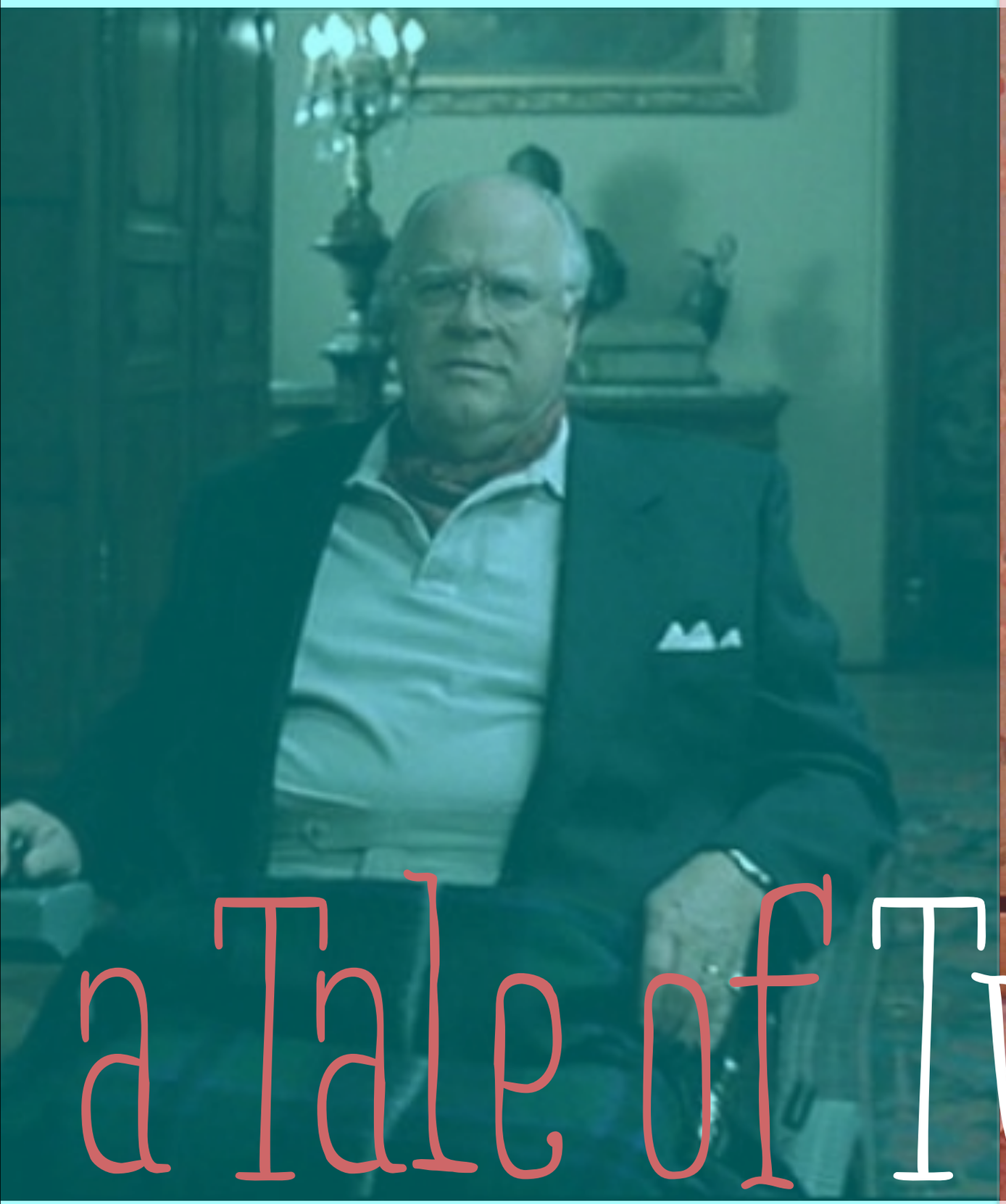
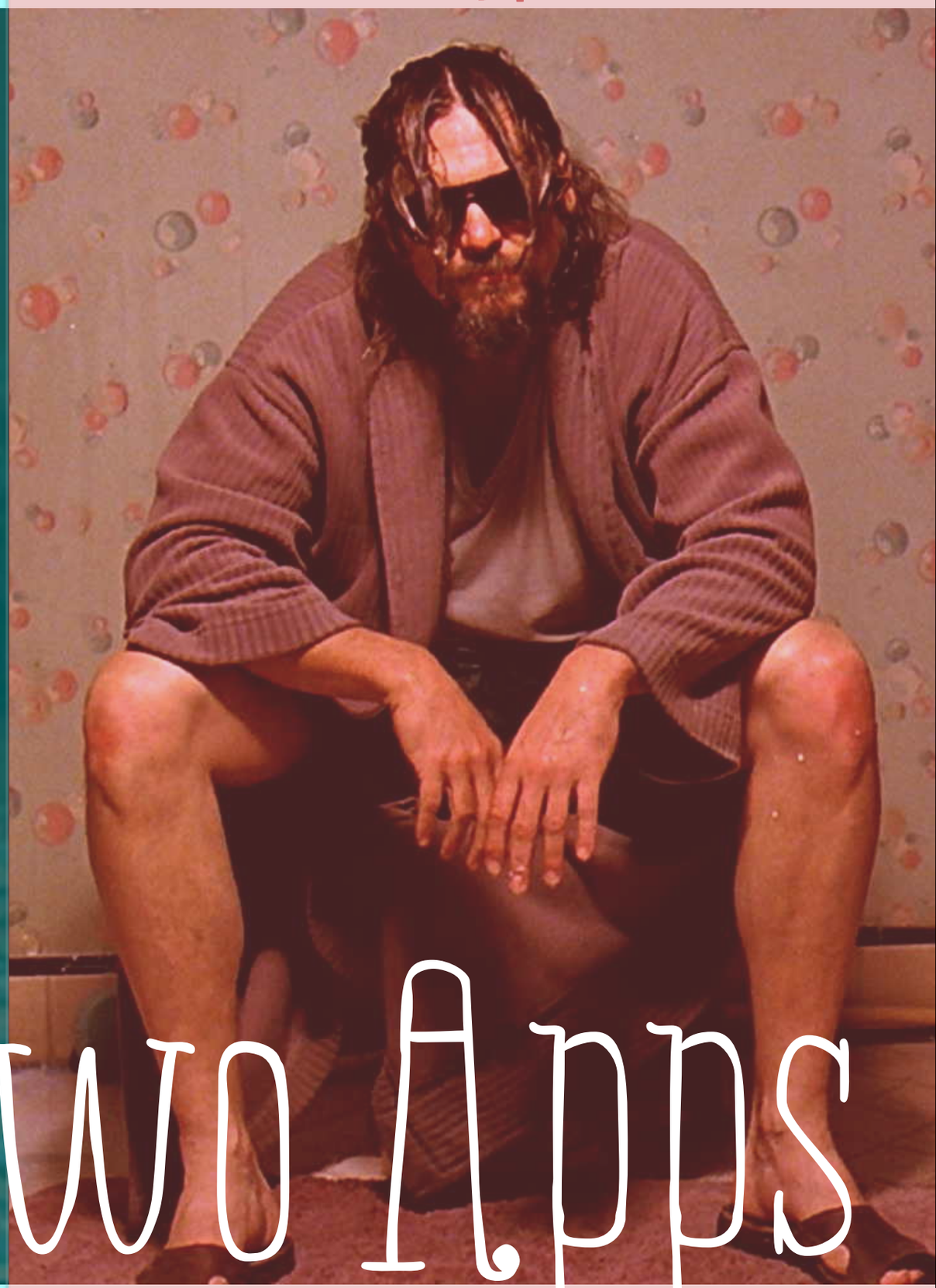
How Ember Changed the Way I Build Web Applications





a Tale of Two Apps

More than just a talk on Ember

Today, we are going to

TALK

about 3 things

A rich

Man

ay, we are going to TALK about 3 things

A rich

Man

A laid back

Dude

e are

g to

K

ings

A rich

A laid back

Man

Dude

A rug

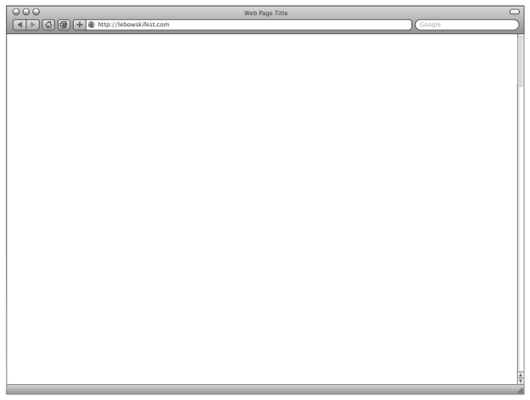
Jamie Wright

@jwright

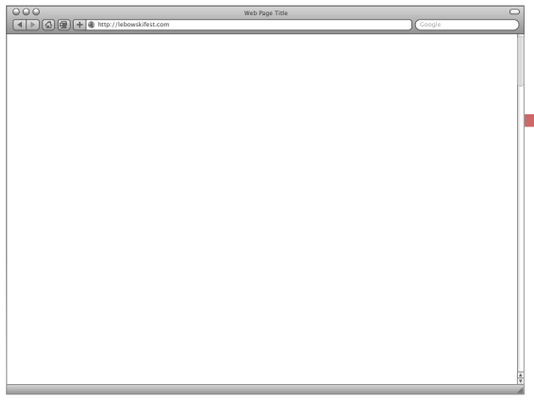
App #1

Server Side Rendering

App #1

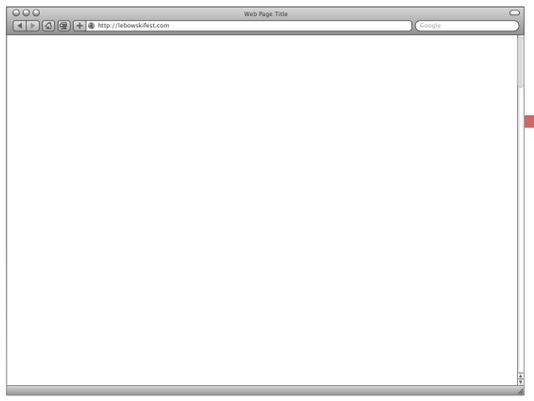


App #1



http://lebowskifest.com

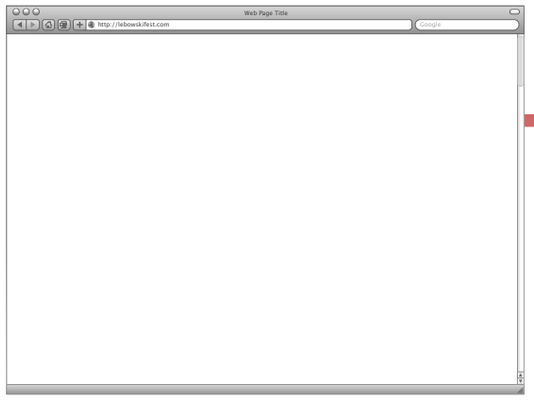
App #1



http://lebowskifest.com

1. Router handles the request

App #1

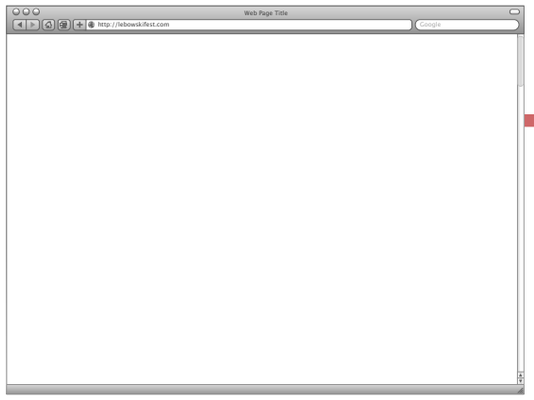


http://lebowskifest.com

1. Router handles the request

2. Controller is created

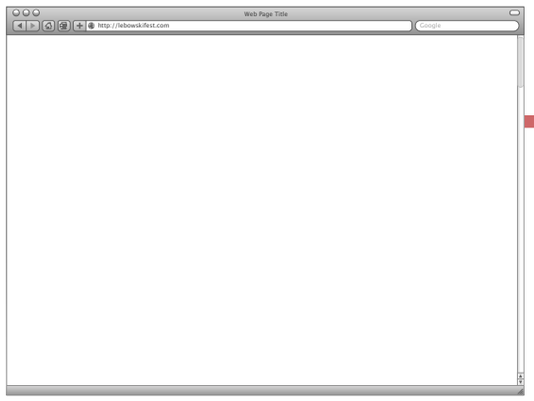
App #1



http://lebowskifest.com

1. Router handles the request
2. Controller is created
3. Controller action is called

App #1



http://lebowskifest.com

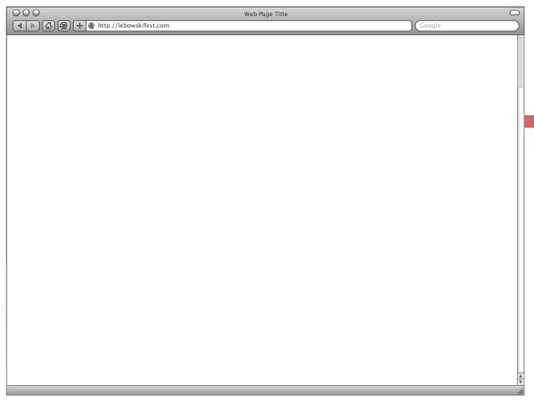
1. Router handles the request

2. Controller is created

3. Controller action is called

4. Controller creates a template & provides data

App #1



http://lebowskifest.com

1. Router handles the request

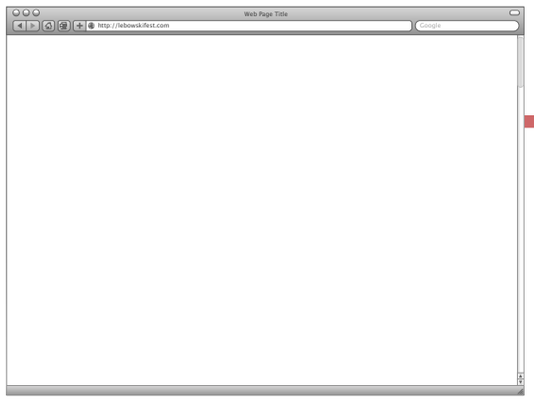
2. Controller is created

3. Controller action is called

4. Controller creates a template & provides data

5. Template is rendered into HTML

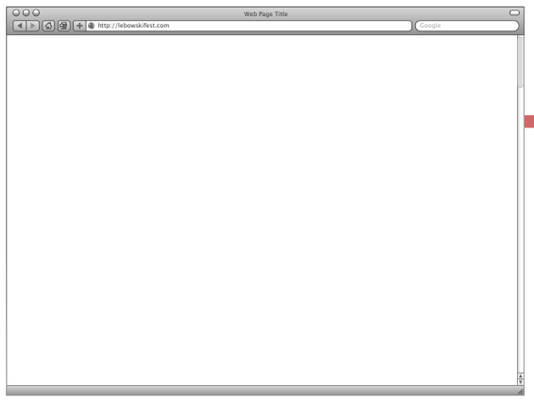
App #1



http://lebowskifest.com

1. Router handles the request
2. Controller is created
3. Controller action is called
4. Controller creates a template & provides data
5. Template is rendered into HTML
6. HTML is returned for the response

