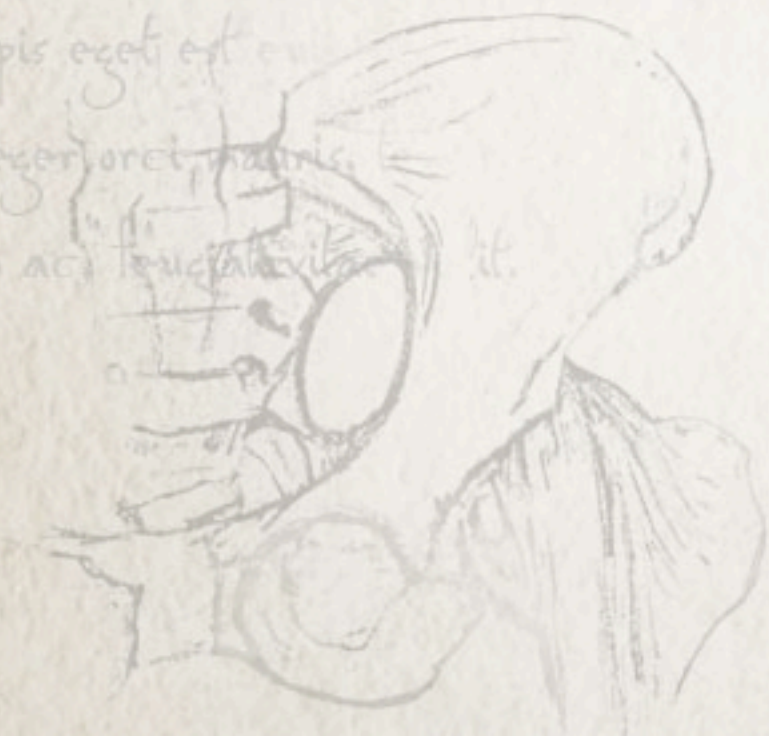


amet dolor. fusce at facilisis
velit. Maecenas lacinia turpis eget



non commodo enim.
tum lacus iaculis digni
usce at facilisis velit
turpis eget est
Integer orci, mauris
tum ac feugiat vivit.



Customizing Collections

- LEVEL 2 -

est enim euismod nec congue odio porta. Integer orci
quis condictum ac feugiat vivit.



ec a turpis in
llam a lacus
cursus digni



Integer mollis risus elementum
amet dolor. fusce at facilisis velit. Maecenas
euismod nec congue odio porta. Integer orci

Review fetching Data from the Server

```
var TodoItems = Backbone.Collection.extend({  
  url: '/todos'  
});
```

*URL to get
JSON data from*

populate collection from server

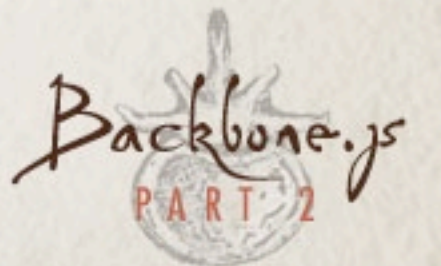
`todoItems.fetch()` **GET /todos**



```
[  
  {description: 'Pick up milk.', status: 'incomplete', id: 1},  
  {description: 'Get a car wash.', status: 'incomplete', id: 2}  
]
```

`todoItems.length` **---> 2**

Customizing Collections



Handling non-standard Response from Server

```
todoItems.fetch()
```

Response from Server

```
{  
  "total": 25, "per_page": 10, "page": 2,  
  "todos": [ {"id": 1, ... }, {"id": 2, ... } ]  
}
```

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

```
todoItems.toJSON(); --> { }
```

