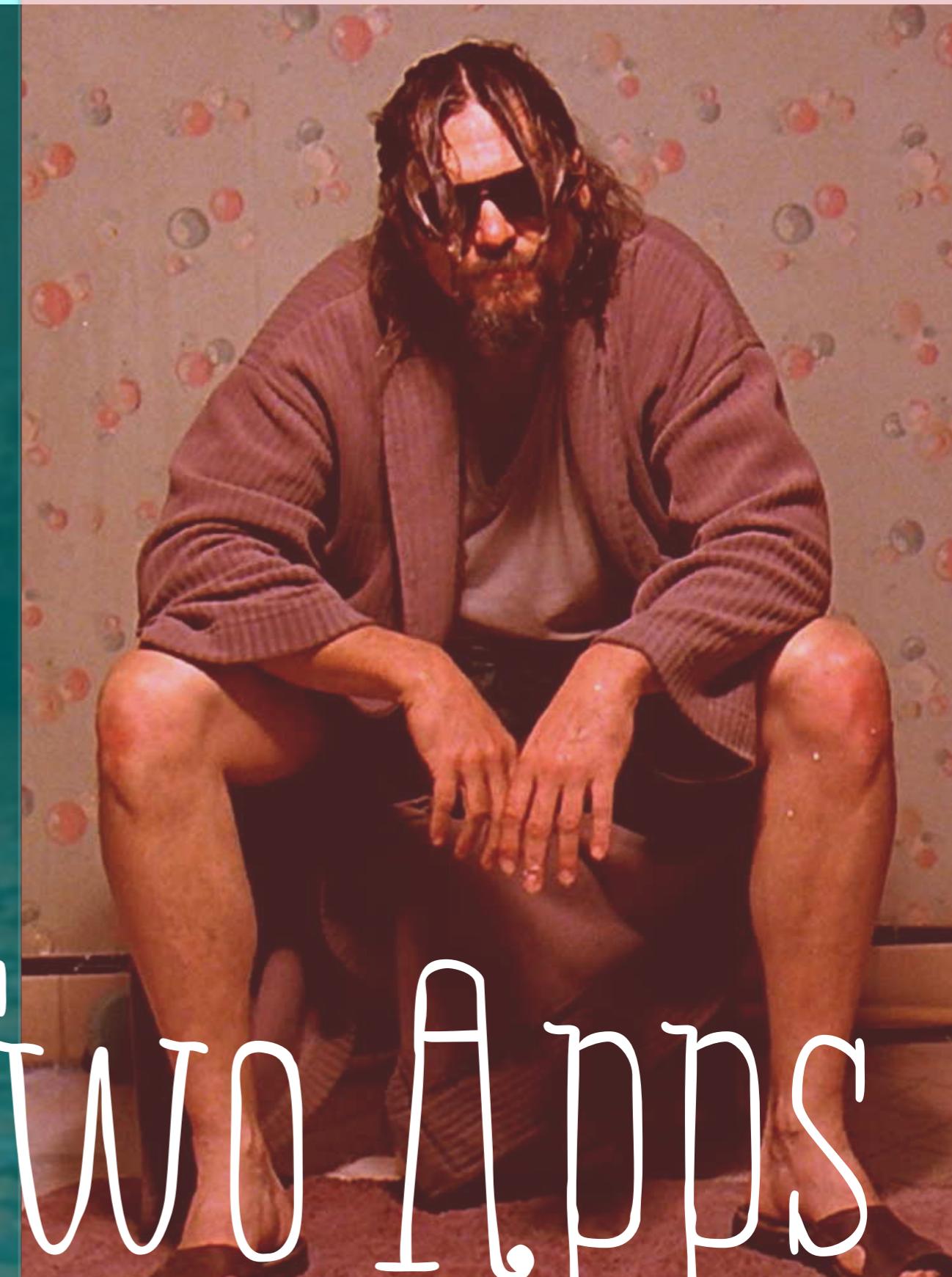


How Ember Changed the Way I Build Web Applications

a Tale of Two Apps





More than just a
talk on Ember

Today, we are
going to
T A L K
about **3** things

A rich



Man

ay, we are
going to
ALK
ut 3 things

A rich



Man

A laid back



Dude

e are

g to

K

lilings

A rich



A laid back



Man

Dude

A rug