App #1



http://lebowskifest.com

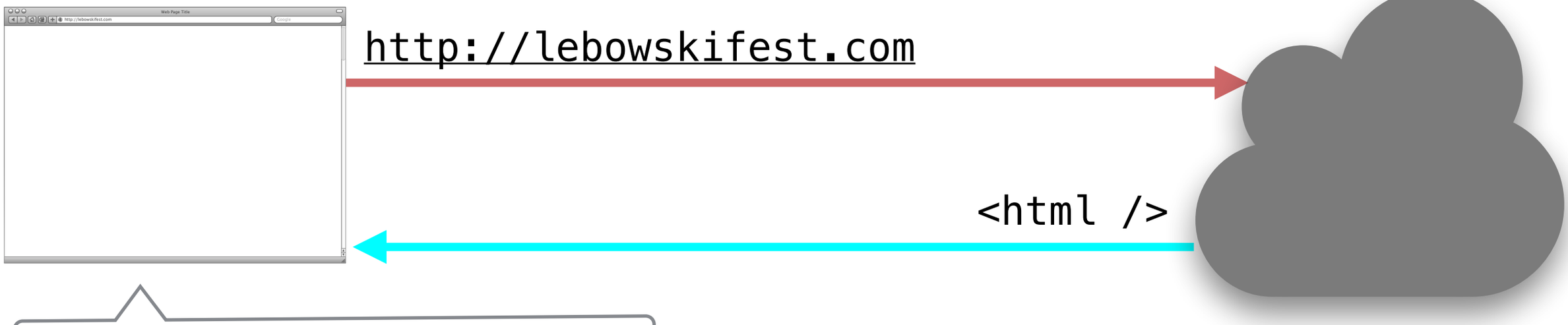
MVC

1. Router handles the request
2. Controller is created
3. Controller action is called
4. Controller creates a template & provides data
5. Template is rendered into HTML
6. HTML is returned for the response

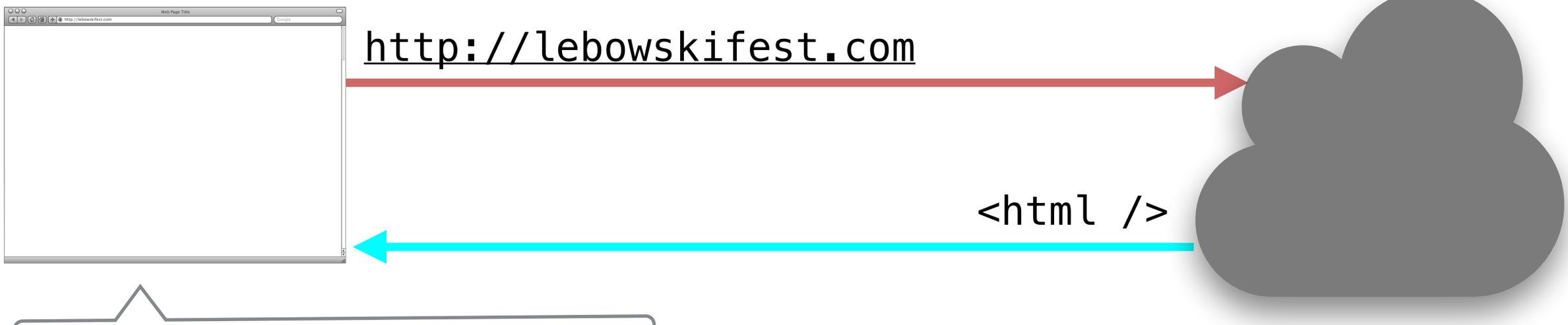
App #1


Web Page Title
http://lebowskifest.com
http://lebowskifest.com
<html />

App #1


Web Page Title
http://lebowskifest.com
http://lebowskifest.com
<html />
1.Browser starts page layout

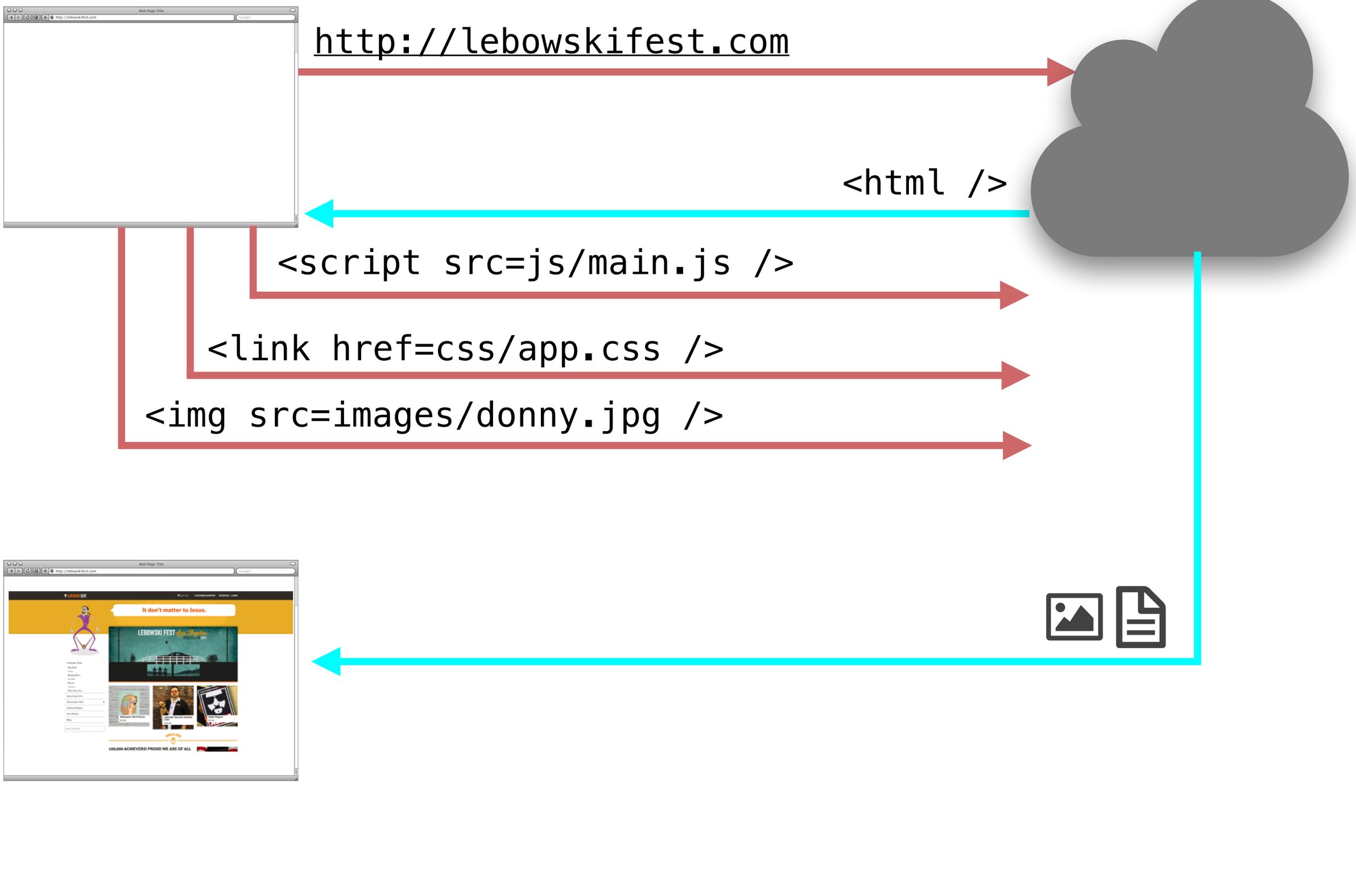
App #1


Web Page Title
http://lebowskifest.com
http://lebowskifest.com
<html />
1. Browser starts page layout
2. Additional sources requested

App #1


http://lebowskifest.com
<html />
<script src=js/main.js />
<link href=css/app.css />

App #1


http://lebowskifest.com
<html />
<script src=js/main.js />
<link href=css/app.css />

It don't matter to Jesus.
LEBOWSKI FEST
100,000 ACHIEVERS! PROUD WE ARE OF ALL

App #1

The Good

App #1

You know the environment

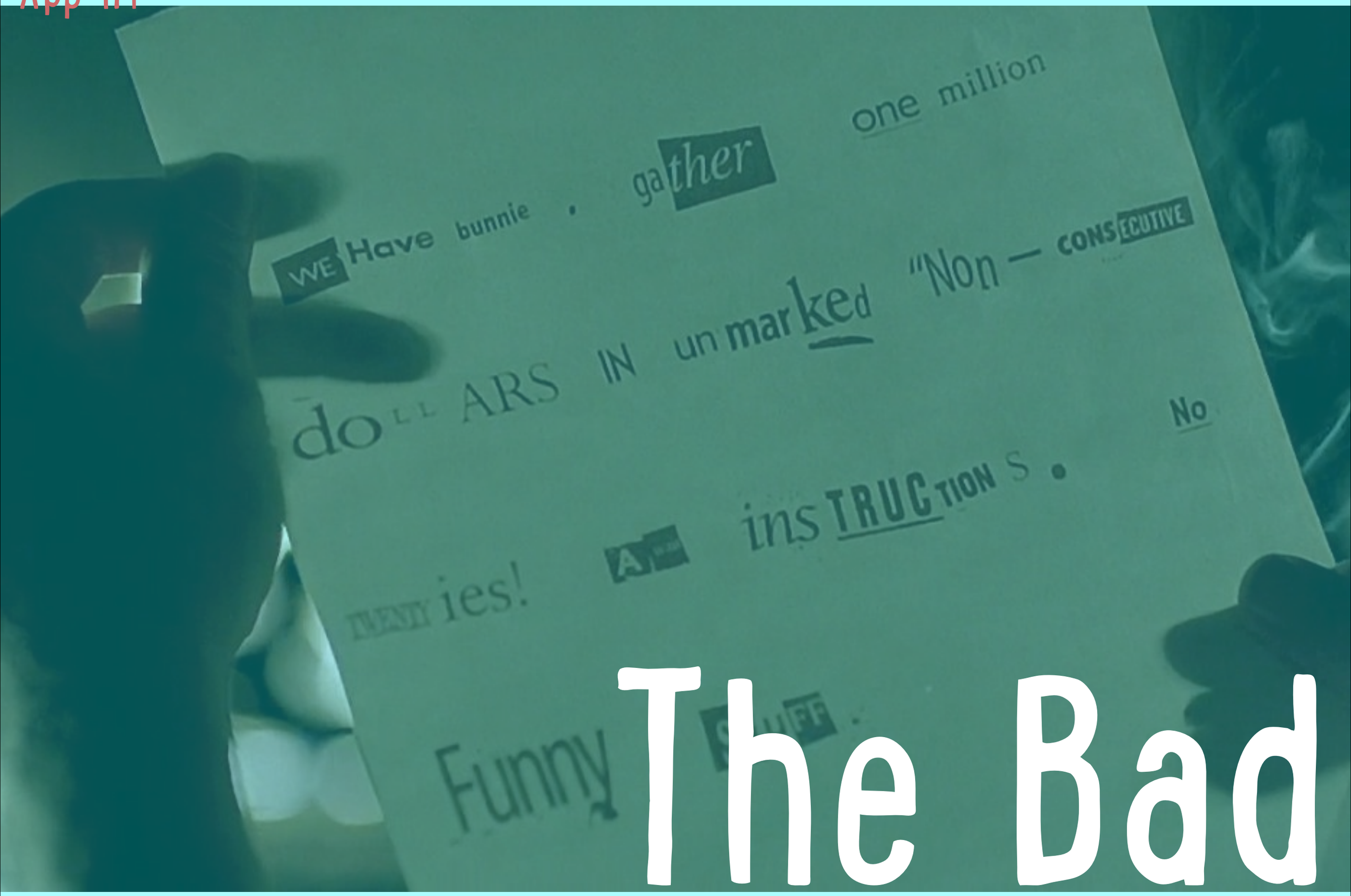
App #1

Caching

App #1

Google can crawl

App #1


WE Have bunnie
gather
one million
ARS IN unmarked "Non— CONSECUTIVE
No
instRUCTIONS.
ies!
Funny

