$el.html(this.template());
  }

}))({el: document.body});
```

```
$(function(){
  App.render();
  App.start();
})
```

```
<body>
  <h1>ToDo List!</h1>
  <div id="app"></div>
</body>
```

Add Initial HTML to the body

Object Initialization

```
var App = new (Backbone.View.extend({
  ...
  start: function(){
    var todos = new App.Collections.TodoItems();
    var todosView = new App.Views.TodoItems({collection: todos});
    this.$el.append(todosView.render().el);
    todos.fetch();
  }
}))({el: document.body});
```

Ajax slows down initial render

```
$(function(){ App.start(); })
```

Bootstrap Model Data

```
var App = new (Backbone.View.extend({
  start: function(bootstrap){
    var todos = new App.Collections.TodoItems(bootstrap.todos);
    var todosView = new App.Views.TodoItems({collection: todos});
    this.$el.append(todosView.render().el);
  }
}))({el: document.body});
```

```
var bootstrap = {
  todos: [
    {id: 1, description: "Pickup Milk.", status: "complete"},
    {id: 2, description: "Pickup Kids.", status: "incomplete"},
  ]
}
```

Bootstrap data comes from rendered HTML page

```
$(function(){ App.start(bootstrap); })
```

Pass in data to start

App Orgnanization

- LEVEL 6 -