Jamie Wright

@jwright

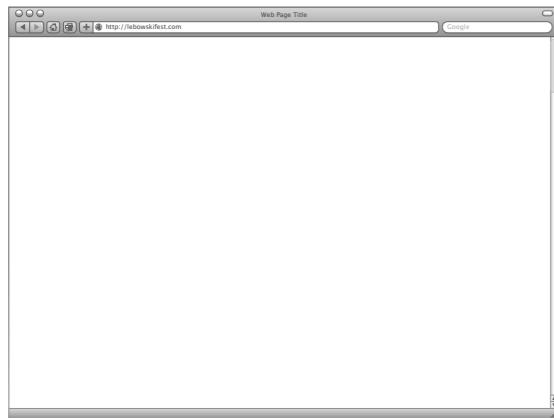


App #1

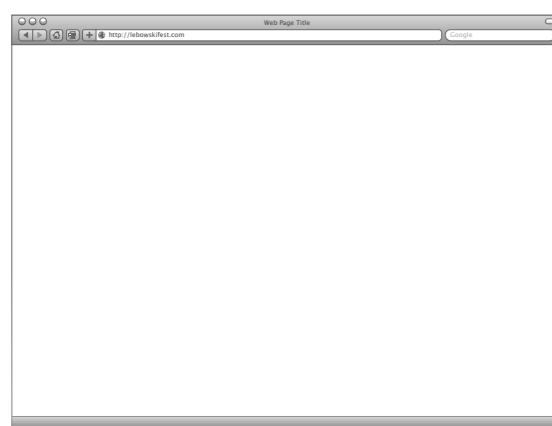


Server Side Rendering

App #1



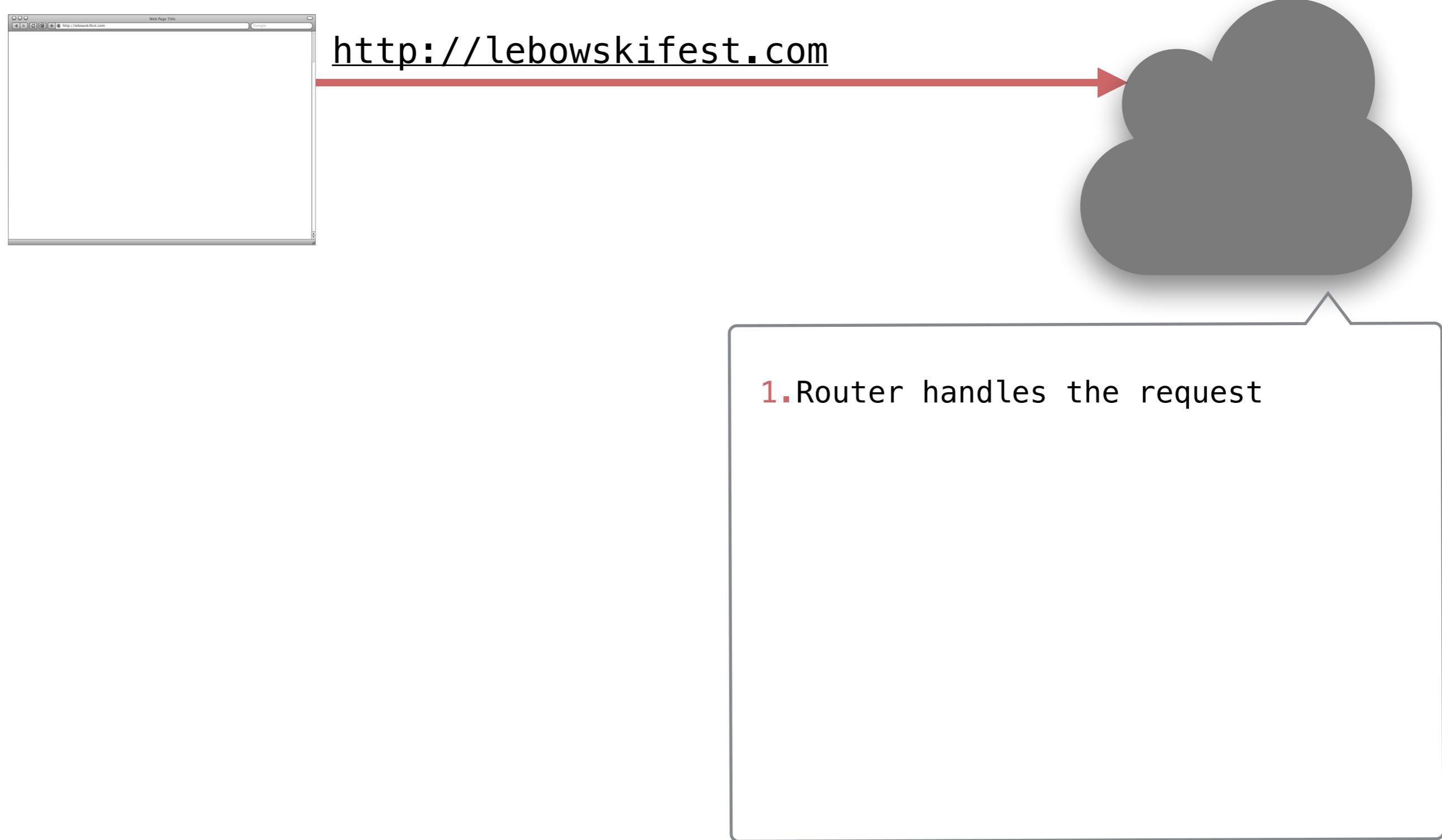
App #1



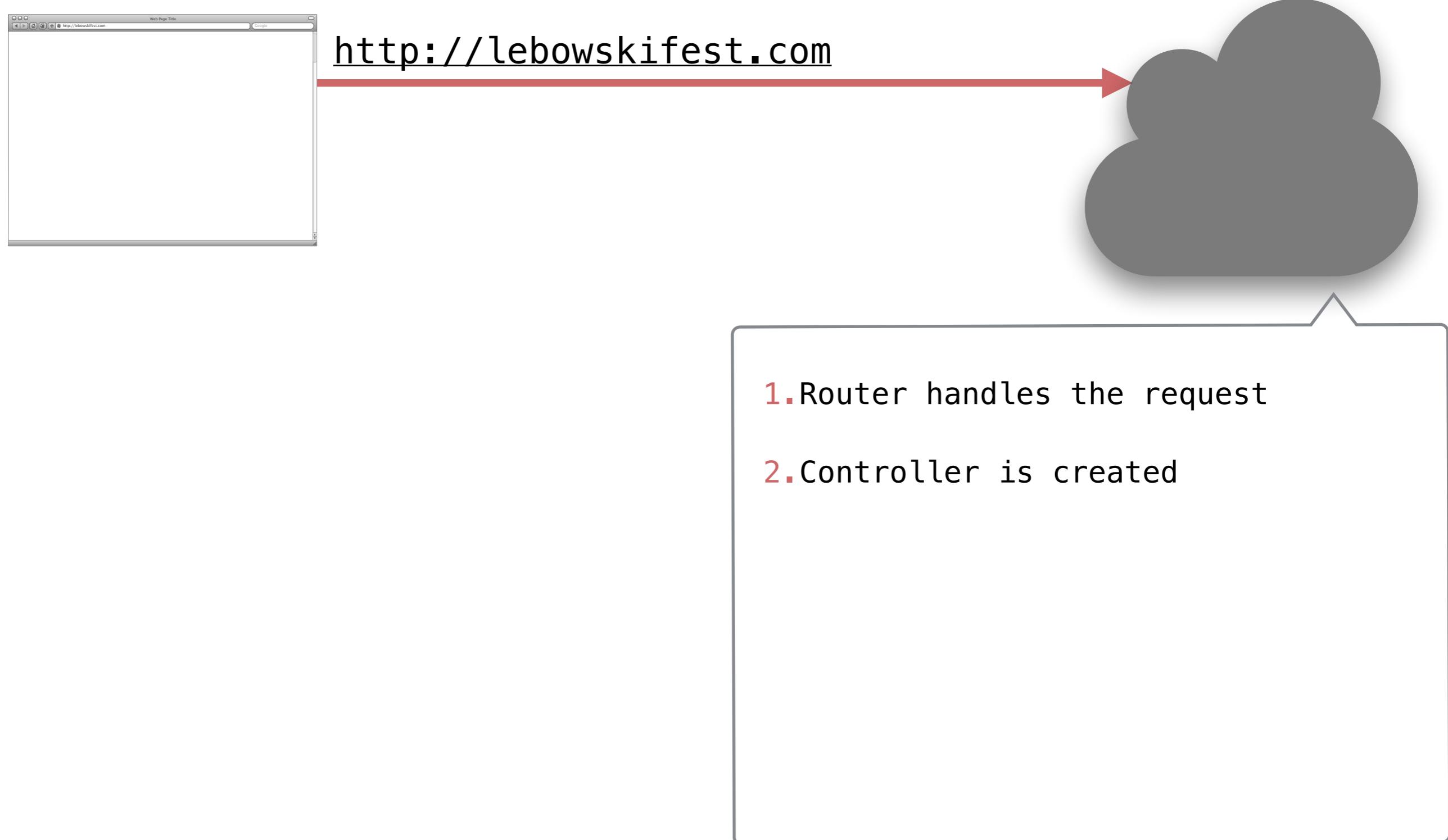
http://lebowskifest.com



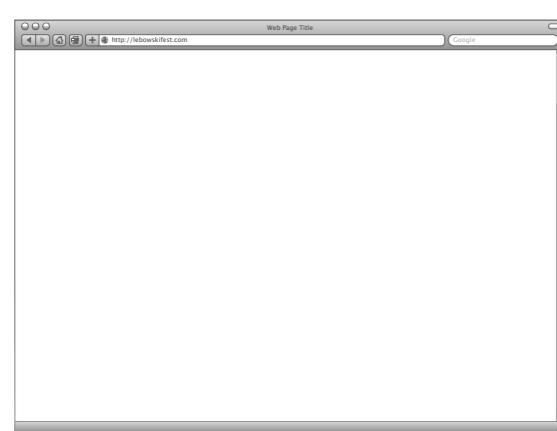
App #1



App #1



App #1