The Bad

App #1

Bad UX in most cases

App #1

Incomplete prototyping

Talk from a Mockup

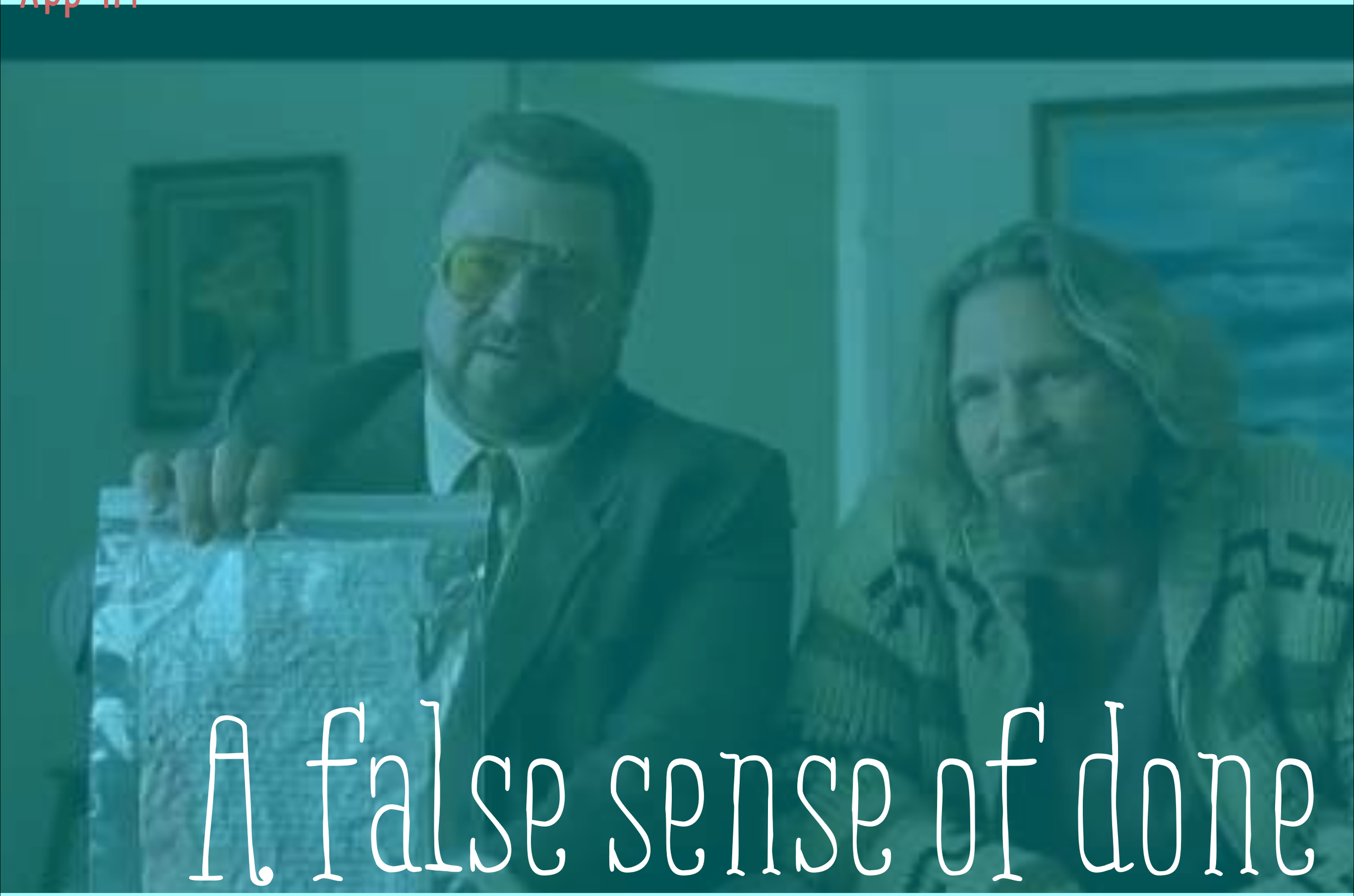
App #1

Incomplete prototyping

App #1

Slower to polish

App #1



A false sense of done

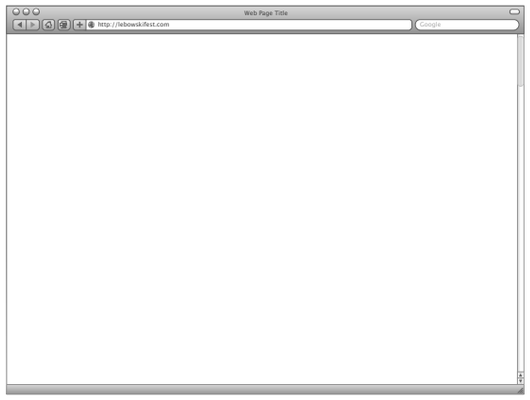
App #1

Boundaries

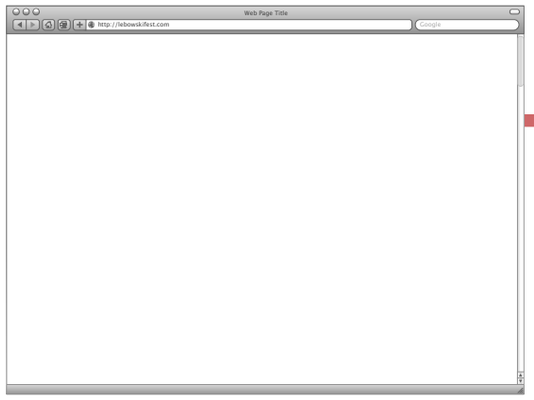
App #2

Client Side Rendering

App #2

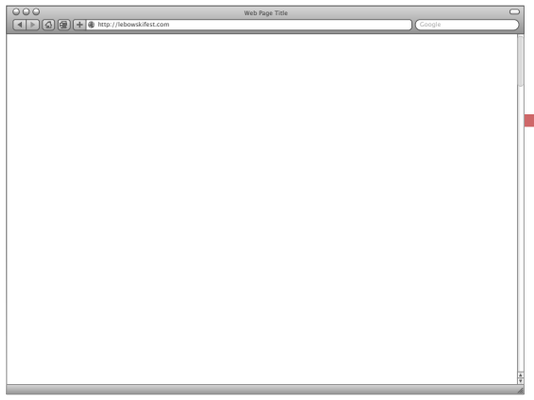


App #2



http://lebowskifest.com

App #2

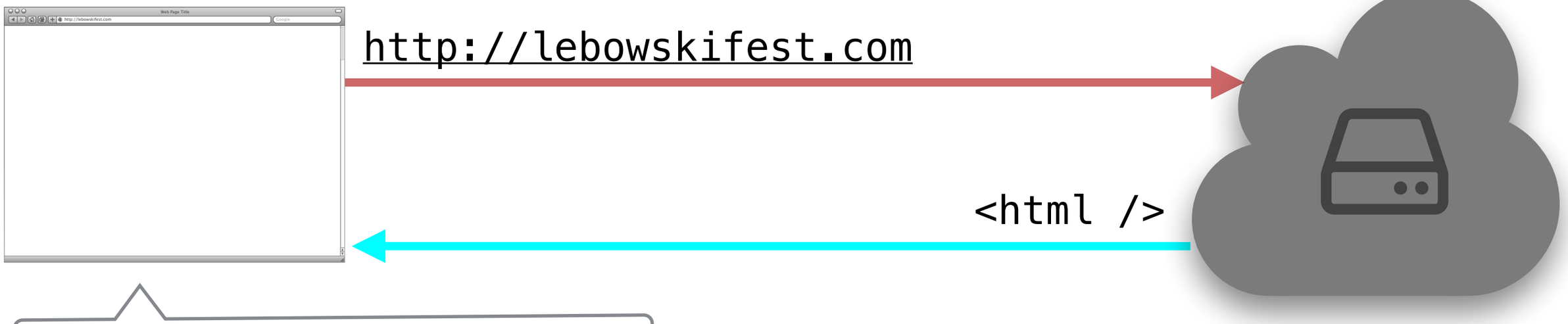


http://lebowskifest.com

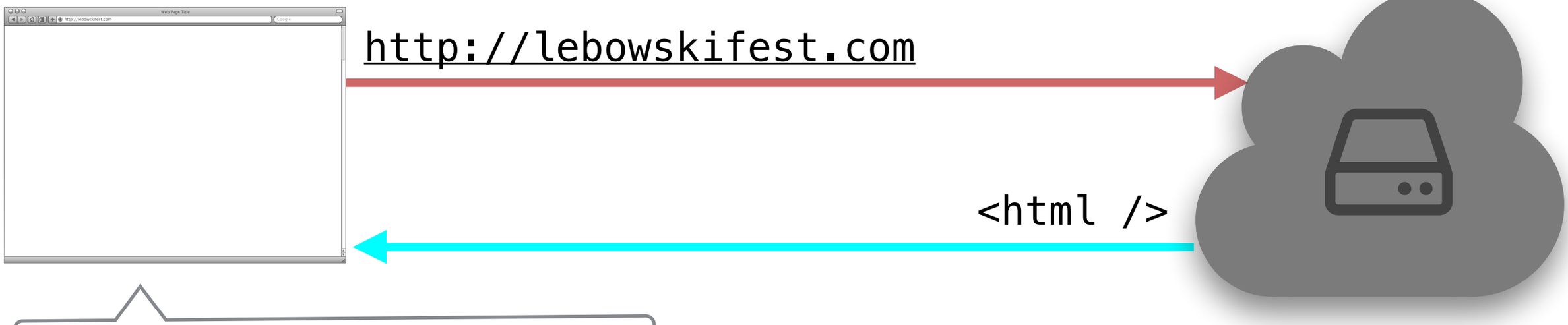
App #2


Web Page Title
http://lebowskifest.com
http://lebowskifest.com
<html />
Google

App #2


Web Page Title
http://lebowskifest.com
http://lebowskifest.com
<html />
1.Browser starts page layout

App #2

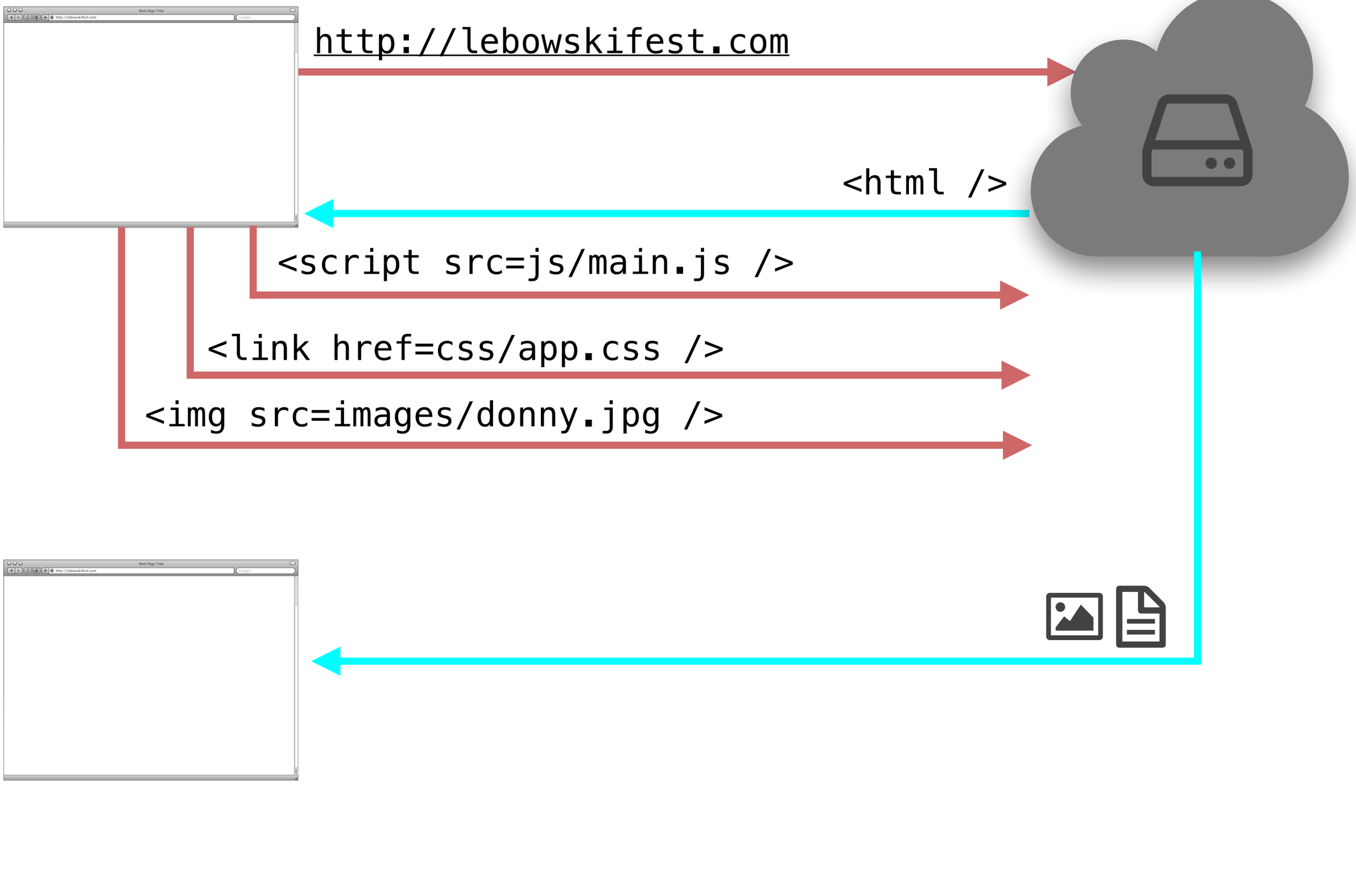


1. Browser starts page layout
2. Additional sources requested

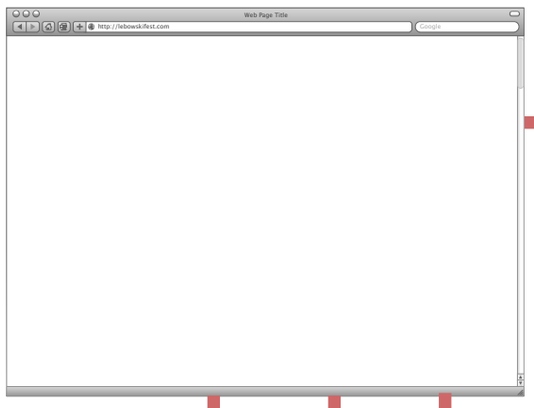
App #2


http://lebowskifest.com
<html />
<script src=js/main.js />
<link href=css/app.css />

App #2


http://lebowskifest.com
<html />
<script src=js/main.js />
<link href=css/app.css />

