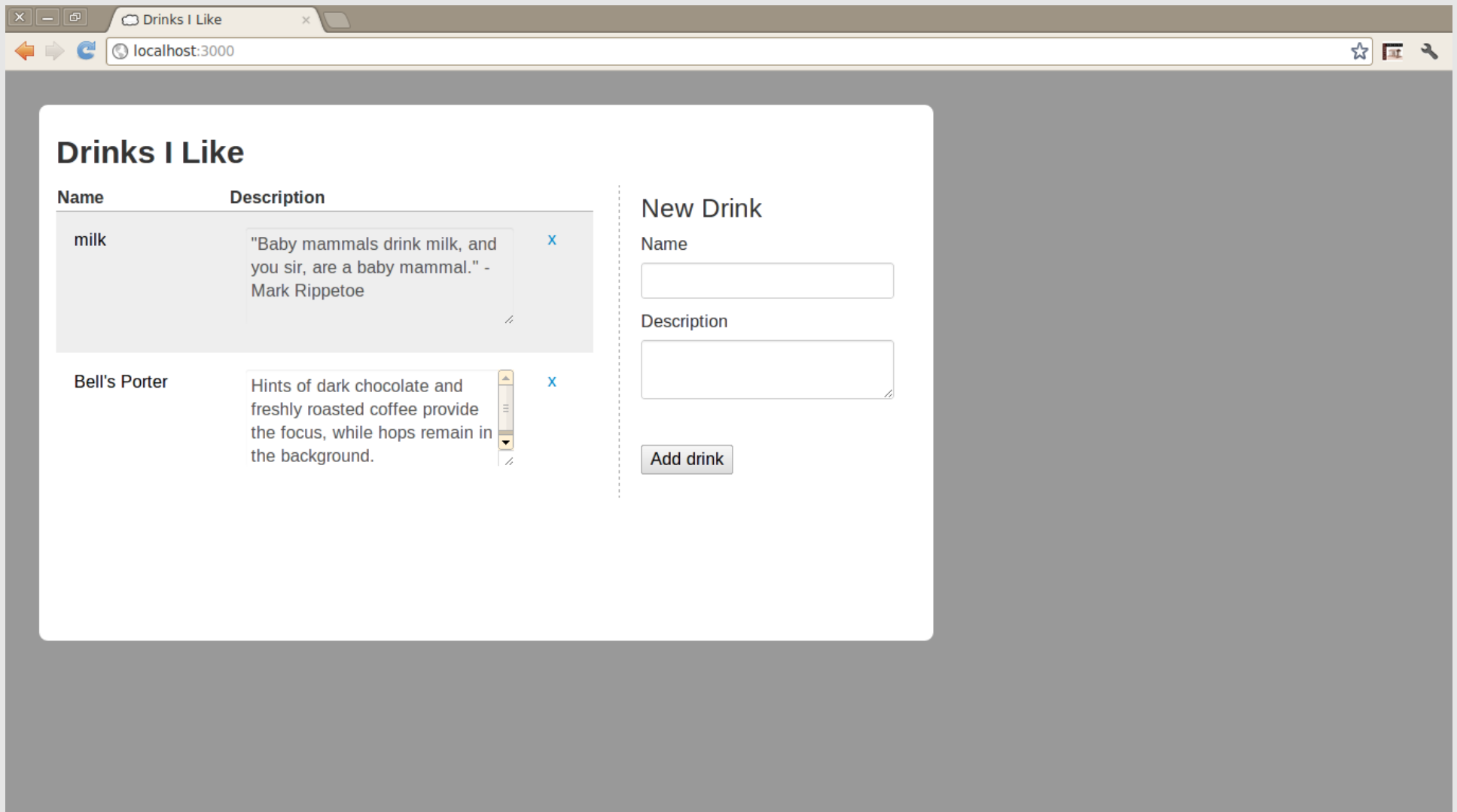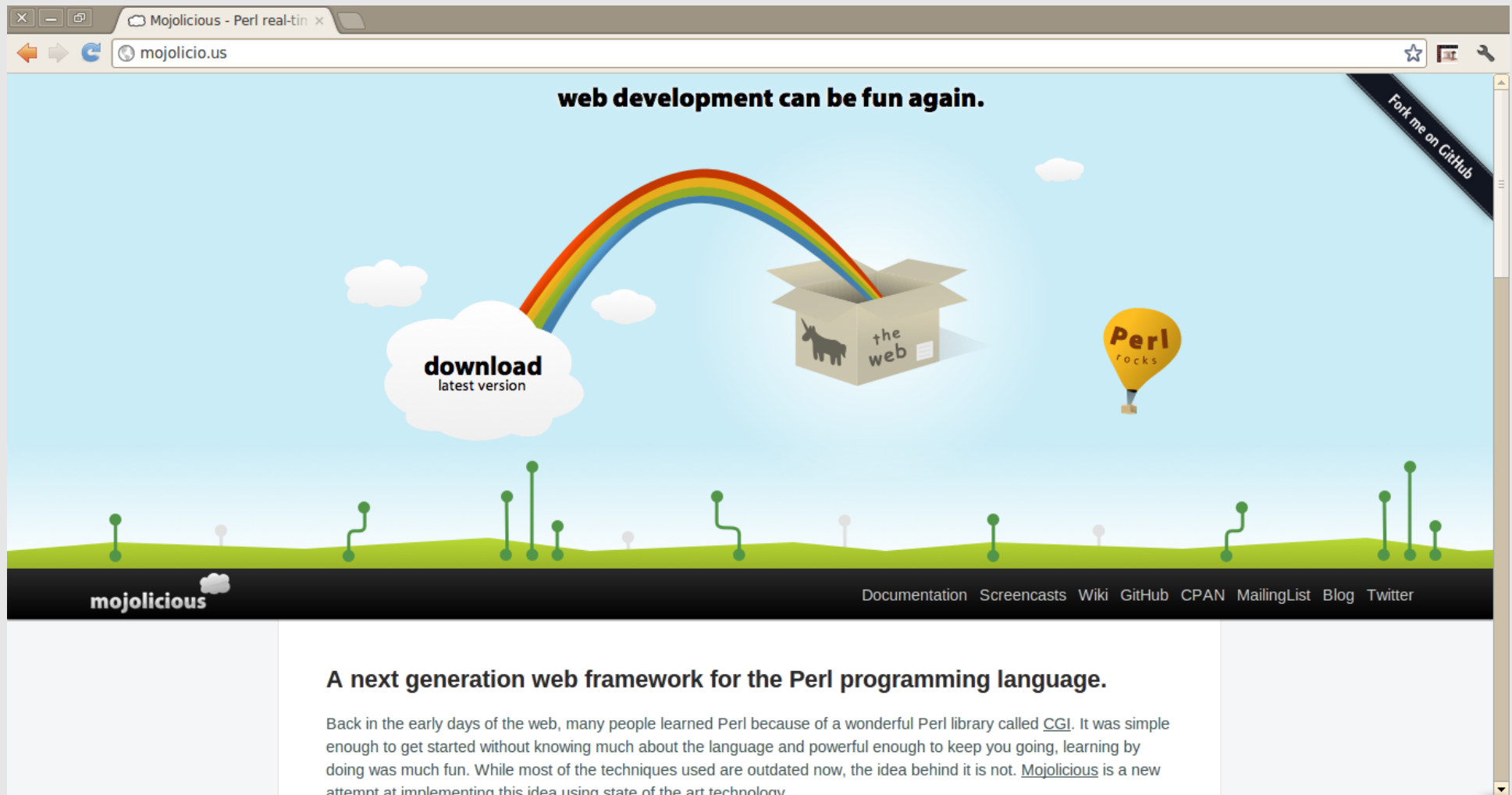