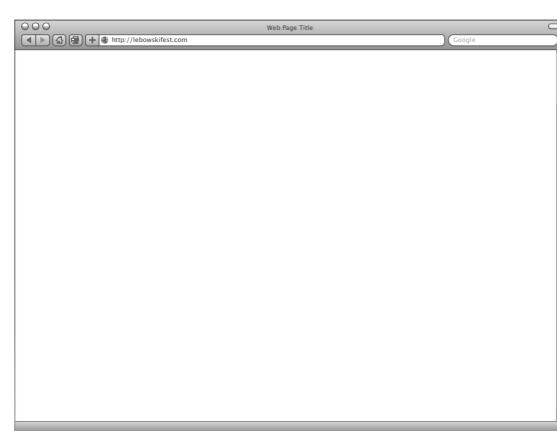


http://lebowskifest.com



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

App #1

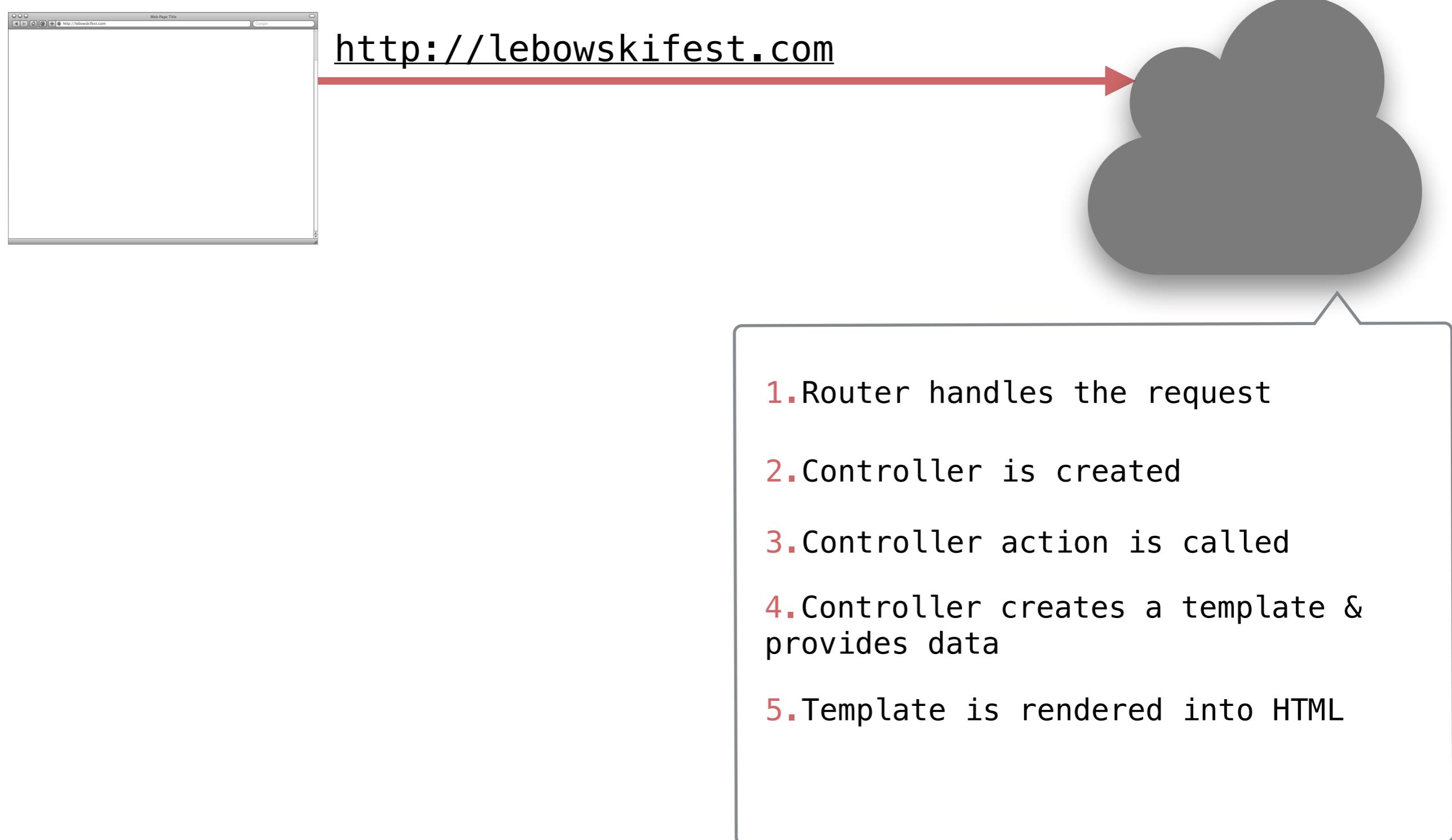


http://lebowskifest.com

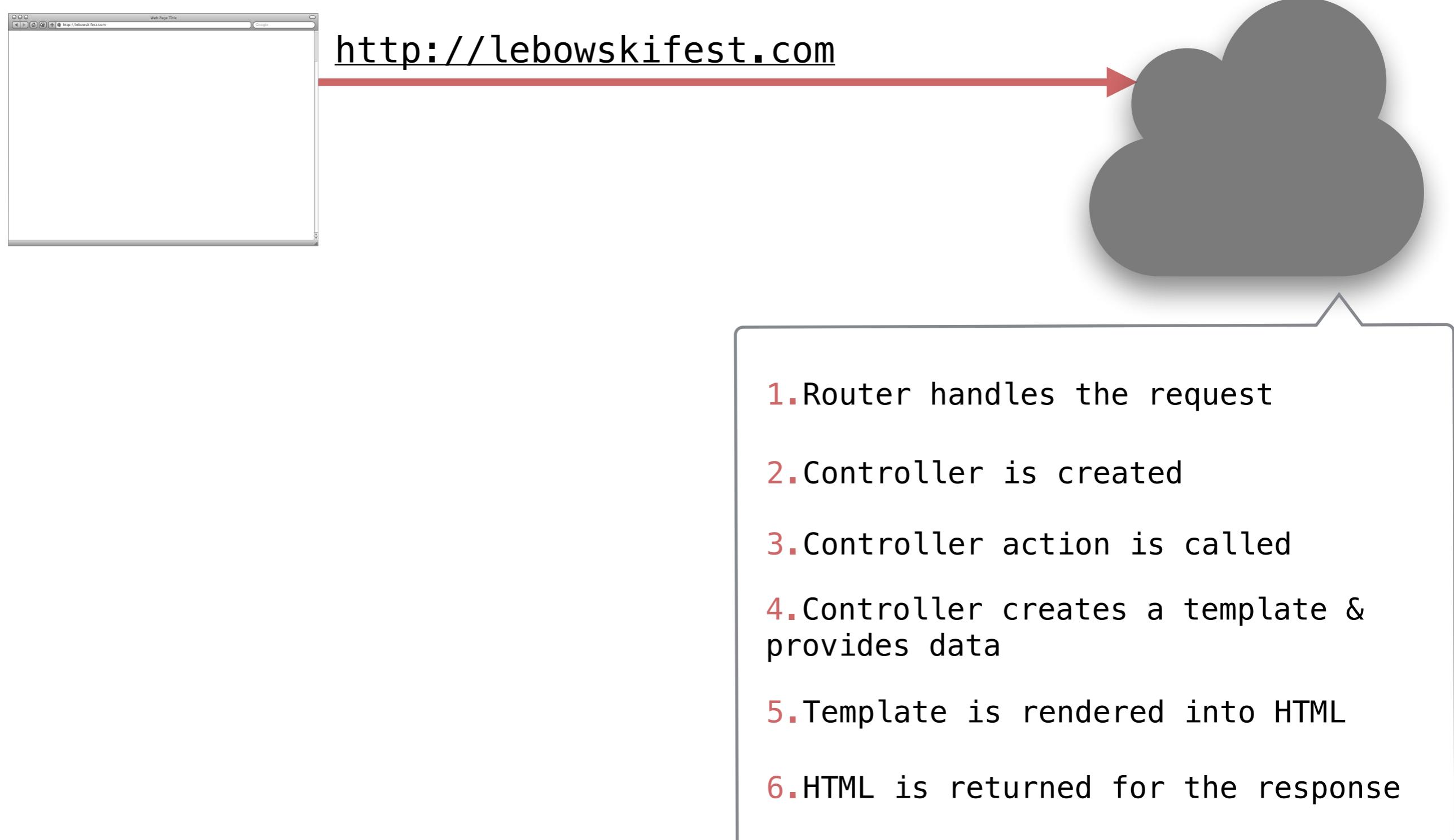


1. Router handles the request
2. Controller is created
3. Controller action is called
4. Controller creates a template & provides data