App #2

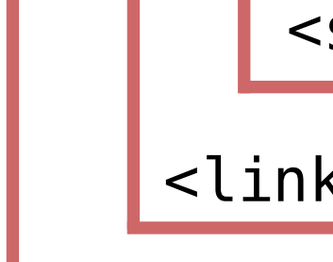


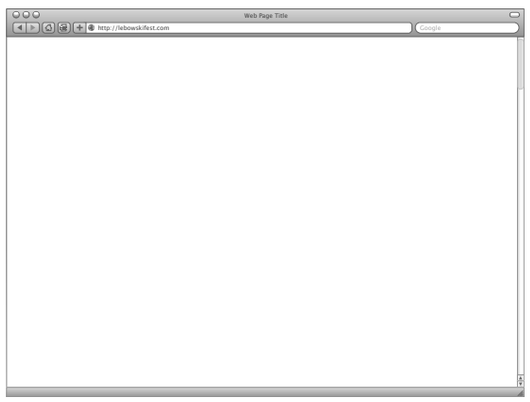
http://lebowskifest.com

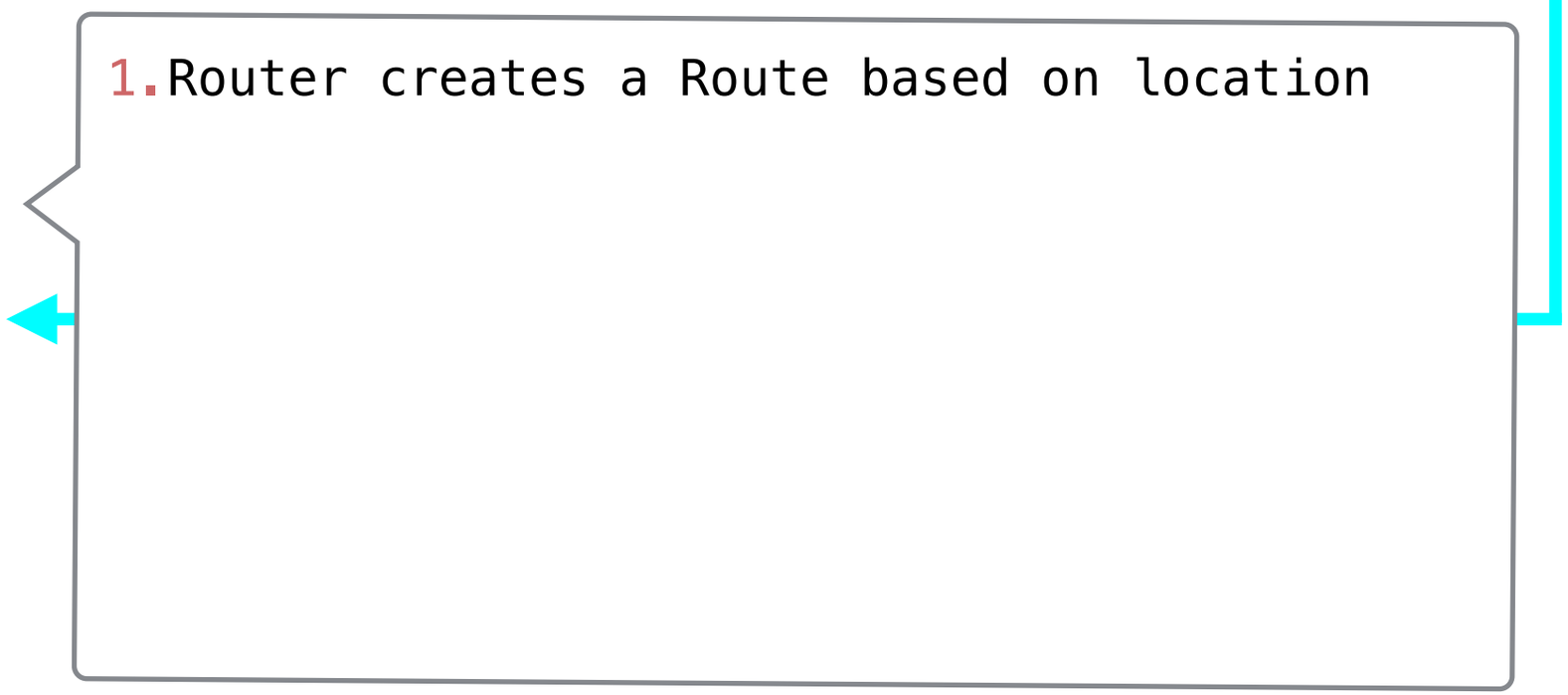
<html />

<script src=js/main.js />

<link href=css/app.css />





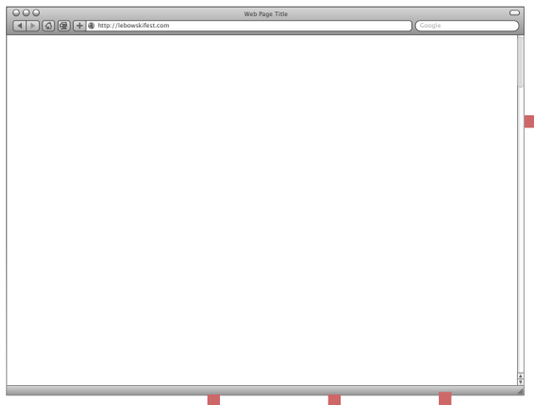


App #2

http://lebowskifest.com/api/scenes

MVC

{}

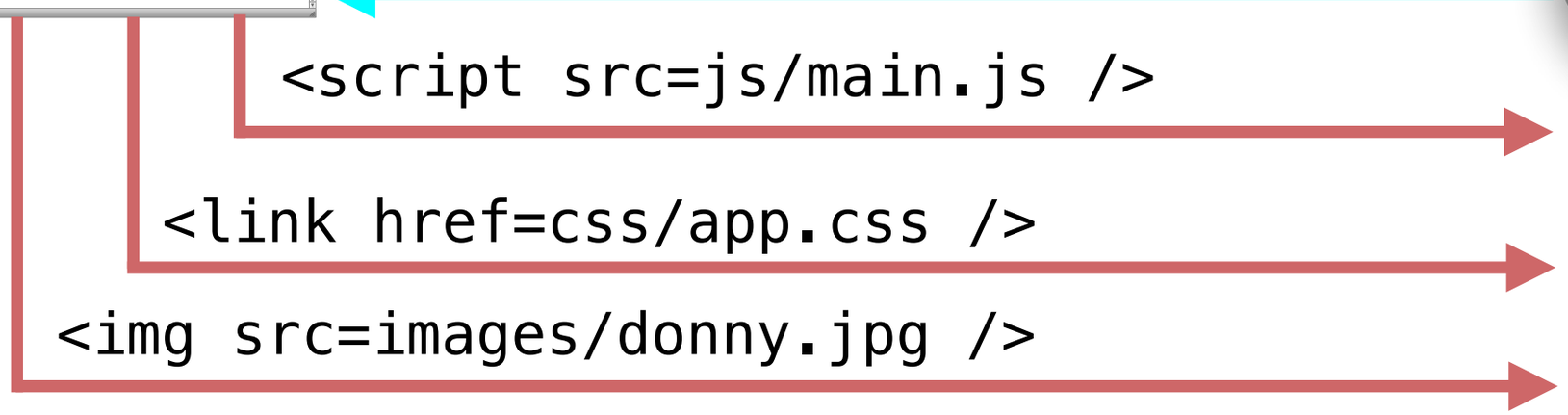


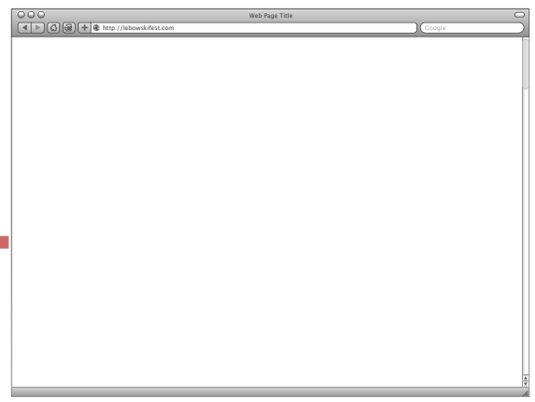
http://lebowskifest.com

<html />

<script src=js/main.js />

<link href=css/app.css />





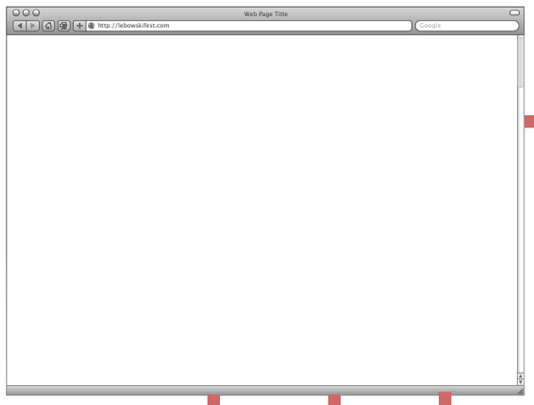
1. Router creates a Route based on location
2. Route loads a Model

App #2

http://lebowskifest.com/api/scenes

MVC

{}

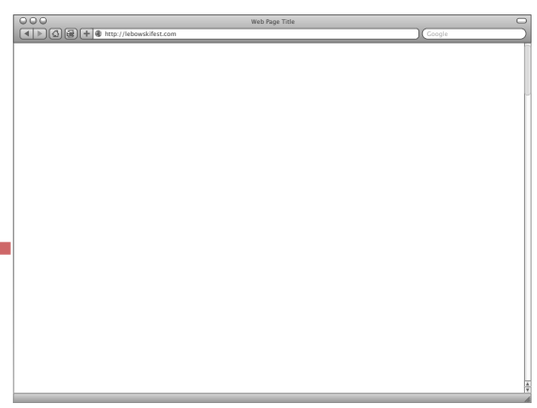


http://lebowskifest.com

<html />

<script src=js/main.js />

<link href=css/app.css />



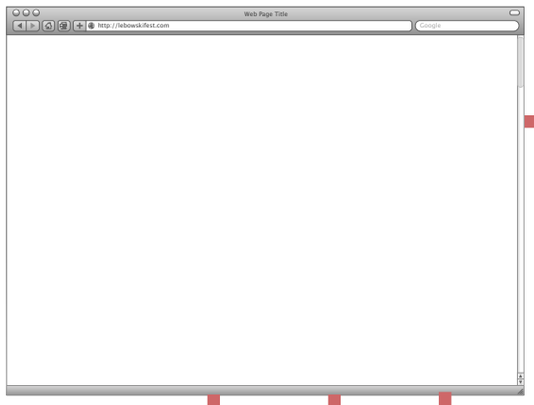
1. Router creates a Route based on location
2. Route loads a Model
3. Route creates a Controller & sets the Model

App #2

http://lebowskifest.com/api/scenes

MVC

{}

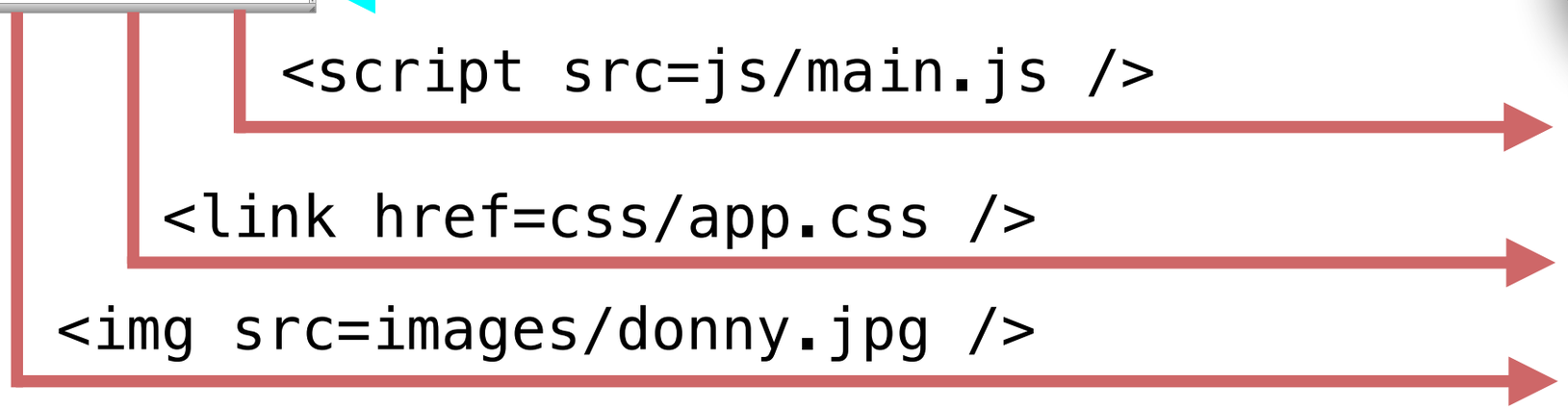


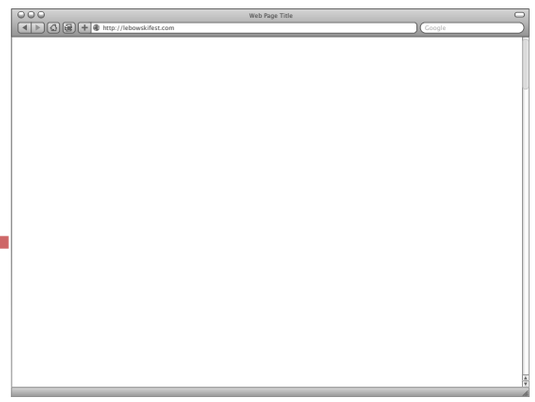
http://lebowskifest.com

<html />

<script src=js/main.js />

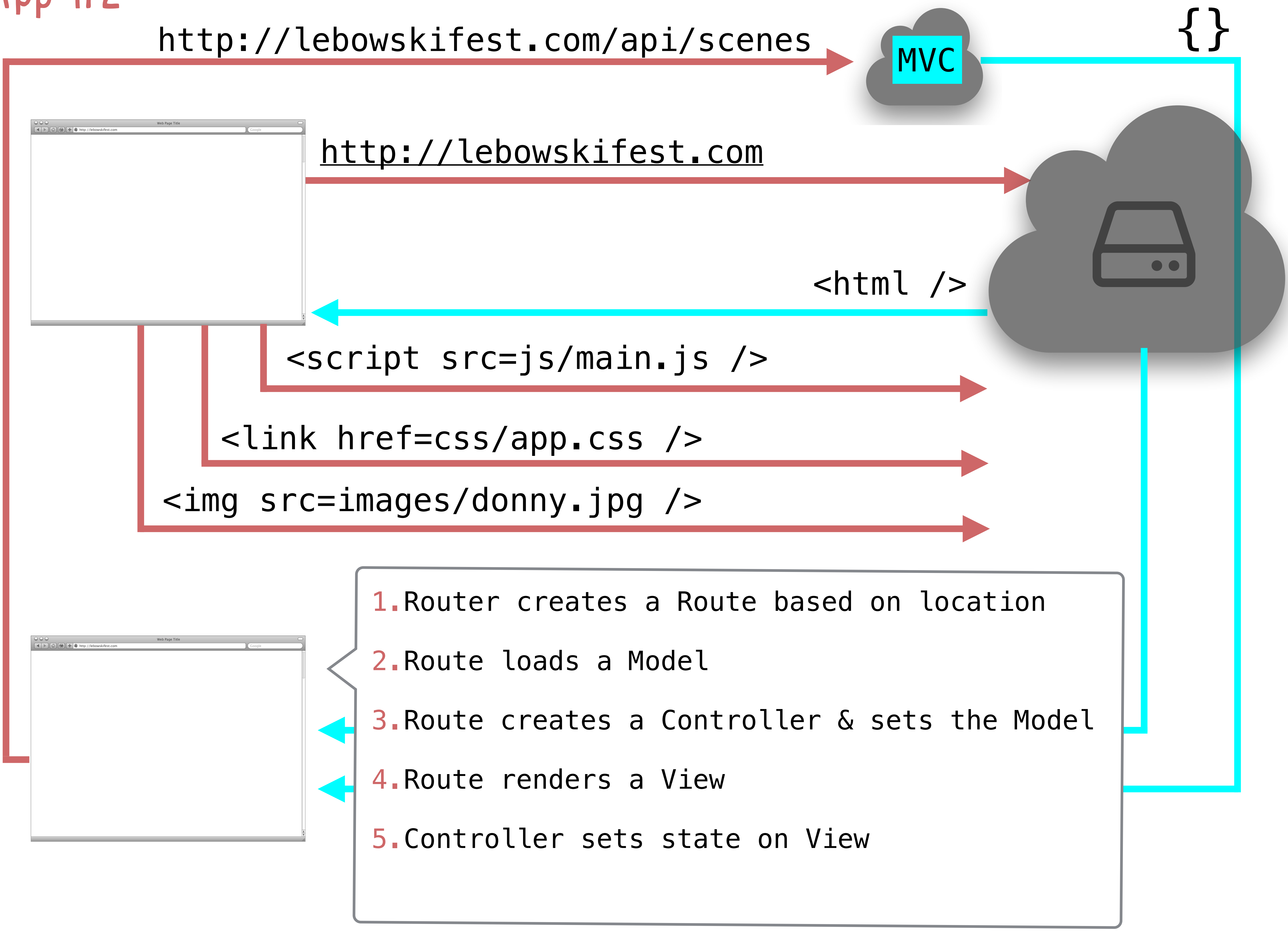
<link href=css/app.css />





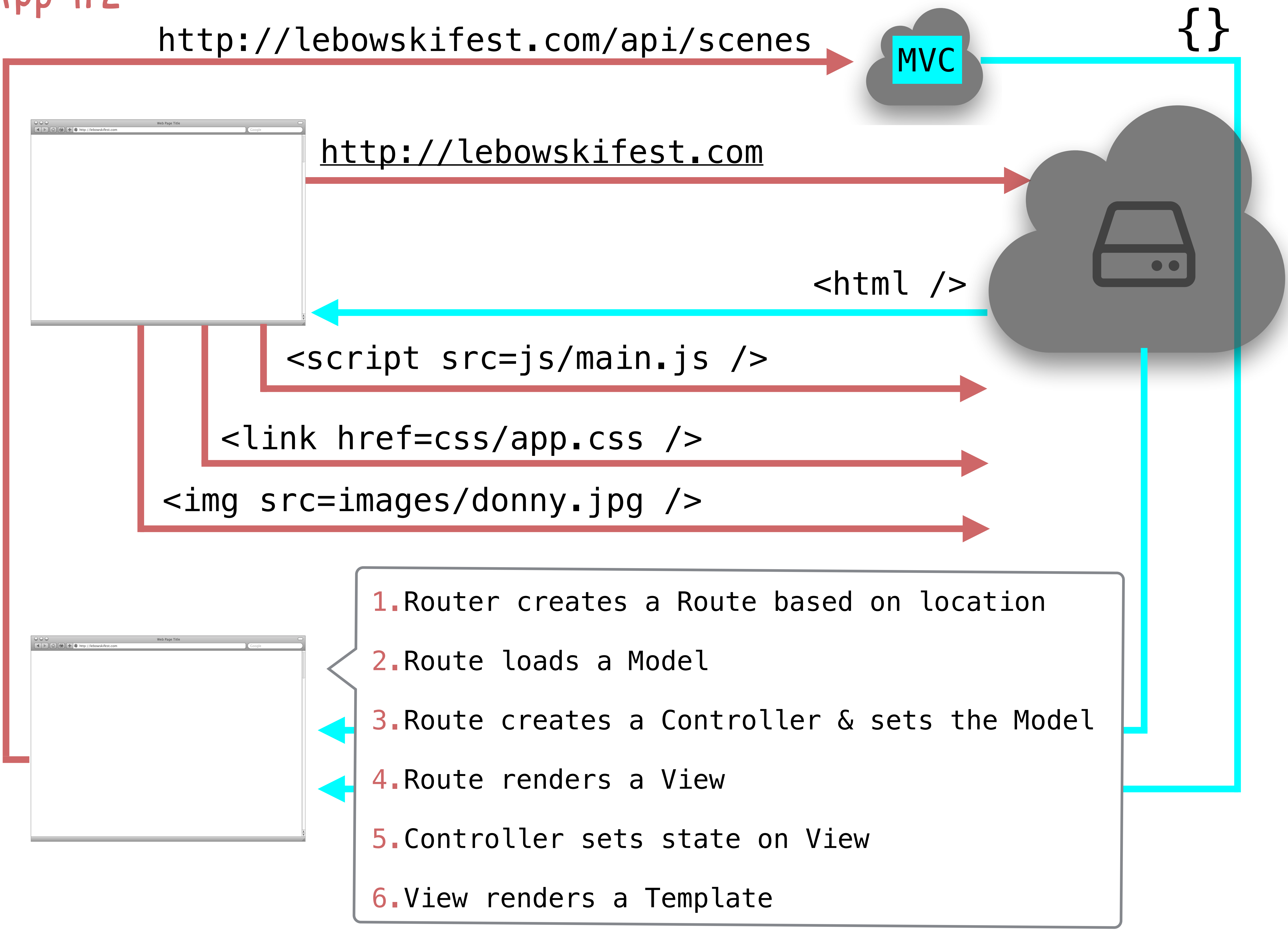
1. Router creates a Route based on location
2. Route loads a Model
3. Route creates a Controller & sets the Model
4. Route renders a View