# Single page webapps with
Mojolicious and Backbone.js
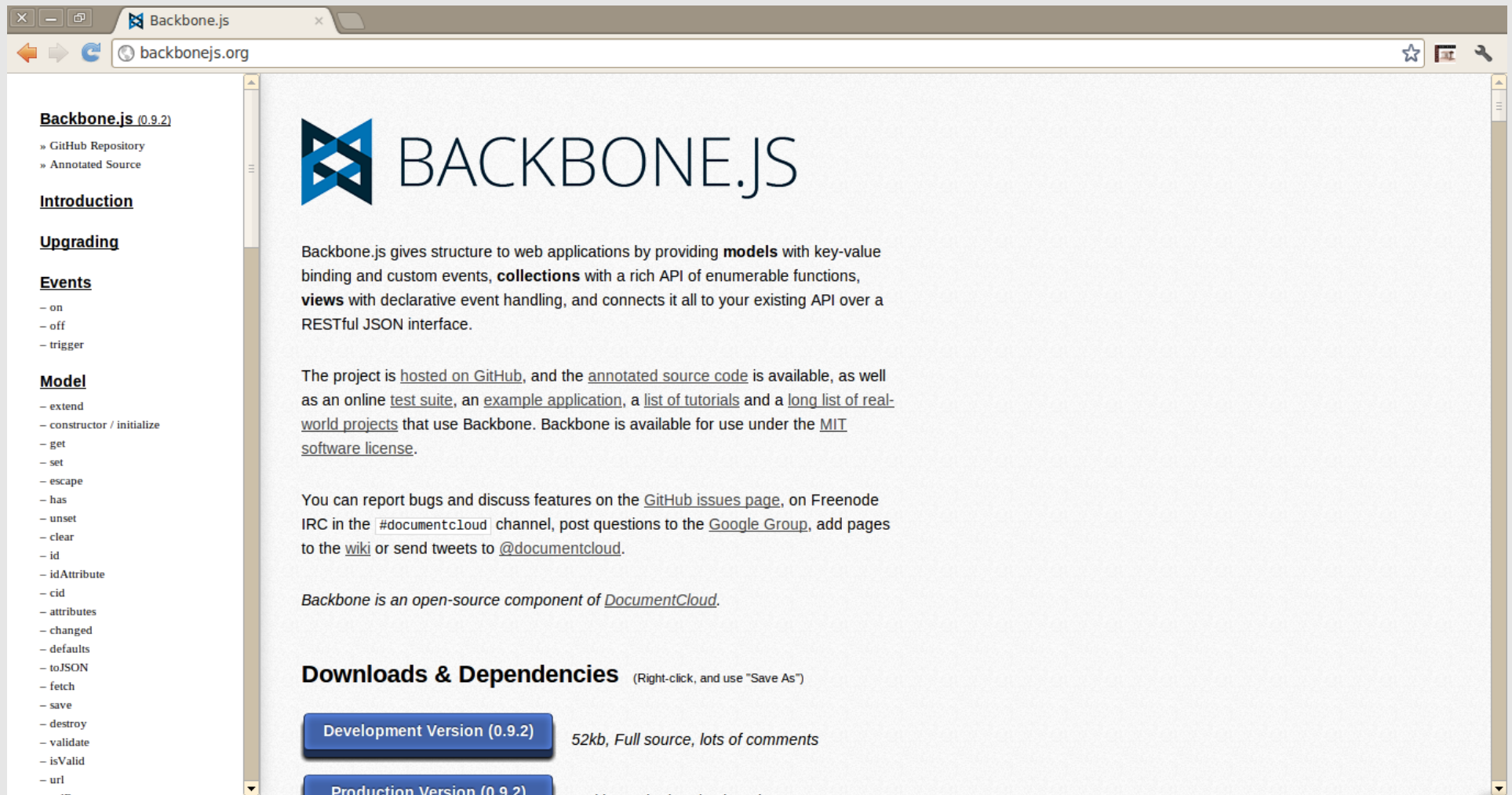
Here's the app...

