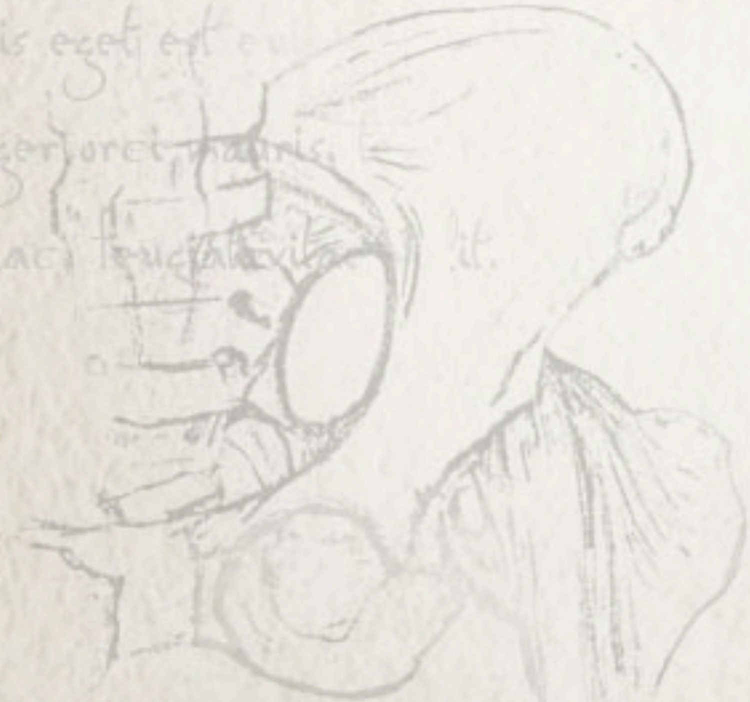
Working with Forms

- LEVEL 5 -



Create an Ajax form for creating new Todos

```
<form action="/todos" method="POST">
  <input name=description value="What do you need to do?" />
  <button>Save</button>
</form>
```

Default description

Use plain jQuery

```
$('button').click(function(e){
  e.preventDefault();
  var uri = $('form').attr('action');
  var description = $('input[name=description]').val();
  $.post(uri, {description: description});
});
```

Wait, shouldn’t we be creating Todos using Backbone?

--> **POST /todos** `{ description: 'Pickup Kids.' }`

Convert to using Backbone

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

Default description

```
var todoItem = new TodoItem({description: "What do you need to do?"});

todoItem.save({description: 'Pickup Kids.'});
```

POST /todos `{ description: 'Pickup Kids.' }`

What about the form?

Build Form with Backbone View

```
var TodoForm = Backbone.View.extend({
  template: _.template('<form>' +
    '<input name=description value="<%= description %>" />' +
    '<button>Save</button></form>'),
  render: function(){
    this.$el.html(this.template(this.model.attributes));
    return this;
  }
});
```

```
var todoItem = new TodoItem({description: "What do you need to do?"});
var todoForm = new TodoForm({model: todoItem});
$('#app').html(todoForm.render().el);
```

```
<form><input name=description value="What do you need to do?" />
<button>Save</button></form>
```

Capture Button Click to Save Model

```
var TodoForm = Backbone.View.extend({
  template: _.template('<form>' +
    '<input name=description value="<%= description %>" />' +
    '<button>Save</button></form>'),
  events: {
    'click button': 'save'
  },
  save: function(e) {
    e.preventDefault();
    var newDescription = this.$('input[name=description]').val();
    this.model.save({description: newDescription});
  }
});
```

POST /todos `{ description: 'Pickup Kids.' }`

How do we save when pressing enter?