App #2


http://lebowskifest.com/api/scenes
MVC
{}
http://lebowskifest.com
<html />
<script src=js/main.js />
<link href=css/app.css />

1. Router creates a Route based on location
2. Route loads a Model
3. Route creates a Controller & sets the Model
4. Route renders a View
5. Controller sets state on View

App #2


http://lebowskifest.com/api/scenes
MVC
{}
http://lebowskifest.com
<html />
<script src=js/main.js />
<link href=css/app.css />

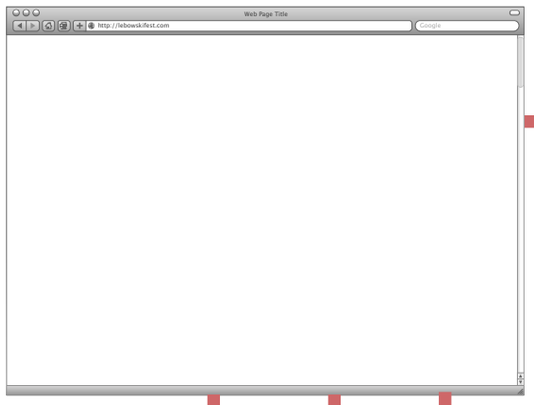
1. Router creates a Route based on location
2. Route loads a Model
3. Route creates a Controller & sets the Model
4. Route renders a View
5. Controller sets state on View
6. View renders a Template

App #2

http://lebowskifest.com/api/scenes

MVC

{}

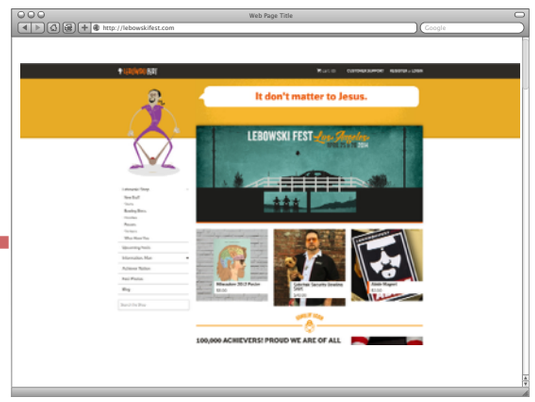


http://lebowskifest.com

<html />

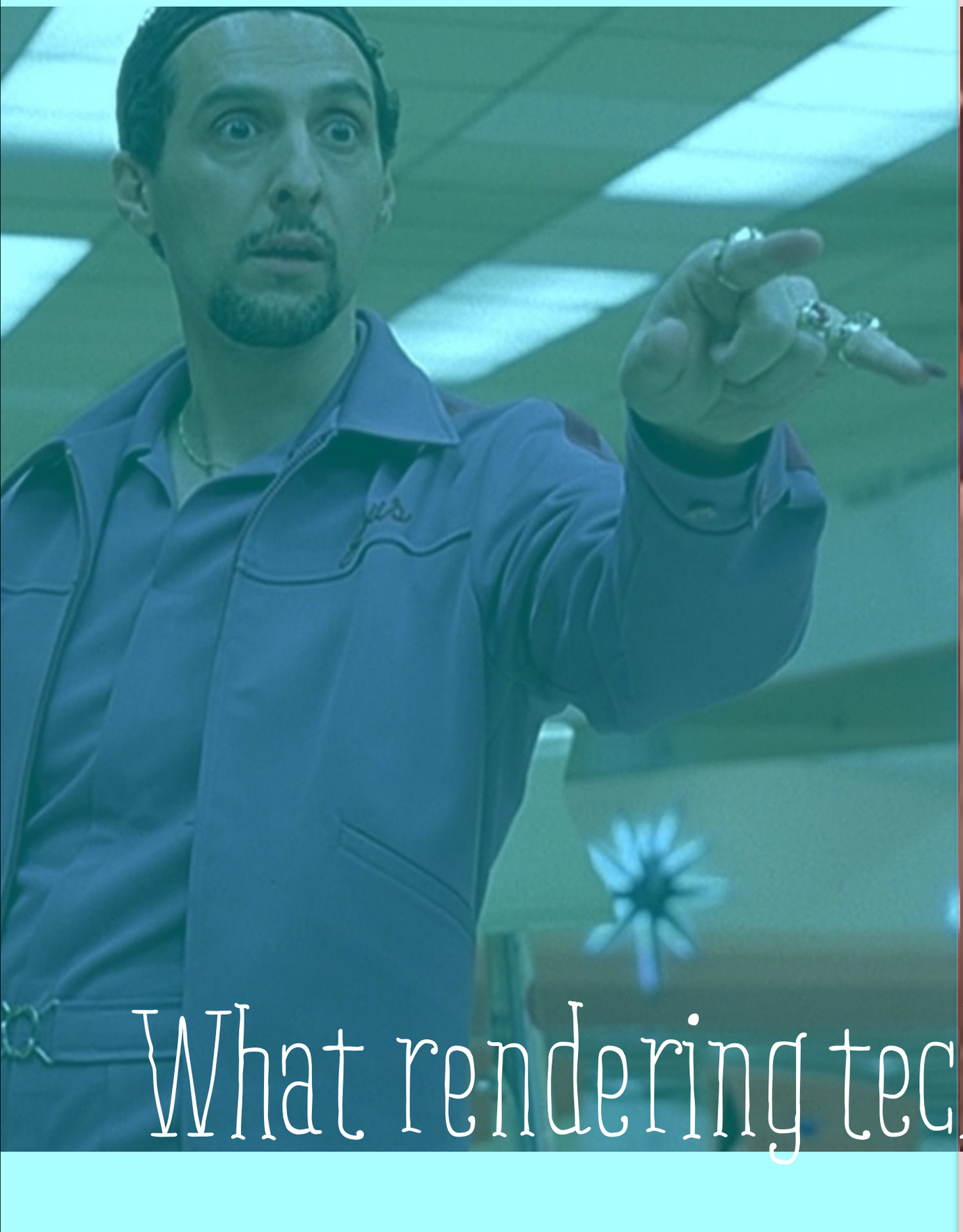
<script src=js/main.js />

<link href=css/app.css />



1. Router creates a Route based on location
2. Route loads a Model
3. Route creates a Controller & sets the Model
4. Route renders a View
5. Controller sets state on View
6. View renders a Template

MVC



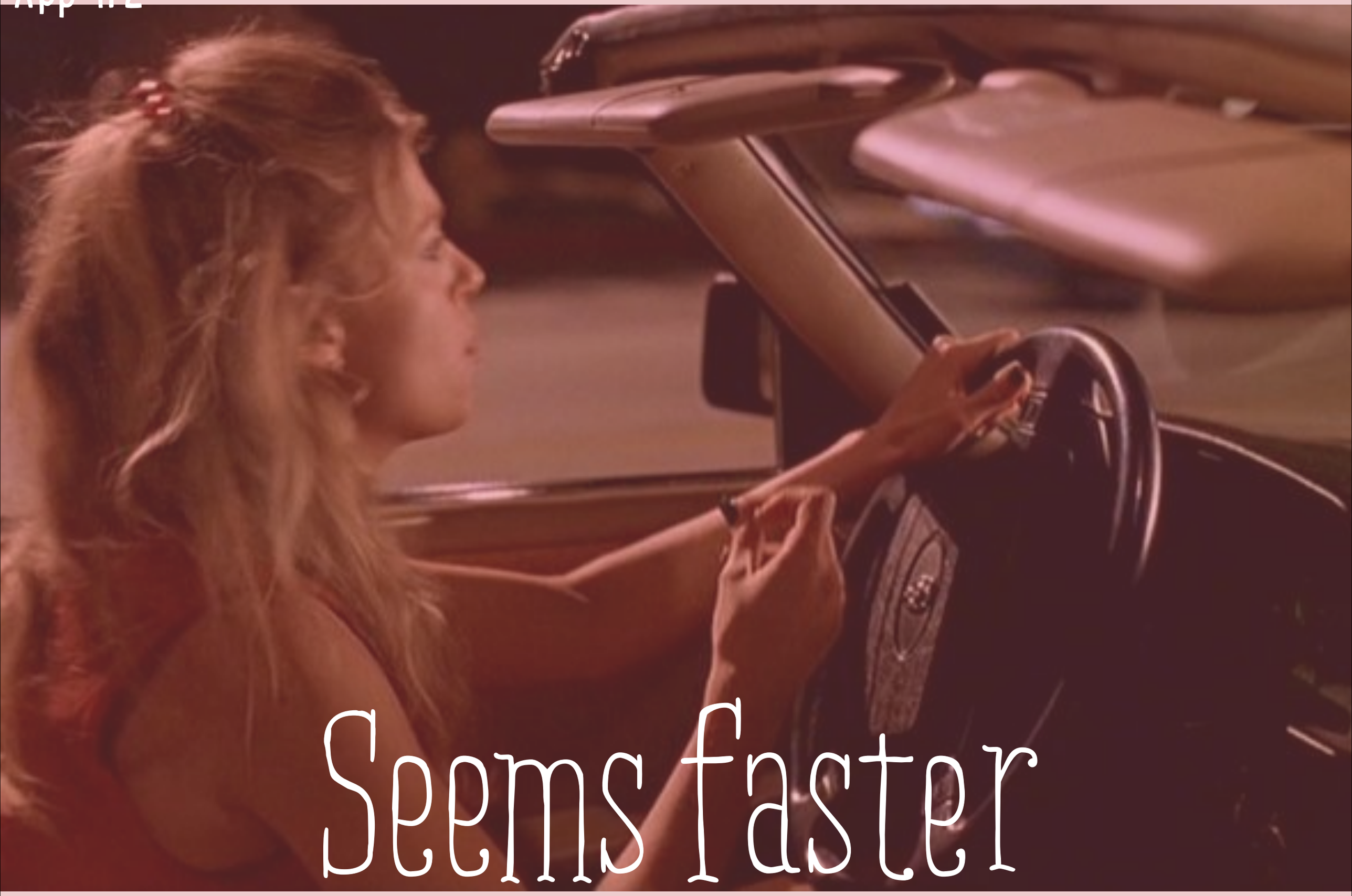
What rendering technique is faster?