Mojolicious, Backbone.js & Bootstrap

Mojolicious, Backbone.js & Bootstrap

Mojolicious, Backbone.js & Bootstrap

REST

- Data formats (e.g., XML, JSON, YAML, etc.)

- HTTP method ("verbs") + resource

  - POST   -> CRUD
  - GET *   -> CRUD
  - PUT *   -> CRUD
  - DELETE *   -> CRUD

  * Should be idempotent

RESTful design

Method	Route
POST	`/api/drink` (JSON object in body)
GET	`/api/drink/` (Return all drinks) `/api/drink/:id` (Return one drink)
PUT	`/api/drink/:id` (JSON obj in body)
DELETE	`/api/drink/:id`

Let's install from GitHub
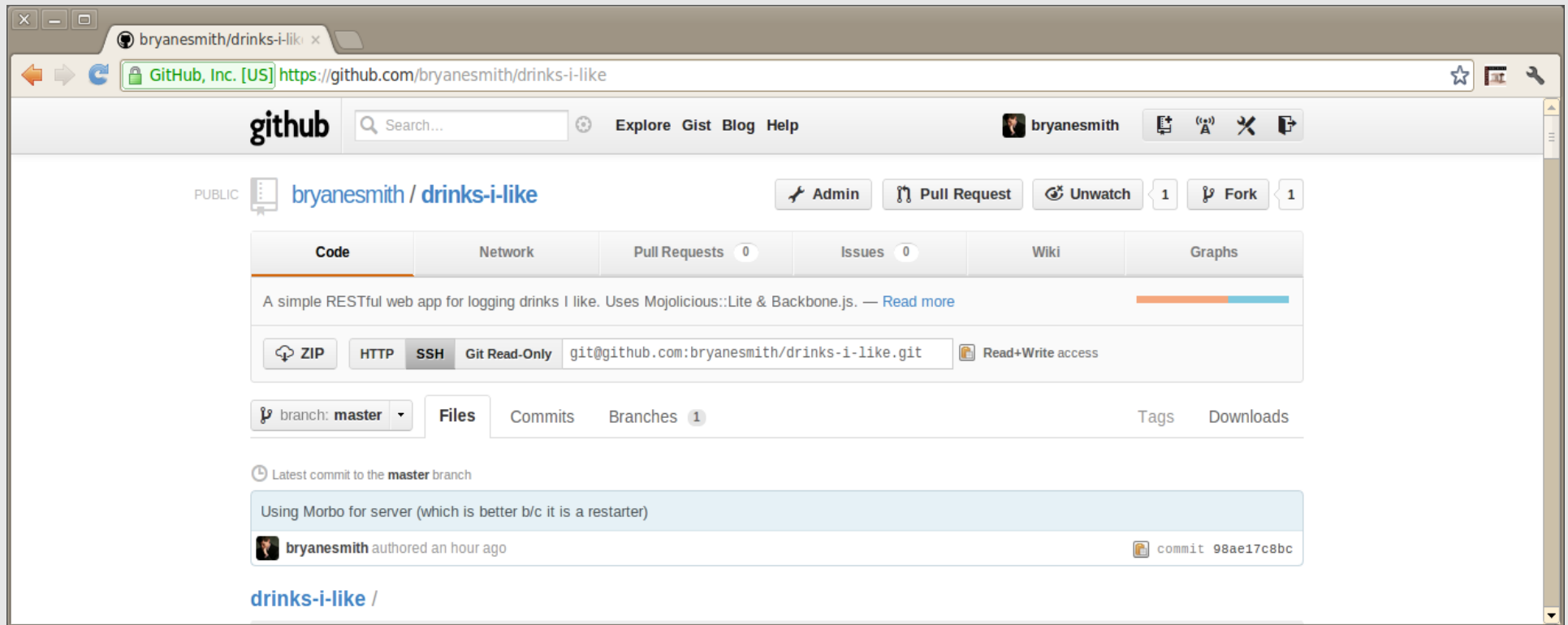
`git clone git@github.com:bryanesmith/drinks-i-like.git`