Customizing Collections

Handling non-standard Response from Server

Response from Server

```
{  
  "total": 25, "per_page": 10, "page": 2,  
  "todos": [ {"id": 1}, {"id": 2} ]  
}
```

```
var TodoItems = Backbone.Collection.extend({  
  parse: function(response){  
    this.perPage = response.per_page;  
    this.page = response.page;  
    this.total = response.total;  
    return response.todos;  
  }  
});
```



todoItems.perPage --> 10

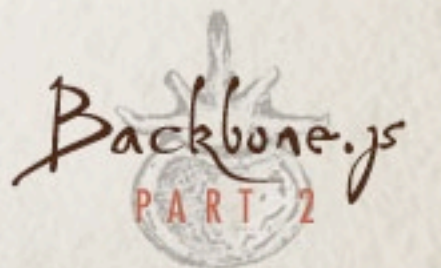
todoItems.page --> 2

todoItems.total --> 25

**These aren't
special properties**

todoItems.toJSON(); --> [{ id: 1, description: 'Pick up milk.' ... }]


Customizing Collections



How do we fetch with extra params?

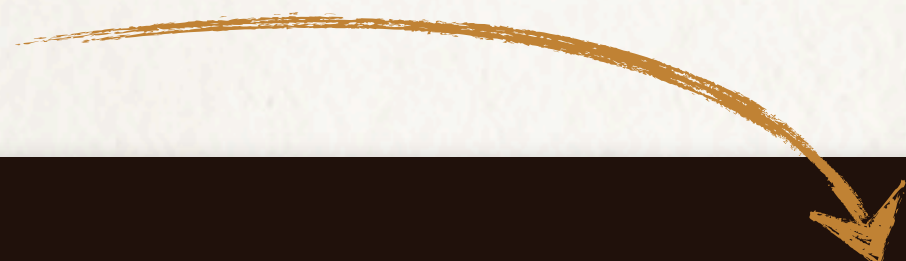

`todoItems.fetch()`

GET /todos



```
{  
  "total": 100, "per_page": 10, "page": 1,  
  "todos": [ {"id": 1}, {"id": 2} ]  
}
```

GET /todos?page=6




```
{  
  "total": 100, "per_page": 10, "page": 6,  
  "todos": [ {"id": 61}, {"id": 62} ]  
}
```



fetching with extra params

```
todoItems.fetch({data: { page: 6 }})
```

GET /todos?page=6



```
{  
  "total": 100, "per_page": 10, "page": 6,  
  "todos": [ {"id": 61}, {"id": 62} ]  
}
```

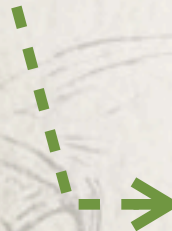


Get next page

```
todoItems.page ---> 1
```

```
todoItems.fetch({data: {page: todoItems.page + 1}});
```

GET /todos?page=2



```
{  
  "total": 100, "per_page": 10, "page": 2,  
  "todos": [ {"id": 11}, {"id": 21} ]  
}
```


Review our Collection View

```
var TodosView = Backbone.View.extend({
  initialize: function(){
    this.collection.on('reset', this.render, this);
  },
  render: function(){
    this.addAll();
    return this;
  },
  addAll: function(){
    this.$el.empty();
    this.collection.forEach(this.addOne);
  },
  addOne: function(todoItem){
    var todoView = new TodoView({model: todoItem});
    this.$el.append(todoView.render().el);
  }
});
```

How do we add a
“next” link?

Render the Next Link

```
var TodosView = Backbone.View.extend({
  template: _.template('<a href="#/todos/p<%= page %>">next page</a>'),
  initialize: function(){
    this.collection.on('reset', this.render, this);
  },
  render: function(){
    this.addAll();
    this.$el.append(this.template({page: this.collection.page + 1}));
    return this;
  },
  ...
});
```

Now all we have to create a route for “/todos/p:page”

Customizing Collections



Review our Router

```
var TodoApp = new (Backbone.Router.extend({
  routes: {
    "": "index"
  },
  initialize: function(){
    this.todoItems = new TodoItems();
    this.todosView = new TodosView({collection: this.todoItems});
    this.todosView.render();
    $('#app').append(this.todosView.el);
  },
  index: function(){
    this.todoItems.fetch();
  }
}));
```