Rendering on THE server is FASTER

App #2

The Good

App #2



Seems faster

App #2

Usually leads to good UX

App #2

Clear definition of boundaries

App #2

Faster to polish

App #2

Better sense of done

App #2

Simplest thing first

App #2

First class citizens

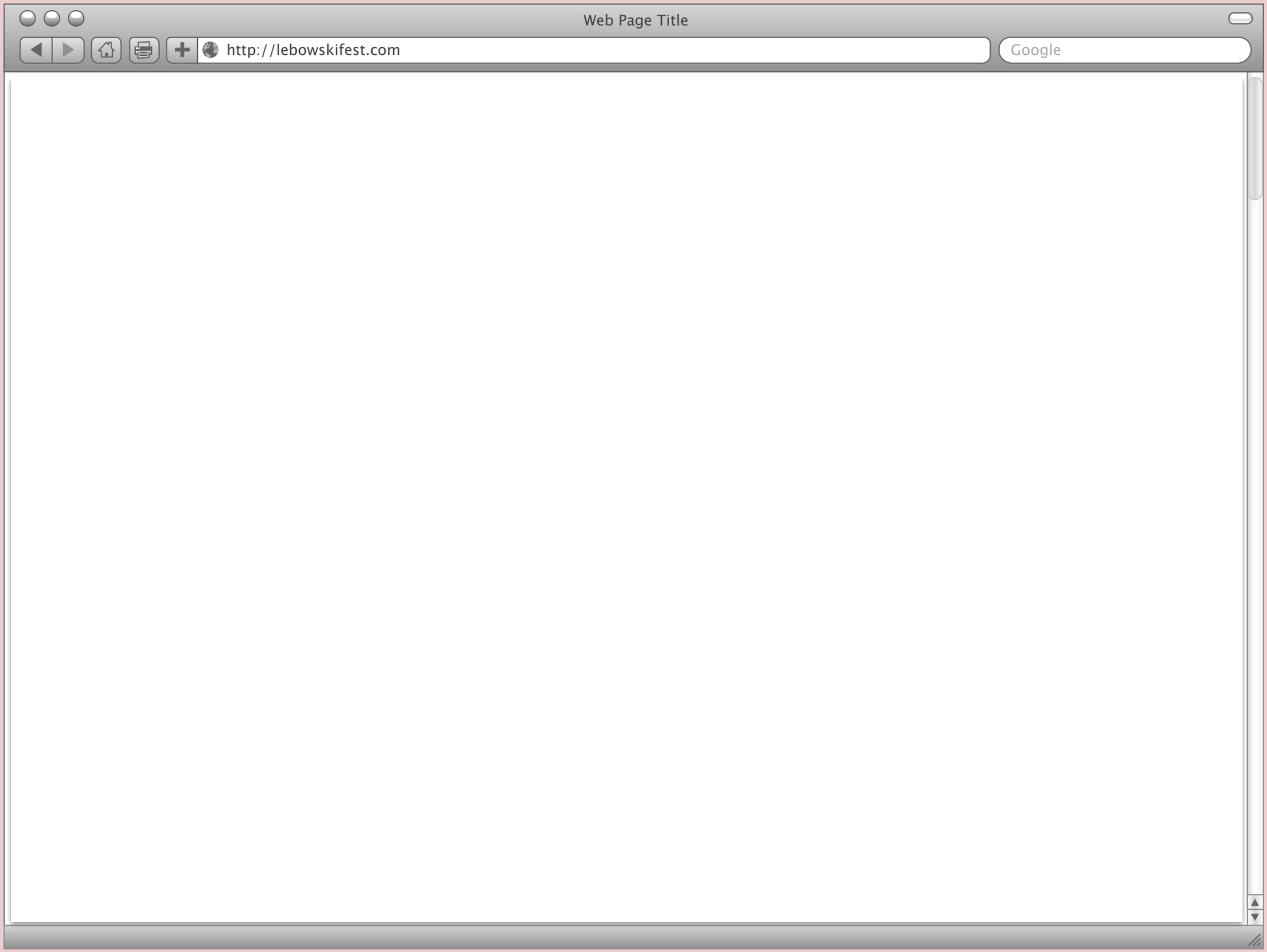
App #2

The Bad

App #2

Error handling

App #2



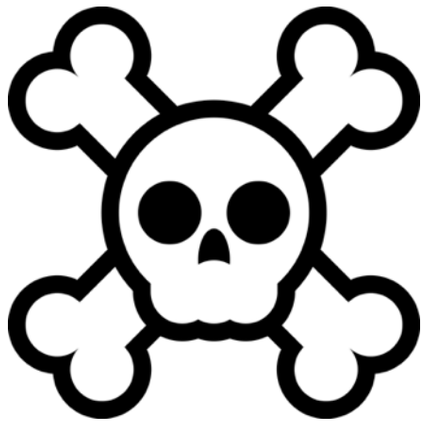
App #2

Web Page Title

http://lebowskifest.com

Google

WHITE PAGE



OF DEATH

App #2

Jump to crawl

App #2

Debugging

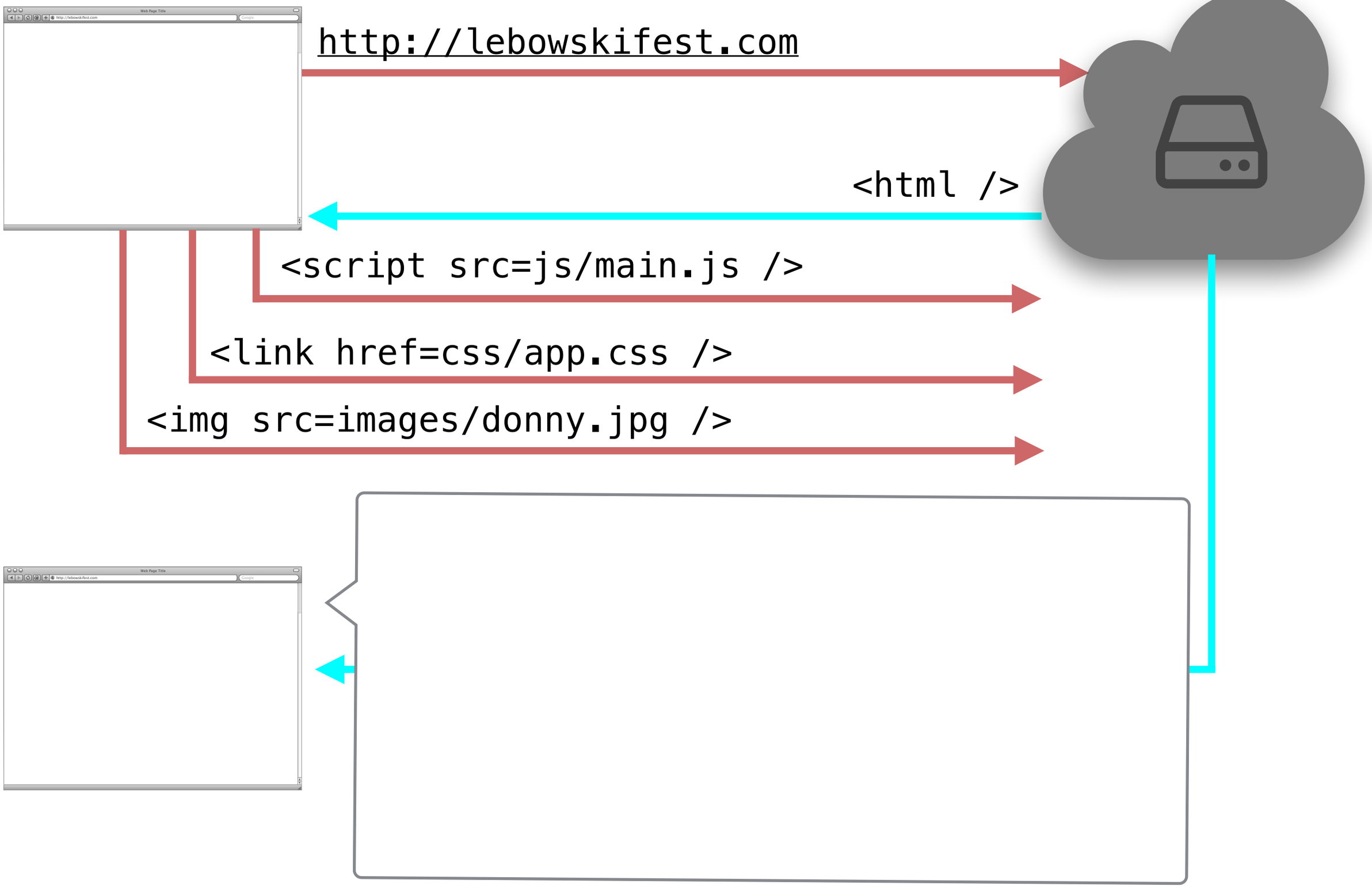
App #2

Progressive Enhancement

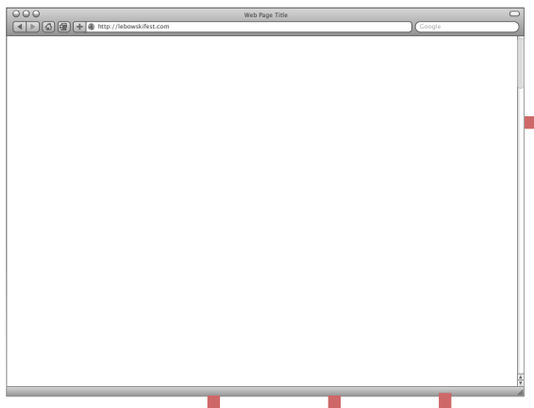
The rug

Ember

App #2


http://lebowskifest.com
<html />
<script src=js/main.js />
<link href=css/app.css />

App #2

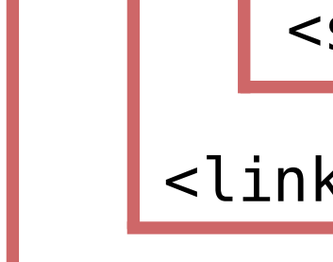


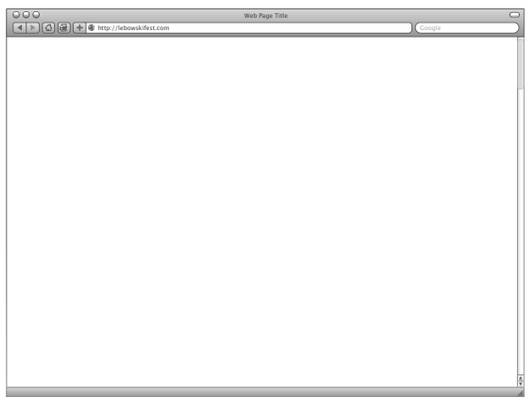
http://lebowskifest.com

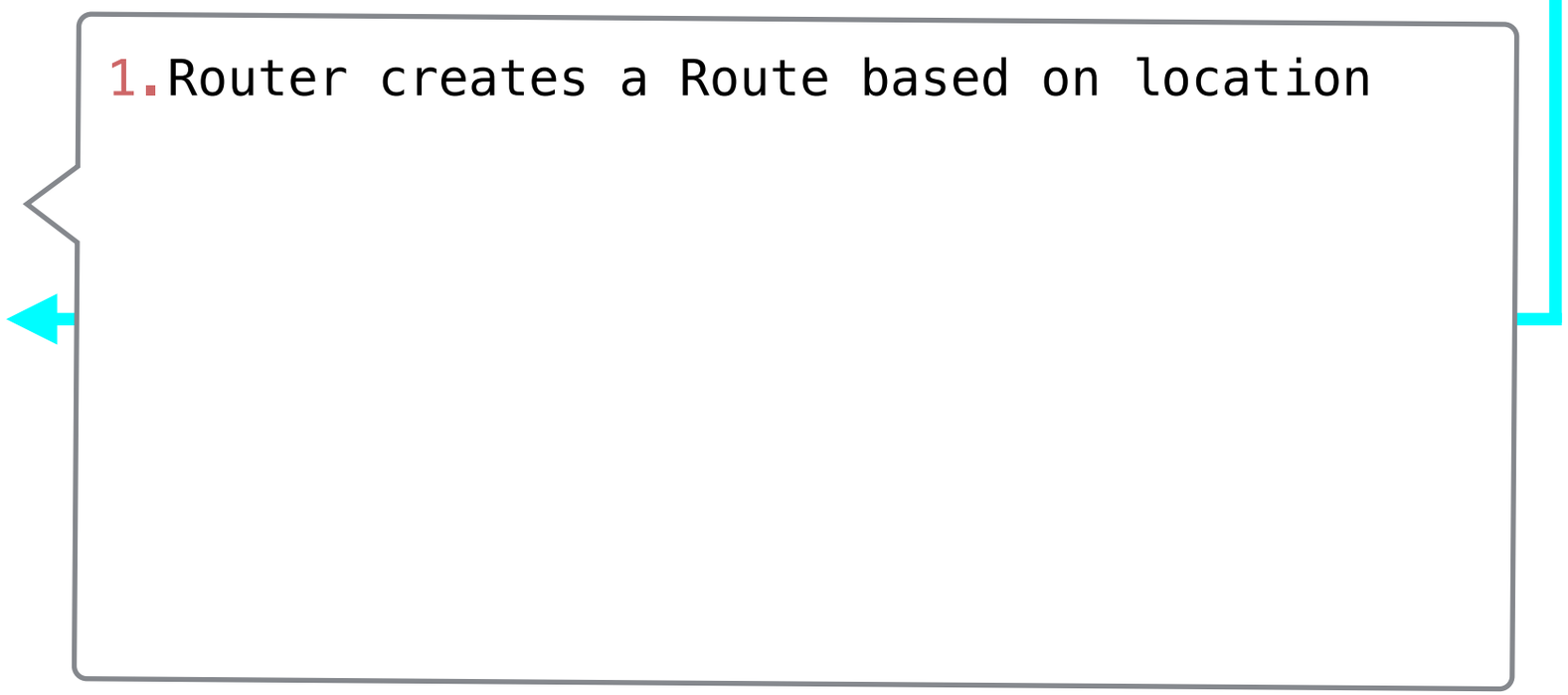
<html />

<script src=js/main.js />

<link href=css/app.css />





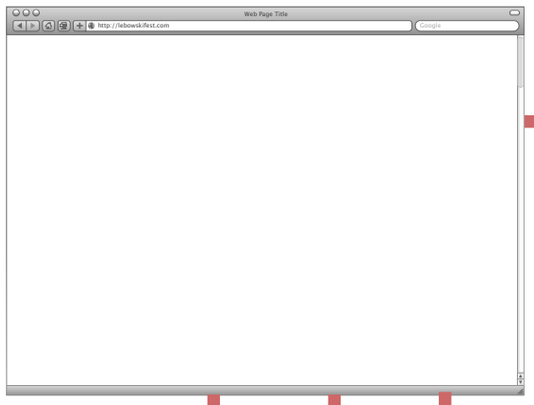


App #2

http://lebowskifest.com/api/scenes

MVC

{}

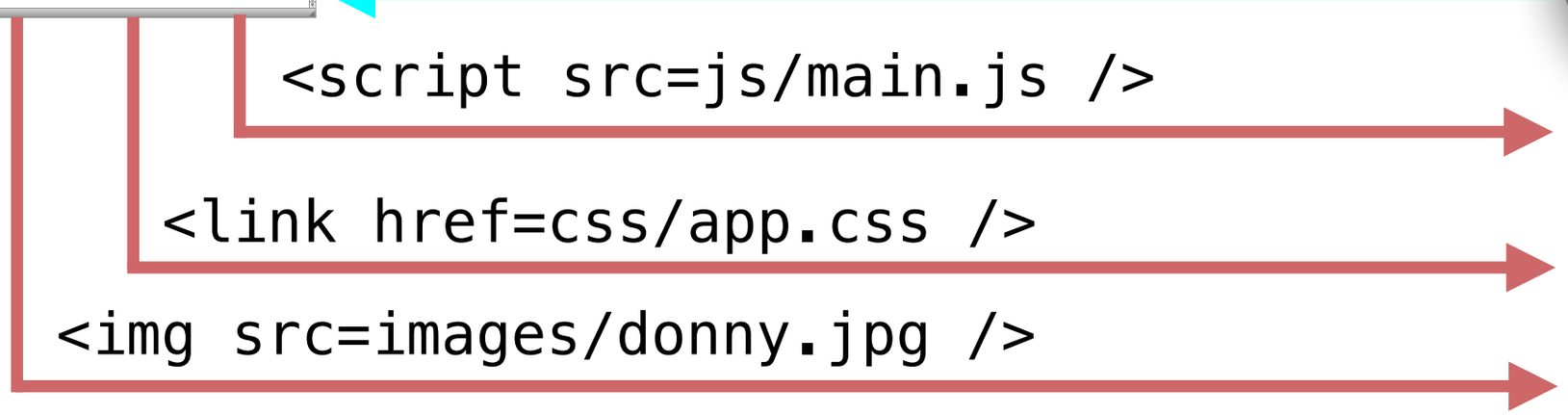


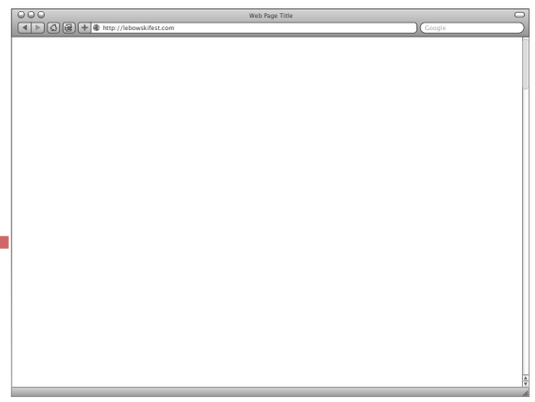
http://lebowskifest.com

<html />

<script src=js/main.js />

<link href=css/app.css />





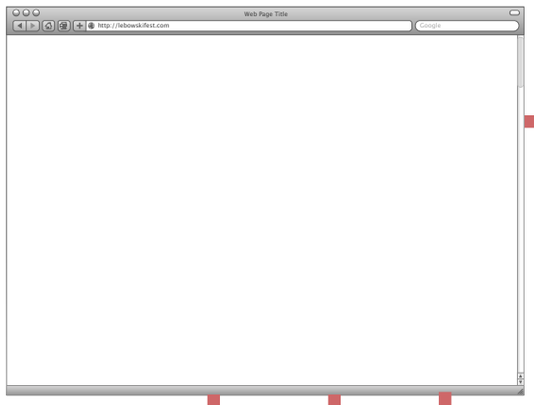
1. Router creates a Route based on location
2. Route loads a Model

App #2

http://lebowskifest.com/api/scenes

MVC

{}

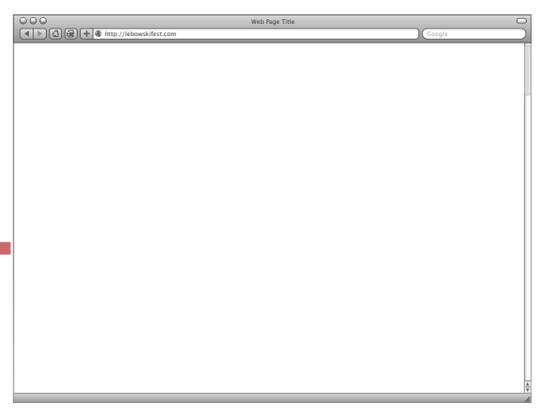


http://lebowskifest.com

<html />

<script src=js/main.js />

<link href=css/app.css />



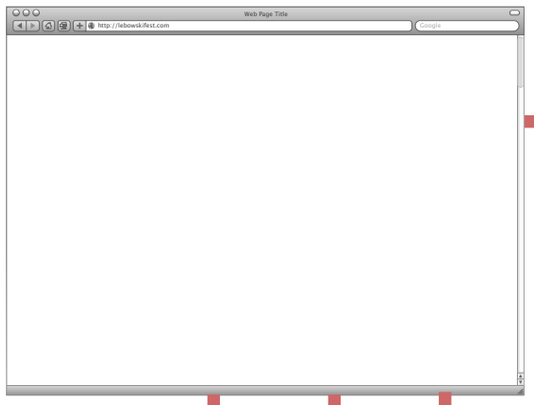
1. Router creates a Route based on location
2. Route loads a Model
3. Route creates a Controller & sets the Model