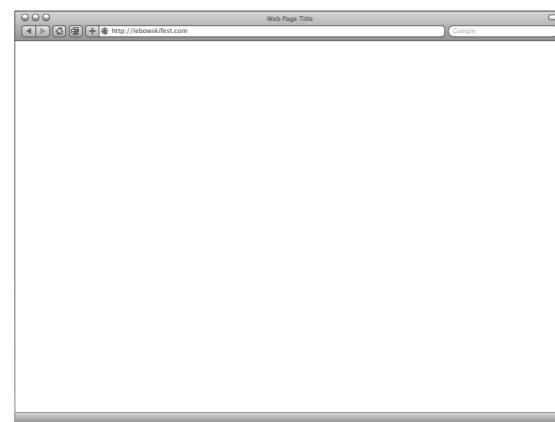
App #1



App #1



App #1



http://lebowskifest.com

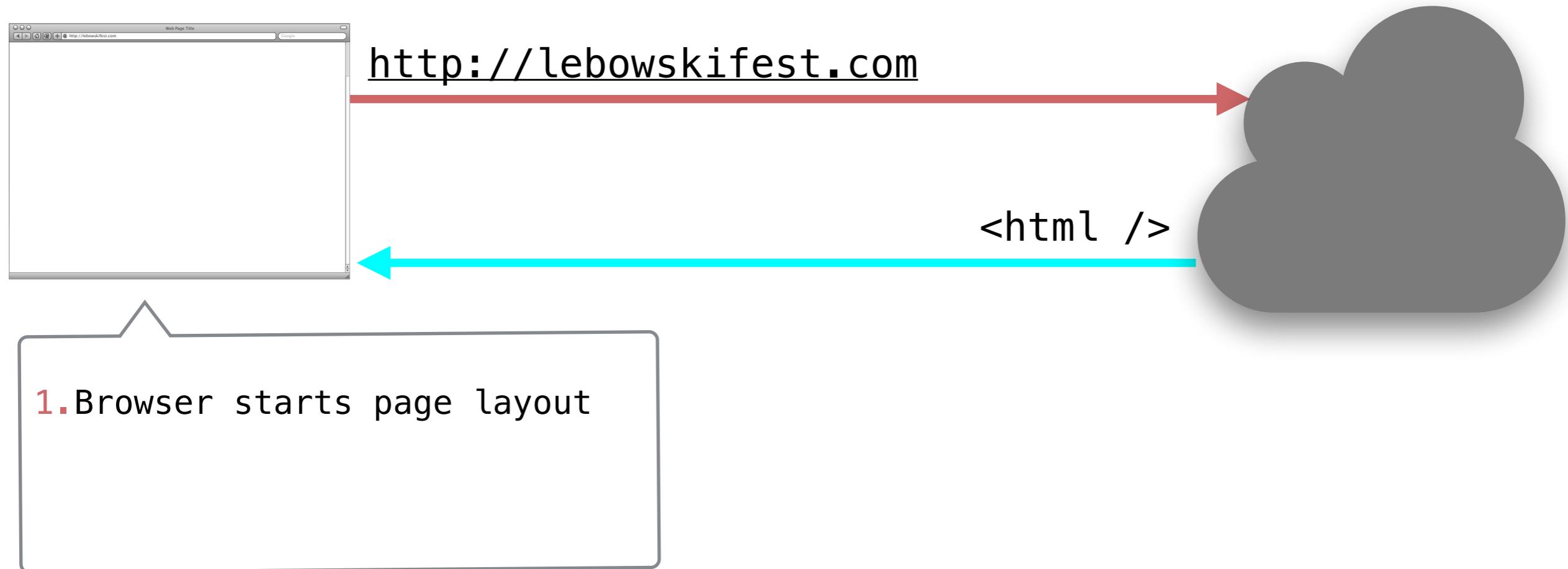


1. Router handles the request
2. Controller is created
3. Controller action is called
4. Controller creates a template & provides data
5. Template is rendered into HTML
6. HTML is returned for the response