Capture Return Key to Save Model

```
var TodoForm = Backbone.View.extend({
  template: _.template('<form>' +
    '<input name=description value="<%= description %>" />' +
    '<button>Save</button></form>'),
  events: {
    submit: 'save'
  },
  save: function(e) {
    e.preventDefault();
    var newDescription = this.$('input[name=description]').val();
    this.model.save({description: newDescription});
  }
});
```

Will call save on either click or pressing Enter/Return

Reusing Form to Edit existing TodoItem

Get existing TodoItem from already fetched collection

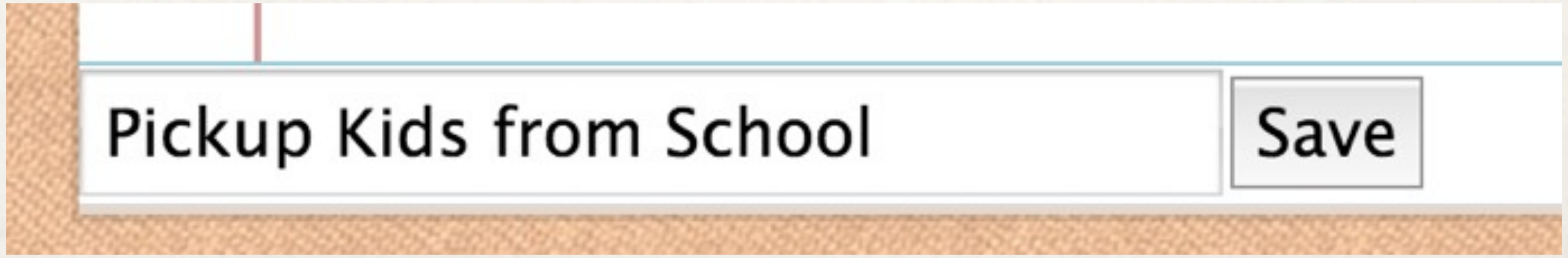
```
var todoItem = todoItems.get(1);
```

Pass in existing model

```
var editTodoForm = new TodoForm({model: todoItem});
```

Replace #app with the HTML of the form

```
$('#app').html(editTodoForm.render().el);
```



Pickup Kids from School

Review of our App's Router

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

Let's add a route to render the TodoForm with an existing TodoItem

Add Route to render Edit Form

```
var TodoApp = new (Backbone.Router.extend({
  routes: {
    "": "index",
    "todos/:id/edit": "edit"
  },
  initialize: function(){
    this.todoItems = new TodoItems();
    this.todosView = new TodosView({collection: this.todoItems});
  },
  index: function(){
    this.todoItems.fetch();
    $('#app').html(this.todosView.render().el);
  }
  edit: function(id){
    var todoForm = new TodoForm({model: this.todoItems.get(id) });
    $('#app').html(todoForm.render().el);
  }
}));
```

Add Route to render New Form

```
var TodoApp = new (Backbone.Router.extend({
  routes: {
    "": "index",
    "todos/:id/edit": "edit",
    "todos/new": "newTodo"
  },
  ...

  newTodo: function(){
    var todoItem = new TodoItem({description: "What do you have to do?"});
    var todoForm = new TodoForm({model: todoItem});
    $('#app').append(todoForm.render().el);
  },

  edit: function(id){
    var todoForm = new TodoForm({model: this.todoItems.get(id) });
    $('#app').html(todoForm.render().el);
  }
}));
```

Get Back to the List after saving

```
var TodoForm = Backbone.View.extend({
  ...
  save: function(e) {
    e.preventDefault();
    var newDescription = this.$('input[name=description]').val();
    this.model.save({description: newDescription});
    Backbone.history.navigate('', { trigger: true });
  }
});
```

We should only call this if the model saved successfully

Get Back to the List after saving success

```
var TodoForm = Backbone.View.extend({
  ...
  save: function(e) {
    e.preventDefault();
    var newDescription = this.$('input[name=description]').val();
    this.model.save({description: newDescription}, {
      success: function(model, response, options){
        Backbone.history.navigate('', { trigger: true });
      }
    });
  }
});
```

✓

What happens if the Todo doesn’t save?

Alert User to save error

```
var TodoForm = Backbone.View.extend({
  ...
  save: function(e) {
    e.preventDefault();
    var newDescription = this.$('input[name=description]').val();
    this.model.save({description: newDescription}, {
      success: function(model, response, options){
        Backbone.history.navigate('', { trigger: true });
      }, error: function(model, xhr, options){
        var errors = JSON.parse(xhr.responseText).errors;
        alert('Oops, something went wrong with saving the TodoItem: ' + errors);
      }
     );
    }
  }
});
```

Working with Forms

- LEVEL 5 -