App #2

http://lebowskifest.com/api/scenes

MVC

{}

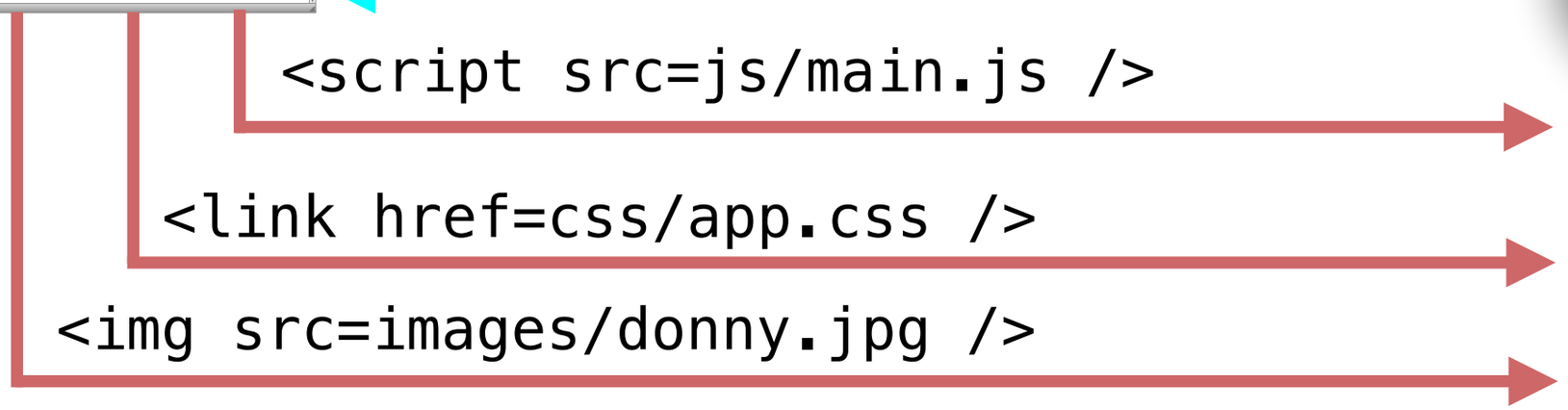


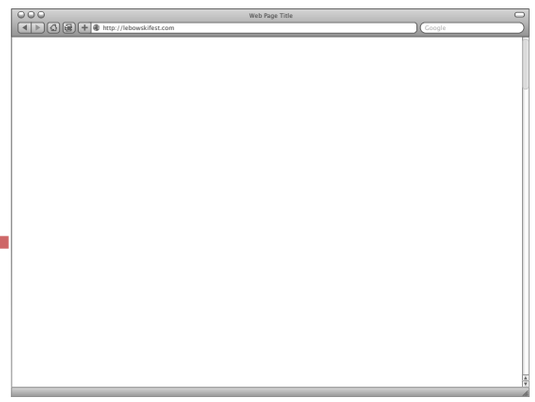
http://lebowskifest.com

<html />

<script src=js/main.js />

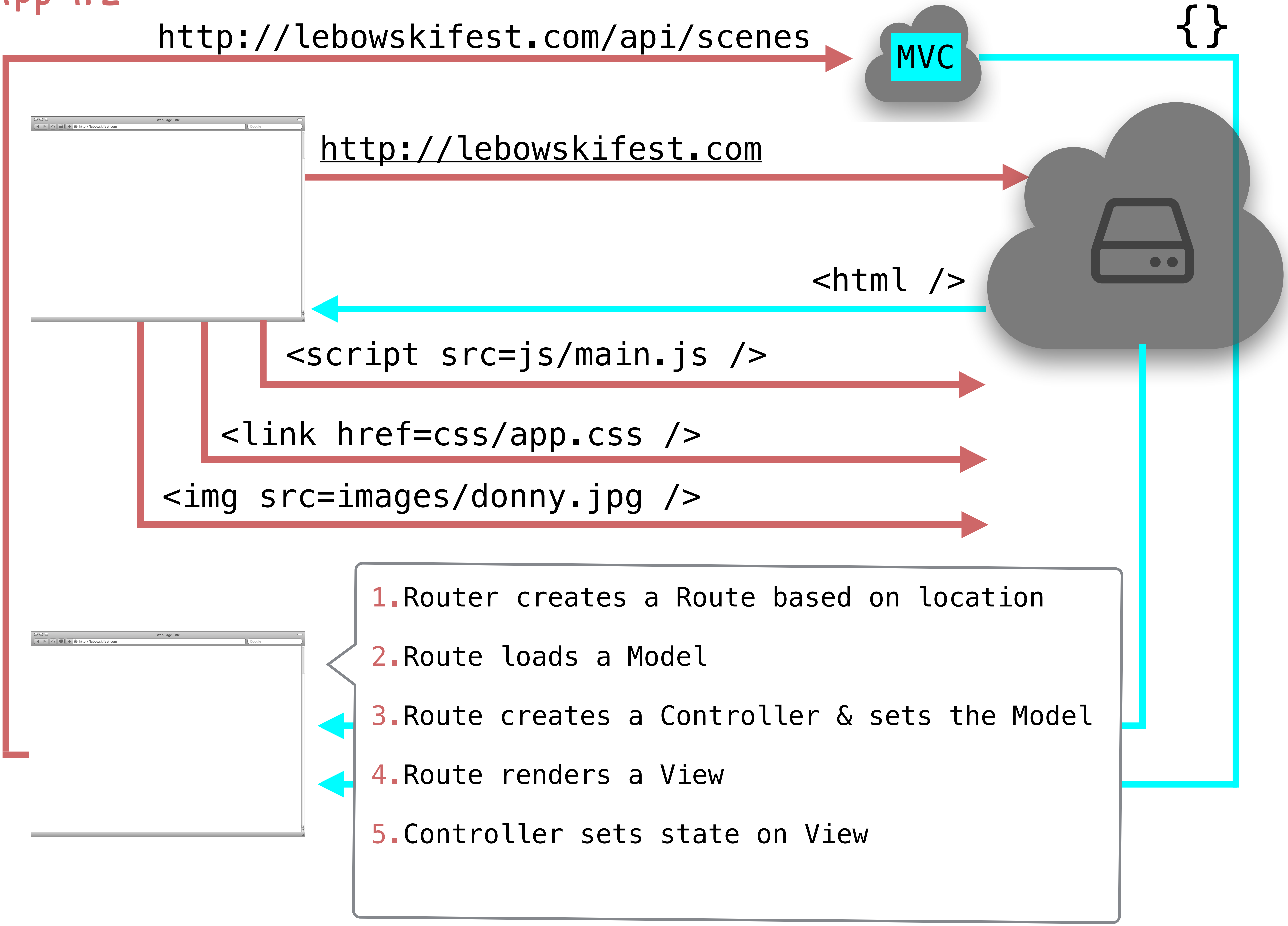
<link href=css/app.css />





1. Router creates a Route based on location
2. Route loads a Model
3. Route creates a Controller & sets the Model
4. Route renders a View

App #2


http://lebowskifest.com/api/scenes
MVC
{}
http://lebowskifest.com
<html />
<script src=js/main.js />
<link href=css/app.css />

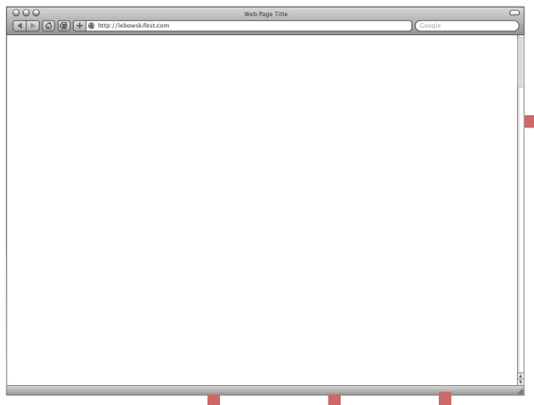
1. Router creates a Route based on location
2. Route loads a Model
3. Route creates a Controller & sets the Model
4. Route renders a View
5. Controller sets state on View

App #2

http://lebowskifest.com/api/scenes

MVC

{}

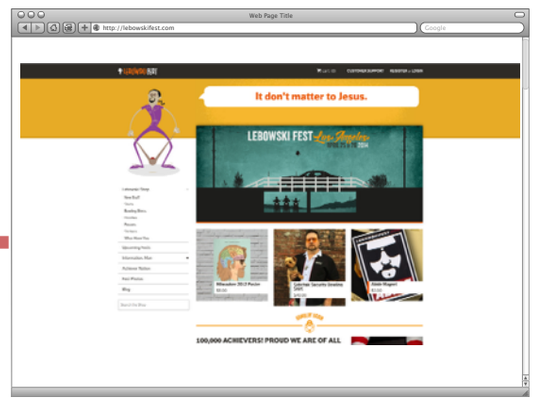


http://lebowskifest.com

<html />

<script src=js/main.js />

<link href=css/app.css />



1. Router creates a Route based on location
2. Route loads a Model
3. Route creates a Controller & sets the Model
4. Route renders a View
5. Controller sets state on View
6. View renders a Template

The rug

```
<script>
  window.App = Ember.Application.create();
</script>
```

Application

The rug

```
<script>
  window.App = Ember.Application.create();
</script>
```

Router

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.route("about");
  });
</script>
```

/about

Router

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.route("about");
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project",
                  { path: "/projects/:project_id" });
  });
</script>
```

/about

/projects

/projects/new

/projects/:project_id

Router

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });
</script>
```

Routes

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });

  App.ProjectsIndexRoute = Ember.Route.extend({});
</script>
```

Routes

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });

  App.ProjectsIndexRoute = Ember.Route.extend({});
  App.ProjectsNewRoute = Ember.Route.extend({});
</script>
```

Routes

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });

  App.ProjectsIndexRoute = Ember.Route.extend({});
  App.ProjectsNewRoute = Ember.Route.extend({});
  App.ProjectRoute = Ember.Route.extend({});
</script>
```

Routes

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });

  App.ProjectsIndexRoute = Ember.Route.extend({});
  App.ProjectsNewRoute = Ember.Route.extend({});
  App.ProjectRoute = Ember.Route.extend({});
</script>
```

Models

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });

  App.ProjectsIndexRoute = Ember.Route.extend({
    model: function() {
      return this.store.findAll('project');
    }
  });

  App.ProjectsNewRoute = Ember.Route.extend({});
  App.ProjectRoute = Ember.Route.extend({});
</script>
```

Models

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });

  App.ProjectsIndexRoute = Ember.Route.extend({
    model: function() {
      return this.store.findAll('project');
    }
  });

  App.ProjectsNewRoute = Ember.Route.extend({});
  App.ProjectRoute = Ember.Route.extend({});
</script>
```

GET /api/projects

Models

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });

  App.ProjectsIndexRoute = Ember.Route.extend({
    model: function() {
      return this.store.findAll('project');
    }
  });

  App.ProjectsNewRoute = Ember.Route.extend({
    model: function() {
      return App.Project.create();
    }
  });

  App.ProjectRoute = Ember.Route.extend({});
</script>
```

Models

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });

  App.ProjectsIndexRoute = Ember.Route.extend({
    model: function() {
      return this.store.findAll('project');
    }
  });

  App.ProjectsNewRoute = Ember.Route.extend({
    model: function() {
      return App.Project.create();
    }
  });

  App.ProjectRoute = Ember.Route.extend({
    model: function(params) {
      return this.store.find('project', params.project_id);
    }
  });
</script>
```

Models

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });

  App.ProjectsIndexRoute = Ember.Route.extend({
    model: function() {
      return this.store.findAll('project');
    }
  });

  App.ProjectsNewRoute = Ember.Route.extend({
    model: function() {
      return App.Project.create();
    }
  });

  App.ProjectRoute = Ember.Route.extend({
    model: function(params) {
      return this.store.find('project', params.project_id);
    }
  });
</script>
```

GET /api/projects/65

Models

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });

  App.ProjectsIndexRoute = Ember.Route.extend({
    model: function() {
      return this.store.findAll('project');
    }
  });

  App.ProjectsNewRoute = Ember.Route.extend({
    model: function() {
      return App.Project.create();
    }
  });

  App.ProjectRoute = Ember.Route.extend({
    model: function(params) {
      return this.store.find('project', params.project_id);
    }
  });
</script>
```

Controllers

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });

  App.ProjectsIndexRoute = Ember.Route.extend({
    model: function() {
      return this.store.findAll('project');
    }
  });

  App.ProjectsNewRoute = Ember.Route.extend({...});
  App.ProjectRoute = Ember.Route.extend({...});
</script>
```

Controllers

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });

  App.ProjectsIndexRoute = Ember.Route.extend({
    model: function() {
      return this.store.findAll('project');
    }
  });

  App.ProjectsNewRoute = Ember.Route.extend({...});
  App.ProjectRoute = Ember.Route.extend({...});

  App.ProjectsIndexController = Ember.ArrayController.extend({});
</script>
```

Controllers

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });

  App.ProjectsIndexRoute = Ember.Route.extend({
    model: function() {
      return this.store.findAll('project');
    }
  });

  App.ProjectsNewRoute = Ember.Route.extend({...});
  App.ProjectRoute = Ember.Route.extend({...});

  App.ProjectsIndexController = Ember.ArrayController.extend({
    sortProperties: ['name', 'createdAt'];
    sortAscending: true;
  });
</script>
```

Controllers

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });

  App.ProjectsIndexRoute = Ember.Route.extend({...});
  App.ProjectsNewRoute = Ember.Route.extend({...});
  App.ProjectRoute = Ember.Route.extend({...});

  App.ProjectsIndexController = Ember.ArrayController.extend({...});
  App.ProjectsNewController = Ember.ObjectController.extend({
    actions: {
      resetForm: function() {
        this.get('model').rollback();
      }
    }
  });
</script>
```

Controllers

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });

  App.ProjectsIndexRoute = Ember.Route.extend({...});
  App.ProjectsNewRoute = Ember.Route.extend({...});
  App.ProjectRoute = Ember.Route.extend({...});

  App.ProjectsIndexController = Ember.ArrayController.extend({...});
  App.ProjectsNewController = Ember.ObjectController.extend({});
  App.ProjectController = Ember.ObjectController.extend({});
</script>
```

Views

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });

  App.ProjectsNewRoute = Ember.Route.extend({...});

  App.ProjectsNewController = Ember.ObjectController.extend({...});
</script>
```

Views

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });

  App.ProjectsNewRoute = Ember.Route.extend({...});

  App.ProjectsNewController = Ember.ObjectController.extend({...});

  App.ProjectsNewView = Ember.View.extend({
    click: function(event) {
      this.get('controller').send('resetForm');
    }
  });
</script>
```

Views

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });

  App.ProjectsIndexRoute = Ember.Route.extend({...});
  App.ProjectsNewRoute = Ember.Route.extend({...});
  App.ProjectRoute = Ember.Route.extend({...});

  App.ProjectsIndexController = Ember.ArrayController.extend({...});
  App.ProjectsNewController = Ember.ObjectController.extend({...});
  App.ProjectController = Ember.ObjectController.extend({});

  App.ProjectsIndexView = Ember.View.extend({});
  App.ProjectsNewView = Ember.View.extend({});
  App.ProjectView = Ember.View.extend({});
</script>
```

Templates

The rug

```
<script>
  window.App = Ember.Application.create();

  App.Router.reopen({
    this.resource("projects", function() {
      this.route("new");
    });
    this.resource("project", { path: "/projects/:project_id" });
  });

  App.ProjectsIndexRoute = Ember.Route.extend({...});
  App.ProjectsNewRoute = Ember.Route.extend({...});
  App.ProjectRoute = Ember.Route.extend({...});