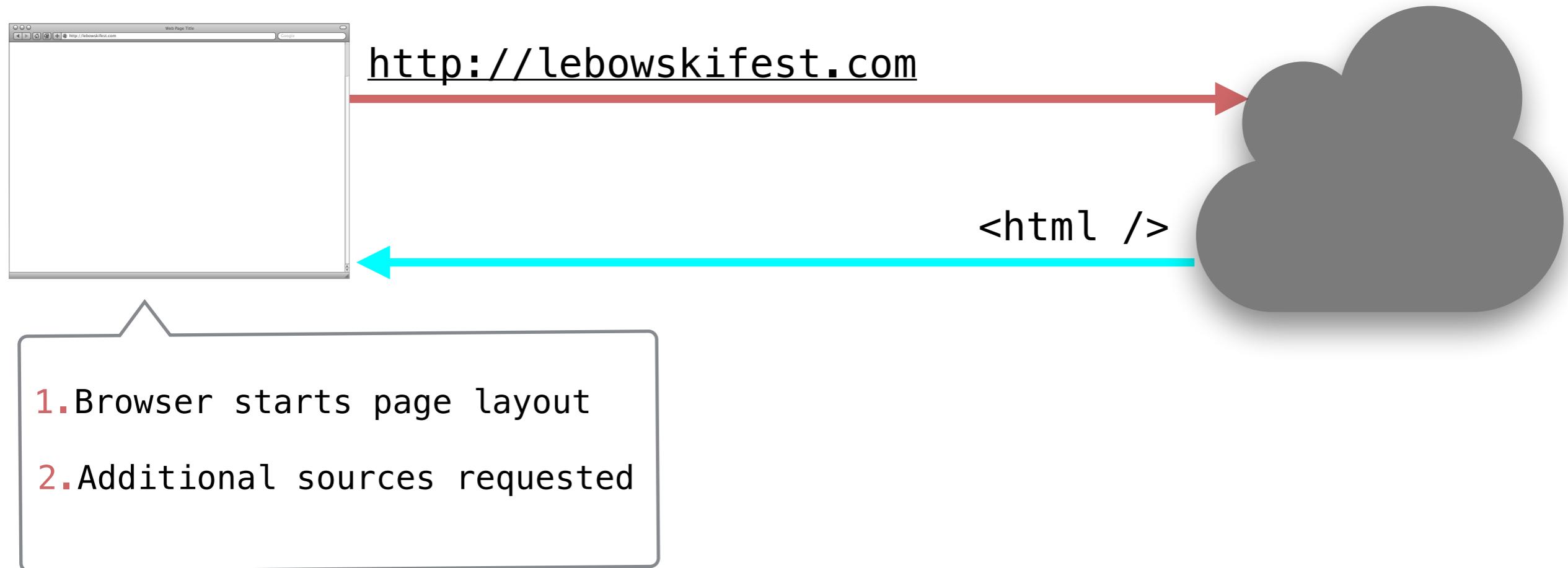
App #1



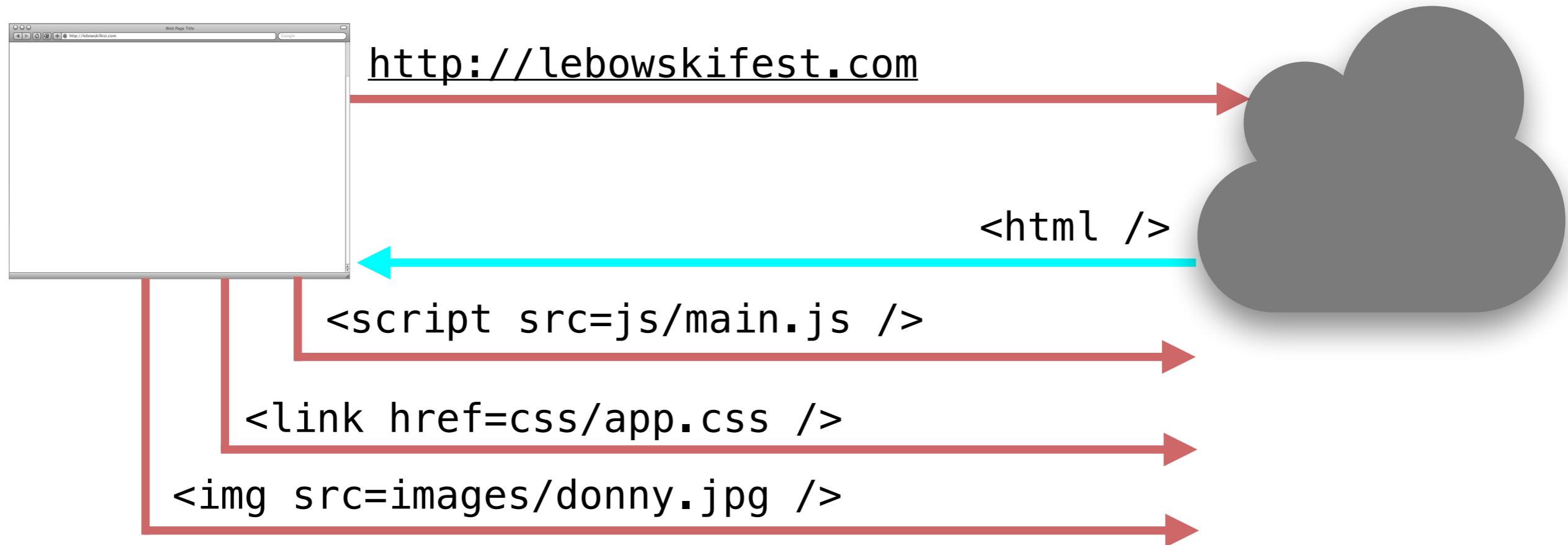
App #1



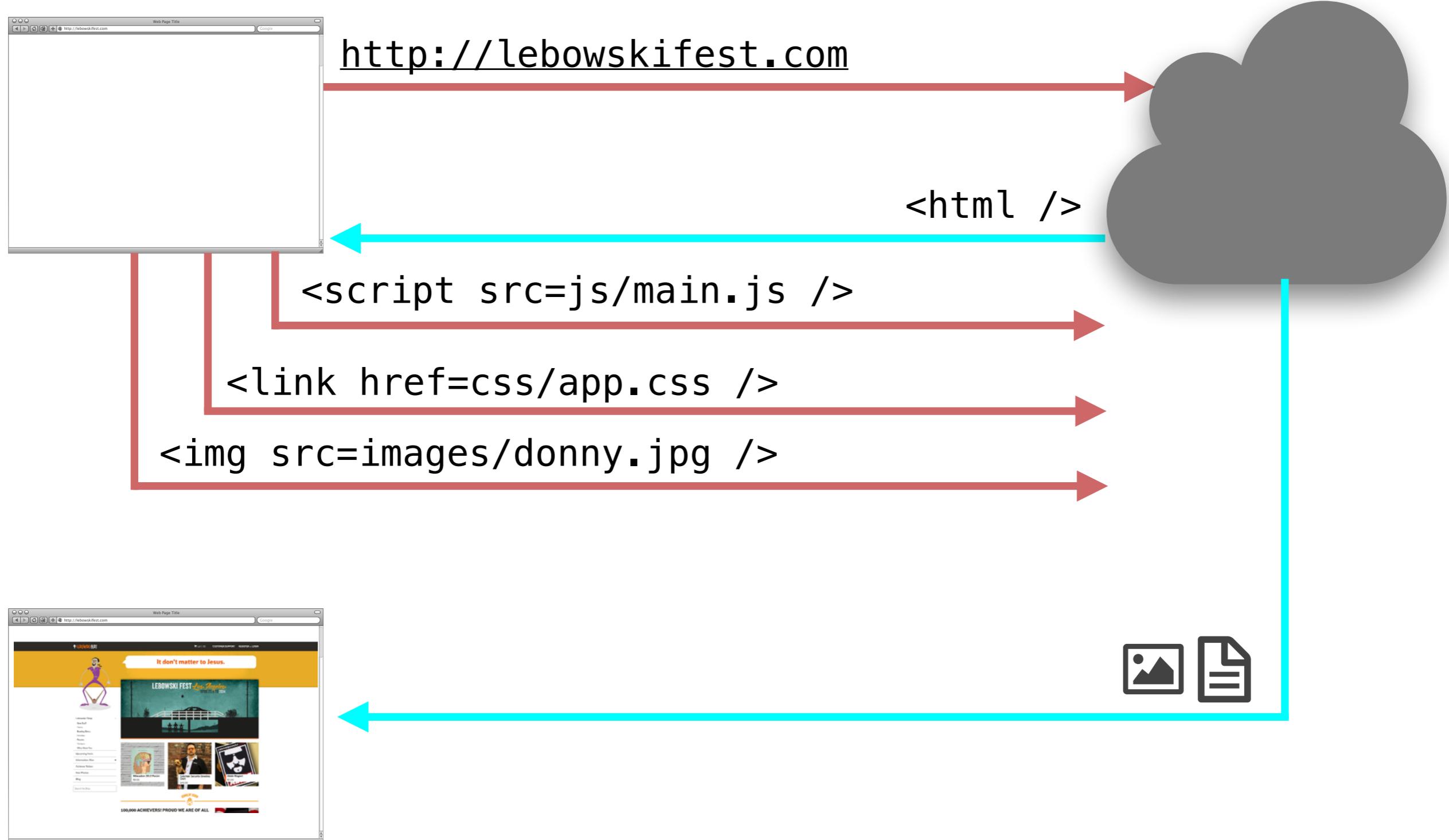
App #1



App #1



App #1



App #1

A photograph of a man with light brown hair and a beard, wearing a black hoodie, looking directly at the camera with a slight smile. He is positioned in the foreground, slightly off-center. In the background, there are several other people, some of whom are wearing white face paint or masks, though they are out of focus.

The Good

App #1



You know the environment

App #1

A photograph of three men looking upwards and slightly to the right. The man in the center has a mustache and is wearing a brown jacket over a light shirt. The man on the left is wearing a dark shirt. The man on the right is wearing a dark t-shirt. They appear to be in an indoor setting with a white ceiling and fluorescent lights.

Caching

App #1



Google can crawl

App #1

The Bad



App #1



Bad UX in most cases

Ticket Information

Jamie

Wright

123 Main St.

Address Line 2

Beverly Hills

California ▾

90210

4195551212

Billing Information

Credit Card Number

1

2014

CVV

My billing address is the same as my ticket information address.

Note: Your billing address must match the address on file for your credit card.

First Name

Last Name

Order Summary

Chicago Grand Tour with
Hancock Tower

Friday the 13th of June, 2014 at 9:30
am

1 Adult Rate Ticket at \$59.95 ea. \$59.95

Activity Total \$59.95

Processing Fee \$6.95

Total \$66.90

PLACE YOUR ORDER

* There is a \$6.95 processing fee per order. This is a flat fee regardless of the number of tickets or tours purchased on an order.

* We only honor promotional codes issued by our company.

Ticket Information

Jamie

Wright

123 Main St.

Address Line 2

Beverly Hills

California ▾

90210

4195551212

Billing Information

Credit Card Number

1

2014

CVV

My billing address is the same as my ticket information address.

Note: Your billing address must match the address on file for your credit card.