Mojolicious

```perl
38  # - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
39  # Routes
40  # - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
41
42  # Home page
43  get '/'                => \&handle_home;
44
45  # Add a drink
46  post '/api/drink'      => \&handle_post_drink;
47
48  # Get all drinks
49  get '/api/drink'       => \&handle_get_drinks;
50
51  # Get drink
52  get '/api/drink/:id'   => \&handle_get_drink;
53
54  # Update a drink
55  put '/api/drink/:id'   => \&handle_put_drink;
56
57  # Delete a drink
58  del '/api/drink/:id'   => \&handle_delete_drink;
59  del '/api/drink'       => \&handle_delete_drink;
60
61  app->start;
```

Backbone.js: Model, Collection, View & misc

public/js/library.js:

```
$(document).ready( function() {
  ...
});
```

Backbone.js: Model, Collection, View & misc

```
var Drink = Backbone.Model.extend({
  urlRoot: '/api/drink'
});
```

Backbone.js: Model, **Collection**, View & misc

```
var DrinkCollection = Backbone.Collection.extend({

  model:Drink,

  url: '/api/drink'

});

var drinks = new DrinkCollection();
```

Backbone.js: Model, Collection, View & misc

```javascript
var DrinksView = Backbone.View.extend({
  el: $('#drinks'),
  templateId: '#drinks-template',
  events: {
    'click .remove-drink' : 'removeDrink',
    'change .title' : 'updateTitle',
    'change .description' : 'updateDescription'
  },
  ...
});
```

Backbone.js: Model, Collection, View & misc

```javascript
var DrinksView = Backbone.View.extend({
  ...
  initialize: function() {
    // Bind events to collection
    this.collection.bind('add', this.render, this);
    this.collection.bind('remove', this.render, this);
    // Grab template
    this.template =  $.template(this.templateId);
    // Render
    this.render();
  },
  ...
});
```

Backbone.js: Model, Collection, View & misc

```javascript
var DrinksView = Backbone.View.extend({
  ...
  render: function() {
    var that = this;


    $(this.el).find('td').remove();


    // Render each drink
    _(this.collection.models).each(function(drink) {
      $.tmpl(that.template, drink.toJSON()).appendTo(that.el);
    });
  },
  ...
});
```

Backbone.js: Model, Collection, View & misc

- ## Add drink?

  ```
  var drink = new Drink({title: title, description: desc });

  drink.save({wait: true});
  ```

- ## Update drink?

  ```
  var drink.set(newValues);

  drink.save({wait: true});
  ```

- ## Delete drink?

  ```
  drink.destroy({success: function(model, res) { coll.remove( drink ); });
  ```

- ## Fetch drinks?

  ```
  drink.fetch();
  ```

Backbone.js: Model, Collection, View & misc

```javascript
// Grab drinks from server
drinks.fetch({

  success: function() {

    var drinksView = new DrinksView( { collection: drinks } );

  }

});
```

Backbone.js: Model, Collection, View & misc

```javascript
// Handle add drink button
$('#add-drink').click(function () {
  // Create drink from form values
  var drink = new Drink({
    title: $('#add-title').val(),
    description: $('#add-description').val()
  });
  // Reset form
  $('#new-drink').each (function(){ this.reset(); });
  drink.save({wait: true});
  drink.fetch({
    success: function() { drinks.add(drink); }
  });
});
```

Awesome

- Asynchronous HTTP **>** page reload
- RESTful design keeps API simple
- Backbone.js makes it ~~easy~~ easier
- Bootstrap makes it pretty
- Mojolicious makes it fun

Needs improvement

- Delete should be idempotent

- Standard urls & Backbone.Router (History API)

- OO design improvements (e.g., fetching & helper functions should be part of view obj)

- More idiomatic Backbone.js

Resources

- Mojolicious
- Backbone.js
- Bootstrap
- drinks-i-link on GitHub
- Designing a RESTful Web Application

About me

- bryanesmith.com