  App.ProjectsIndexController = Ember.ArrayController.extend({...});
  App.ProjectsNewController = Ember.ObjectController.extend({...});
  App.ProjectController = Ember.ObjectController.extend({});

  App.ProjectsIndexView = Ember.View.extend({});
  App.ProjectsNewView = Ember.View.extend({});
  App.ProjectView = Ember.View.extend({});
</script>

<script type="text/x-handlebars">
  <div>
    {{outlet}}
  </div>
</script>
```

Templates

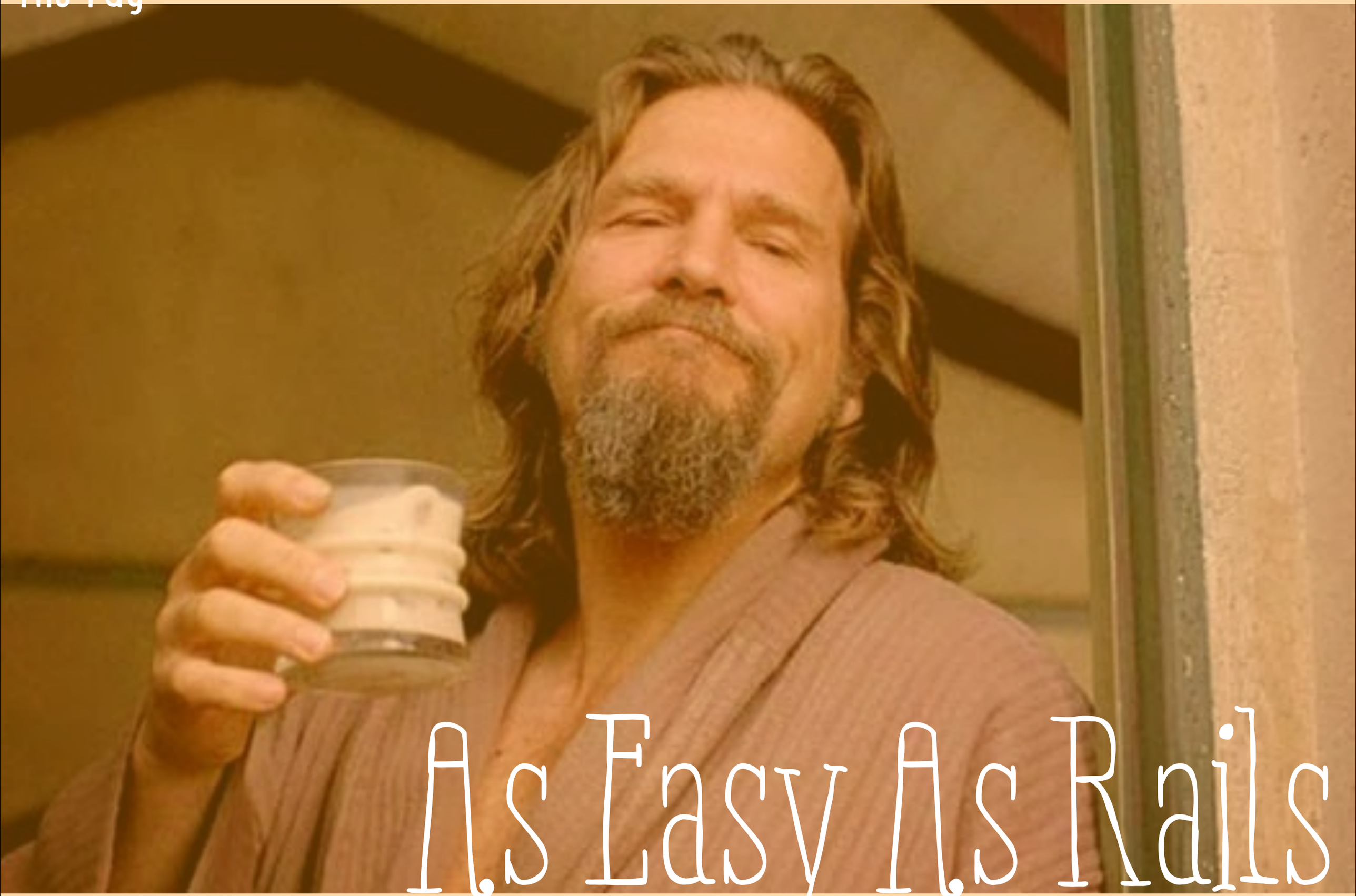
The rug

```
<script type="text/x-handlebars" data-template-name="projects/index">
  <table>
    <thead>
      <tr>
        <th>Name</th>
        <th>Created</th>
      </tr>
    </thead>
    <tbody>
      {{#each}}
        <tr>
          <td>{{name}}</td>
          <td>{{formattedDate  format='LL' value=createdAt}}</td>
        </tr>
      {{/each}}
    </tbody>
  </table>
</script>
```

Templates

The rug

Conventions



As Easy As Rails

The rug

Getting to done

The rug

```
<script>

module("Project Creation", {
  setup: function() {
    App.reset();
  }
});

test("creating a project displays the new project", function(){
  visit("/projects/new");
  fillIn("input[placeholder='Name']", "A new project");
  click(".submit");
  andThen(function() {
    ok(find("h1:contains('A new project')").length, "The
project's name should display");
  });
});

</script>
```

Integration Testing

The rug

```
<script>
  var model = null;
  var controller = null;

  module "App.ProjectsNewController::validations", {
    setup: function() {
      App.reset();
      this.model = Ember.Object.create();
      this.controller =
App.ProjectsNewController.create(content: @model);
    }
  });

  test("name must be present", function() {
    this.controller.set('name', '')
    errors = this.controller.get('errors')
    deepEqual errors.name, ["can't be blank"]
  });
</script>
```

Unit Testing

The rug

```
<script>
  window.App = Ember.Application.create();

  App.ProjectsIndexRoute = Ember.Route.extend({
    model: function() {
      return this.store.findAll('project');
    }
  });
</script>
```

Fixtures

The rug

```
<script>
  window.App = Ember.Application.create();

  App.ProjectsIndexRoute = Ember.Route.extend({
    model: function() {
      return this.store.findAll('project');
    }
  });

  App.ApplicationAdapter = DS.FixtureAdapter.extend();
</script>
```

Fixtures

The rug

```
<script>
  window.App = Ember.Application.create();

  App.ProjectsIndexRoute = Ember.Route.extend({
    model: function() {
      return this.store.findAll('project');
    }
  });

  App.ApplicationAdapter = DS.FixtureAdapter.extend();

  App.Project.FIXTURES = [
    {
      id: 1,
      name: 'The Jeffery Lebowski Project',
      createdAt: new Date()
    },
    {
      id: 2,
      name: 'The Dude\'s Project',
      createdAt: new Date()
    }
  ];
</script>
```

Fixtures

The rug

```
<script>
  window.App = Ember.Application.create();

  App.ProjectsIndexRoute = Ember.Route.extend({
    model: function() {
      return this.store.findAll('project');
    }
  });

  App.ProjectAdapter = DS.ActiveModelAdapter.extend();

  App.ApplicationAdapter = DS.FixtureAdapter.extend();
</script>
```

Fixtures

The rug

```
<script>
// config/server.js

module.exports = {
  drawRoutes: function(app) {
    app.get('/api/projects', function(req, res) {
      res.json([
        { id: 1, name: 'Blah', created_at: new Date() },
        { id: 2, name: 'Blah Again', created_at: new Date() },
      ]);
    }
  }
}

</script>
```

API Stubbing

The rug

```
require 'spec_helper'

describe Api::ProjectsController do
  describe "GET 'index'" do
    context 'with no authentication' do
      it 'should return a 401' do
        get :index, format: :json
        response.status.should == 401
      end
    end
  end
end
```

API Building

The rug

```
class Api::ProjectsController < Api::ApiController
  respond_to :json

  def index
    render nothing: true
  end
end
```

API Building

The rug

```
class Api::ProjectsController < Api::ApiController
  respond_to :json

  def index
    authenticate!
    render nothing: true
  end
end
```

API Building

The rug

```
require 'spec_helper'

describe Api::ProjectsController do
  describe "GET 'index'" do
    context 'with no authentication' do
      it 'should return a 401' do
        get :index, format: :json
        response.status.should == 401
      end
    end

    context 'with valid credentials' do
      let(:user) { FactoryGirl.create(:user) }

      it 'should return a 200' do
        get :index, format: :json, 'HTTP_AUTHORIZATION' => "Token
token=#{user.api_token}"
        response.status.should == 200
      end
    end
  end
end
```

API Building

The rug

```
require 'spec_helper'

describe Api::ProjectsController do
  describe "GET 'index'" do
    context 'with no authentication' do
      it 'should return a 401' do
        get :index, format: :json
        response.status.should == 401
      end
    end

    context 'with valid credentials' do
      let(:user) { FactoryGirl.create(:user) }

      it 'should return a 200' do
        get :index, format: :json, 'HTTP_AUTHORIZATION' => "Token token=#{user.api_token}"
        response.status.should == 200
      end

      it 'returns all the current goals for the current user for that week' do
        projects = [Project.new(name: 'My first project')]
        Project.stubs(:where).returns projects
        get :index, format: :json, 'HTTP_AUTHORIZATION' => "Token token=#{user.api_token}"
        json = JSON.parse(response.body)['projects']
        json.length.should == 1
        json[0]['name'].should == 'My first project'
      end
    end
  end
end
```

API Building

The rug

```
class Api::ProjectsController < Api::ApiController
  respond_to :json

  def index
    authenticate!
    @projects = Project.where(owner_id: current_user.id)
    respond_with @projects, each_serializer: ProjectSerializer
  end
end
```

API Building

```
$ lineman build
```

Building

The rug

```
Running "common" task

Running "coffee:compile" (coffee) task
File generated/js/app.coffee.js created.
File generated/js/spec.coffee.js created.
>> Destination (generated/js/spec-helpers.coffee.js) not written because compiled files were empty.

Running "less:compile" (less) task
>> Destination not written because no source files were found.
File generated/css/app.less.css created.

Running "jshint:files" (jshint) task

Running "handlebars:compile" (handlebars) task
>> Destination not written because compiled files were empty.

Running "jst:compile" (jst) task
File "generated/template/underscore.js" created.

Running "concat:js" (concat) task
File "generated/js/app-78512a8e6fcb4e70b8e2694693651800.js" created.

Running "concat:spec" (concat) task
File "generated/js/spec.js" created.

Running "concat:css" (concat) task
File "generated/css/app-f7a1da849888d2c913a5374c7dc0cbbe.css" created.

Running "images:dev" (images) task
Copying images to 'generated/img'

Running "webfonts:dev" (webfonts) task
Copying webfonts to 'generated/webfonts'

Running "pages:dev" (pages) task
generated/index.html generated from app/pages/index.us

Running "dist" task

Running "uglify:js" (uglify) task
File "dist/js/app-78512a8e6fcb4e70b8e2694693651800.js" created.

Running "cssmin:compress" (cssmin) task
File dist/css/app-f7a1da849888d2c913a5374c7dc0cbbe.css created.

Running "images:dist" (images) task
Copying images to 'dist/img'

Running "webfonts:dist" (webfonts) task
Copying webfonts to 'dist/webfonts'

Running "pages:dist" (pages) task
dist/index.html generated from app/pages/index.us

Done, without errors.
```

Building

```
$ tree dist

dist
├── css
│   └── app.css
├── favicon.ico
├── index.html
└── js
    └── app.js
```

The rug

```
$ heroku config:add BUILDPACK_URL=https://github.com/ddollar/heroku-buildpack-multi.git
```

The rug

```
$ cat .buildpacks
https://github.com/heroku/heroku-buildpack-ruby
https://github.com/testdouble/heroku-buildpack-lineman
```

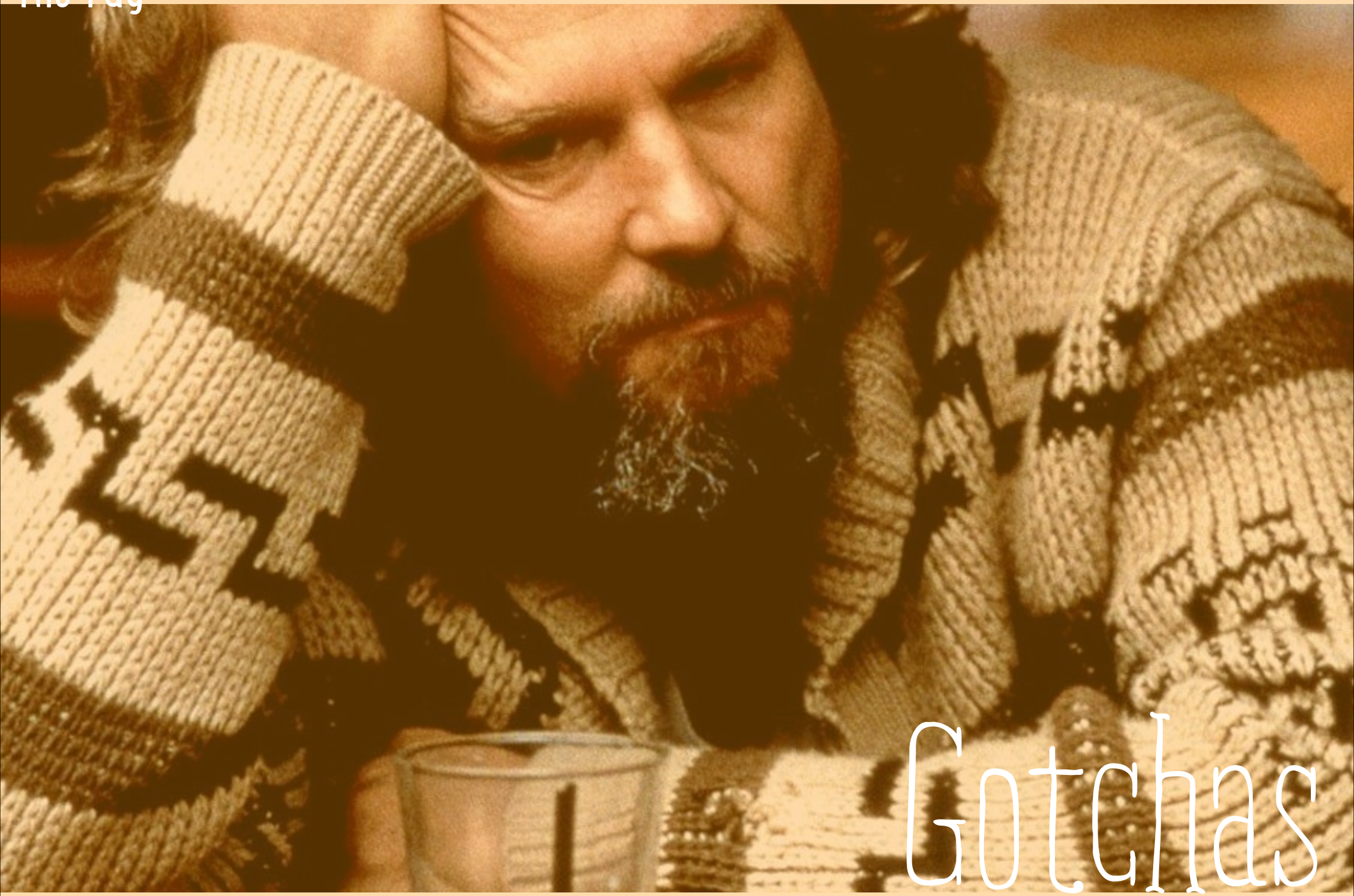
Deployment

The rug

```
$ git push heroku master
```

Deployment

The rug



Gotchas

The rug

Going against the grain

The rug

Handlebars

The rug

Documentation

The rug

Thanks, Jim

Questions?

Why Ember and not*?