First Name

Last Name

Order Summary

Chicago Grand Tour with
Hancock Tower

Friday the 13th of June, 2014 at 9:30
am

1 Adult Rate Ticket at \$59.95 ea. \$59.95

Activity Total \$59.95

Processing Fee \$6.95

Total \$66.90

PLACE YOUR ORDER

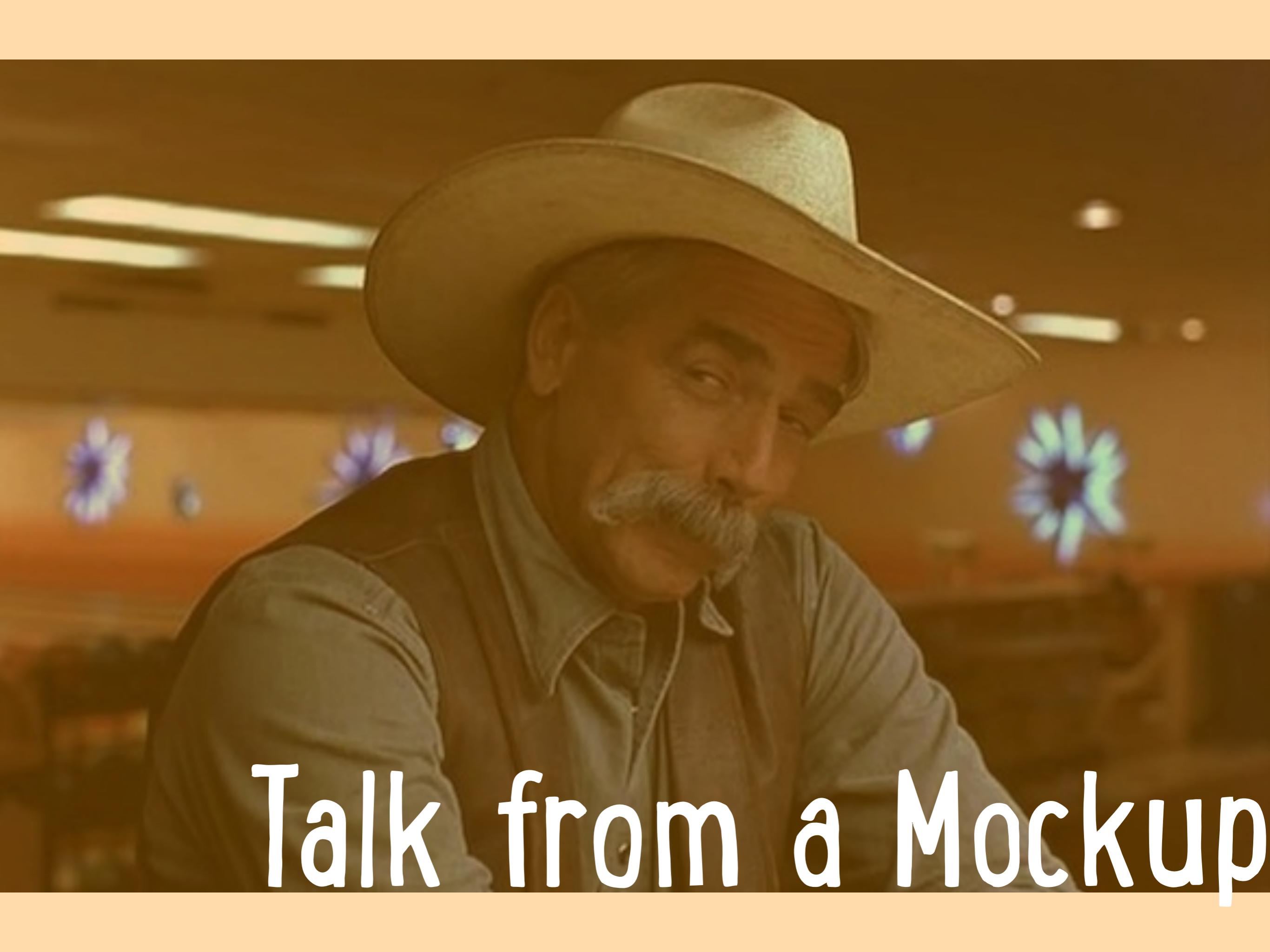
* There is a \$6.95 processing fee per order. This is a flat fee regardless of the number of tickets or tours purchased on an order.

* We only honor promotional codes issued by our company.

App #1



Incomplete prototyping

A close-up photograph of a man with a full, grey beard and mustache, wearing a light-colored fedora hat. He is looking slightly to his right with a thoughtful expression. In the background, there are several small, colorful flowers (purple and green) growing in a field under a clear blue sky.

Talk from a Mockup

App #1



Incomplete prototyping

App #1

A woman with long blonde hair, wearing a green bikini top and matching bottoms, is smiling and holding a bottle of perfume. She is standing in a bathroom with a white tiled floor and walls. A toilet paper holder with two rolls of toilet paper is visible in the background.

slower to polish

App #1



App #1



Boundaries

```
// app/assets/javascripts/availability.js

$( 'select.monthYearSelect' ).change( function(e) {
  var country = $(e.target).data("country-code");
  var city = $(e.target).data("city-permalink");
  var tour = $(e.target).data("tour-id");

  $.get( "/" + country + "/" + city + "/" + tour + "/tickets/
availability/" + $(e.target).val() + "/dates" );

  e.preventDefault();
}) ;
```

```
# app/controllers/availability_controller.rb

class AvailabilityController < ApplicationController
  def dates
    @dates = OfferAvailability.new(params[:id],
      Date.today,
      Date.new(year, month, -1)).dates
  end
end
```

```
// app/views/availability/dates.js.erb

$( "select.daySelect" ).empty();
$( "select.daySelect" ).html("<%= j content_tag(:option,
  \"Which Day?\", value: '', class: \"disabled\", disabled: \"disabled\",
  selected: \"selected\") +
  options_from_collection_for_select(@dates, :id, :display) %>");
$( "select.daySelect" ).removeClass("selected");

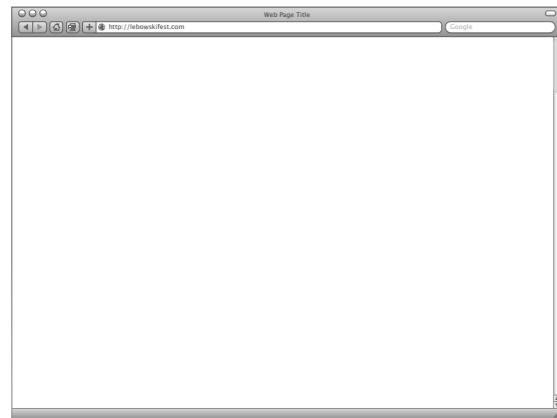
<% if @dates.size == 1 %>
  $("select.daySelect").val($(".select.daySelect option:nth-
child(2)").val()).change();
  $("select.daySelect").addClass("selected");
<% end %>
```