Implementing the Page Route

```
var TodoApp = new (Backbone.Router.extend({
  routes: {
    "todos/p:page": "page",
    "": "index"
  },
  page: function(page){
    this.todoItems.fetch({data: {page: page}});
  },
  initialize: function(){
    this.todoItems = new TodoItems();
    this.todosView = new TodosView({collection: this.todoItems});
    this.todosView.render();
    $('#app').append(this.todosView.el);
  },
  index: function(){
    this.todoItems.fetch();
  }
}));
```



Sorting Collections by Comparator

Sort by Status

```
var TodoItems = Backbone.Collection.extend({  
  comparator: 'status'  
});
```

```
todoItems.fetch();
```



```
[  
  {description: 'Get a car wash.', status: 'complete', id: 2},  
  {description: 'Pick up milk.', status: 'incomplete', id: 1}  
]
```


Sorting Collections by function

Passing a function in as a comparator

```
var TodoItems = Backbone.Collection.extend({  
  comparator: function(todo1, todo2) {  
    return todo1.get('status') < todo2.get('status')  
  }  
});
```

**Sort by status
in reverse order**

todoItems.fetch()



```
[  
  {description: 'Pick up milk.', status: 'incomplete', id: 1},  
  {description: 'Get a car wash', status: 'complete', id: 2}  
]
```

Customizing Collections



Aggregate Values

Display amount of completed todo items

```
var TodoItems = Backbone.Collection.extend({  
  completedCount: function() {  
    return this.where({status: 'complete'}).length;  
  }  
});
```

**Returns filtered
array of models**

```
[  
  {description: 'Pick up milk.', status: 'incomplete', id: 1},  
  {description: 'Get a car wash', status: 'complete', id: 2}  
]
```

todoItems.completedCount() -----> 1

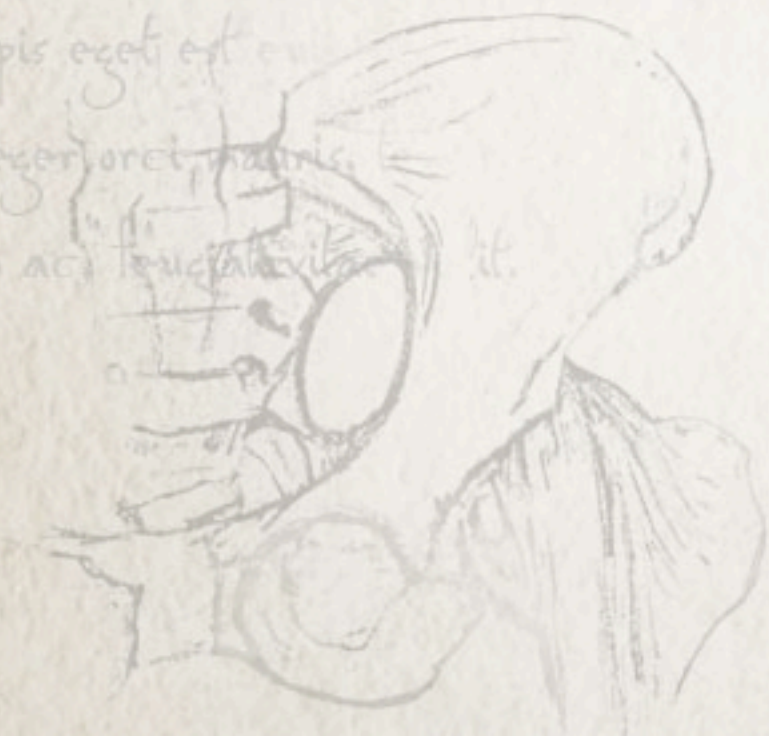
Customizing Collections



amet dolor. fusce at facilisis
velit. Maecenas lacinia turpis eget



non commodo enim.
tum lacus iaculis dis
sce at facilisis velit
turpis eget est eu
Integer orci, mauris.
tum ac feugiat vivit.



Customizing Collections

- LEVEL 2 -

est enim euismod nec congue odio porta. Integer orci
quis condictum ac feugiat



ec a turpis in
llam a lacus
cursus



Integer mollis risus elementum
amet dolor. fusce at facilisis velit. Maecenas
euismod nec congue odio porta. Integer orci