

More Models

- LEVEL 1 -



ReIntroducing Backbone.js



“Get your truth out of the DOM”

- Jeremy Ashkenas

Provides client-side app structure

Models to represent data

Views to hook up models to the DOM

Synchronizes data to/from server

More Models



Reviewing how Model Data is fetched from Server

```
var TodoItem = Backbone.Model.extend({urlRoot: '/todos'});
```

Fetch todo with id = 1

```
var todoItem = new TodoItem({id: 1})
```

```
todoItem.fetch();
```

GET /todos/1

--> { id: 1, description: 'Pick up milk', status: 'incomplete' }



*RESTful Web Service
(Rails Flavor)*

More Models



Parsing non-standard JSON into your Models

```
todoItem.fetch();
```

A non-standard Response from Server

```
{todo:{id: 1, description: 'Pick up milk', status: 'incomplete'}}
```

```
var TodoItem = Backbone.Model.extend({  
  parse: function(response){  
    return response;  
  }  
});
```

**Default Backbone
just returns response**

```
todoItem.attributes; --> { }
```



More Models




Server Breaking Conventions

```
todoItem.fetch();
```

Response from Server

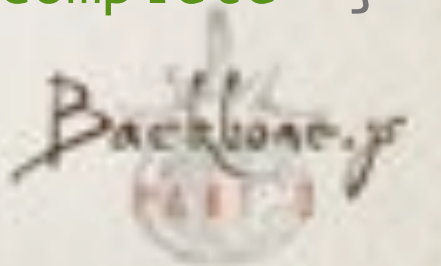
```
{todo:{id: 1, description: 'Pick up milk', status: 'incomplete'}}
```

```
var TodoItem = Backbone.Model.extend({  
  parse: function(response){  
    return response.todo;  
  }  
});
```



```
todoItem.attributes; --> { id: 1, description: 'Pick up milk', status: 'incomplete' }
```

More Models



Instantiating Models doesn't call parse by Default

```
var todoItem = new TodoItem({  
  todo: {id: 1, description: 'Pick up milk', status: 'incomplete'}  
});
```



```
todoItem.attributes; --> { }
```



```
var todoItem = new TodoItem({  
  todo:{id: 1, description: 'Pick up milk', status: 'incomplete'}  
}, { parse: true });
```



```
todoItem.attributes;
```

```
--> { id: 1, description: 'Pick up milk', status: 'incomplete' }
```

More Models



Changing Attribute Names

Response from Server

```
{todo:{id: 1, desc: 'Pick up milk', status: 'incomplete'}}
```

```
var TodoItem = Backbone.Model.extend({  
  parse: function(response){  
    response = response.todo;  
    response.description = response.desc;  
    delete response.desc;  
    return response;  
  }  
});
```



```
todoItem.attributes;
```

```
--> { id: 1, description: 'Pick up milk', status: 'incomplete' }
```

More Models



Sending JSON back to the Server

Update the todo

```
todoItem.set({description: 'Pick up cookies.'});
```

```
todoItem.save();
```

PUT /todos/1

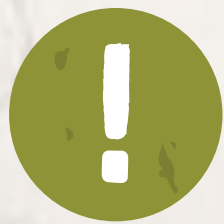
Sends to JSON() to server

```
{id: 1, description: 'Pick up cookies.', status: 'incomplete'}
```



Server expects JSON with root and desc

```
{todo:{id: 1, desc: 'Pick up cookies.', status: 'incomplete'}}
```



**So how do we modify the JSON
before we send it to the server?**

More Models



Overriding the toJSON function

```
var TodoItem = Backbone.Model.extend({  
  toJSON: function(){  
    return _.clone(this.attributes);  
  }  
});
```

default backbone



```
{  
  id: 1,  
  description: 'Pick up cookies.',  
  status: 'incomplete'  
}
```

```
var TodoItem = Backbone.Model.extend({  
  toJSON: function(){  
    return { todo: _.clone(this.attributes) };  
  }  
});
```



```
{  
  todo: { id: 1, description: 'Pick up cookies.', status: 'incomplete' }  
}
```

More Models



Overriding the toJSON function

```
var TodoItem = Backbone.Model.extend({  
  toJSON: function(){  
    var attrs = _.clone(this.attributes);  
    attrs.desc = attrs.description;  
    attrs = _.pick(attrs, 'desc', 'status');  
    return { todo: attrs };  
  }  
});
```

*returns an object
with only the 'desc'
and 'status' properties*



```
todoItem.toJSON();
```

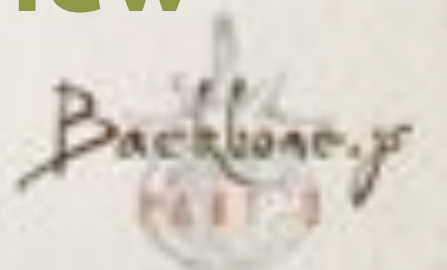


```
{  
  todo: { desc: 'Pick up cookies.', status: 'incomplete' }  
}
```



**Now we
can't use
toJSON in
View**

More Models



Render View with Attributes

```
var TodoView = Backbone.View.extend({  
  render: function(){  
    this.$el.html(this.template(this.model.toJSON()));  
    return this;  
  }  
});
```

*toJSON no
longer works
for templates*



Use model.attributes instead

```
var TodoView = Backbone.View.extend({  
  render: function(){  
    this.$el.html(this.template(this.model.attributes));  
    return this;  
  }  
});
```



More Models



Unconventional ID Attribute

```
var todoItem = new TodoItem({id: 1})
```

```
todoItem.fetch();
```

--> { _id: 1, description: 'Pick up milk', status: 'incomplete' }

todoItem.id --> undefined



```
var TodoItem = Backbone.Model.extend({  
  idAttribute: '_id'  
});
```

todoItem.id --> 1



More Models