App #2

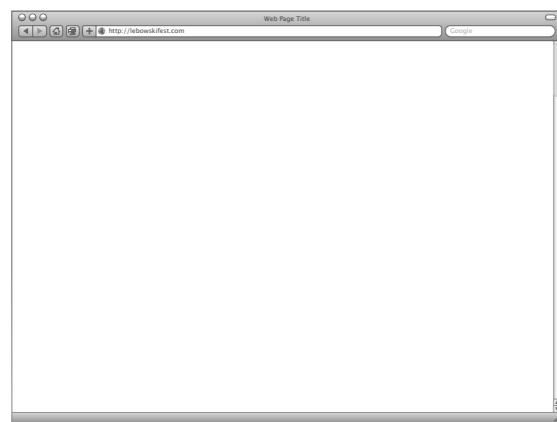


Client Side Rendering

App #2



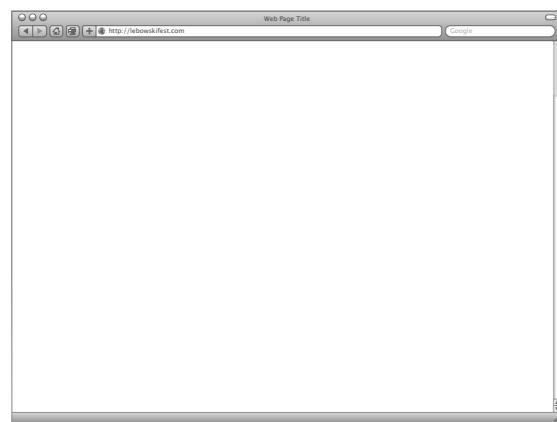
App #2



http://lebowskifest.com



App #2



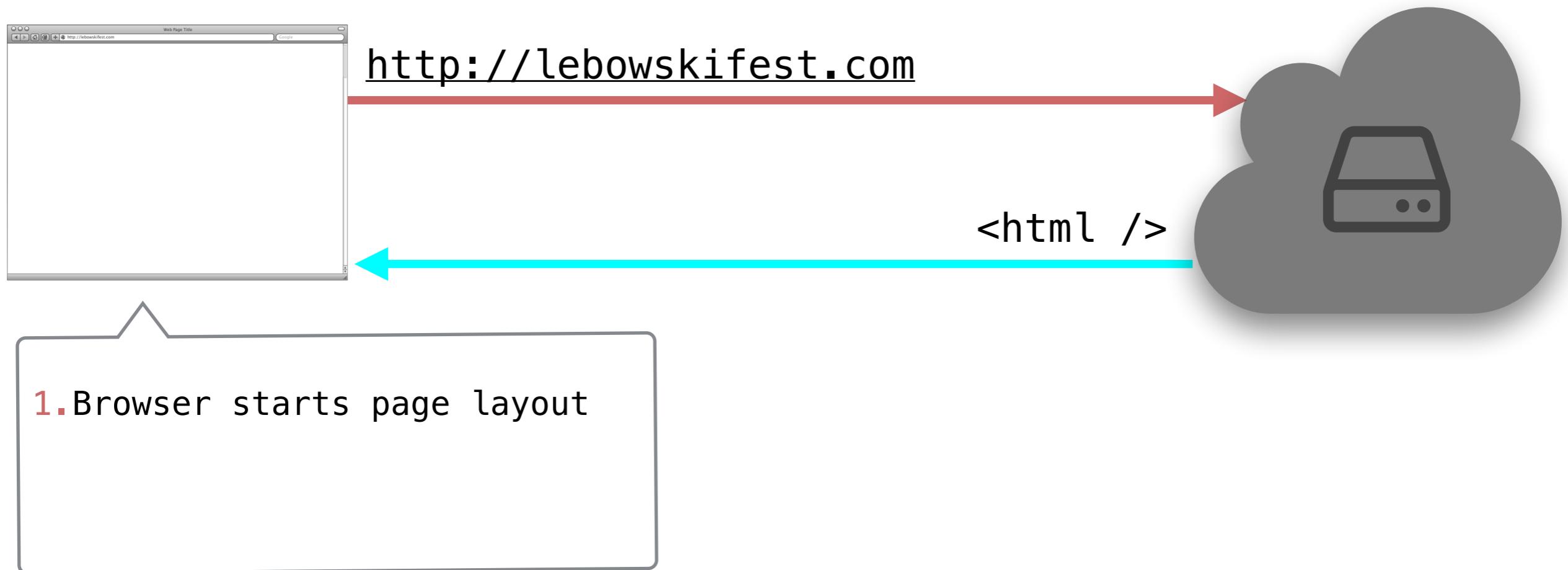
http://lebowskifest.com



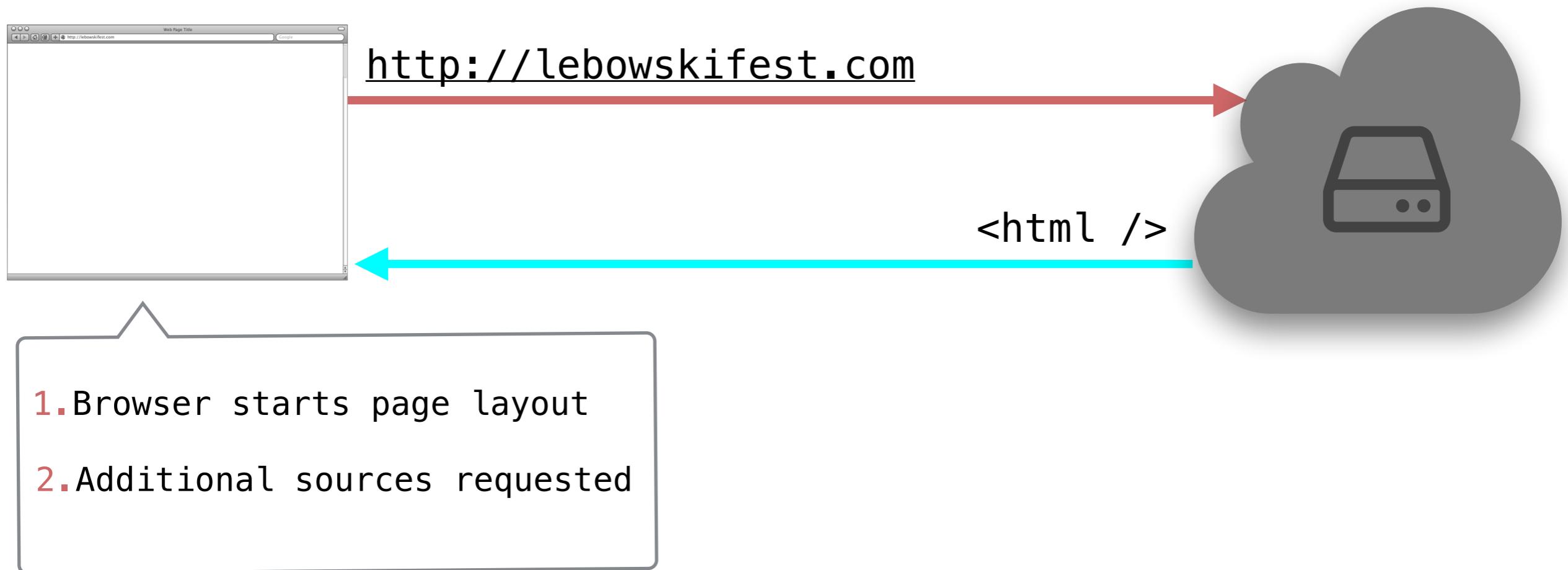
App #2



App #2



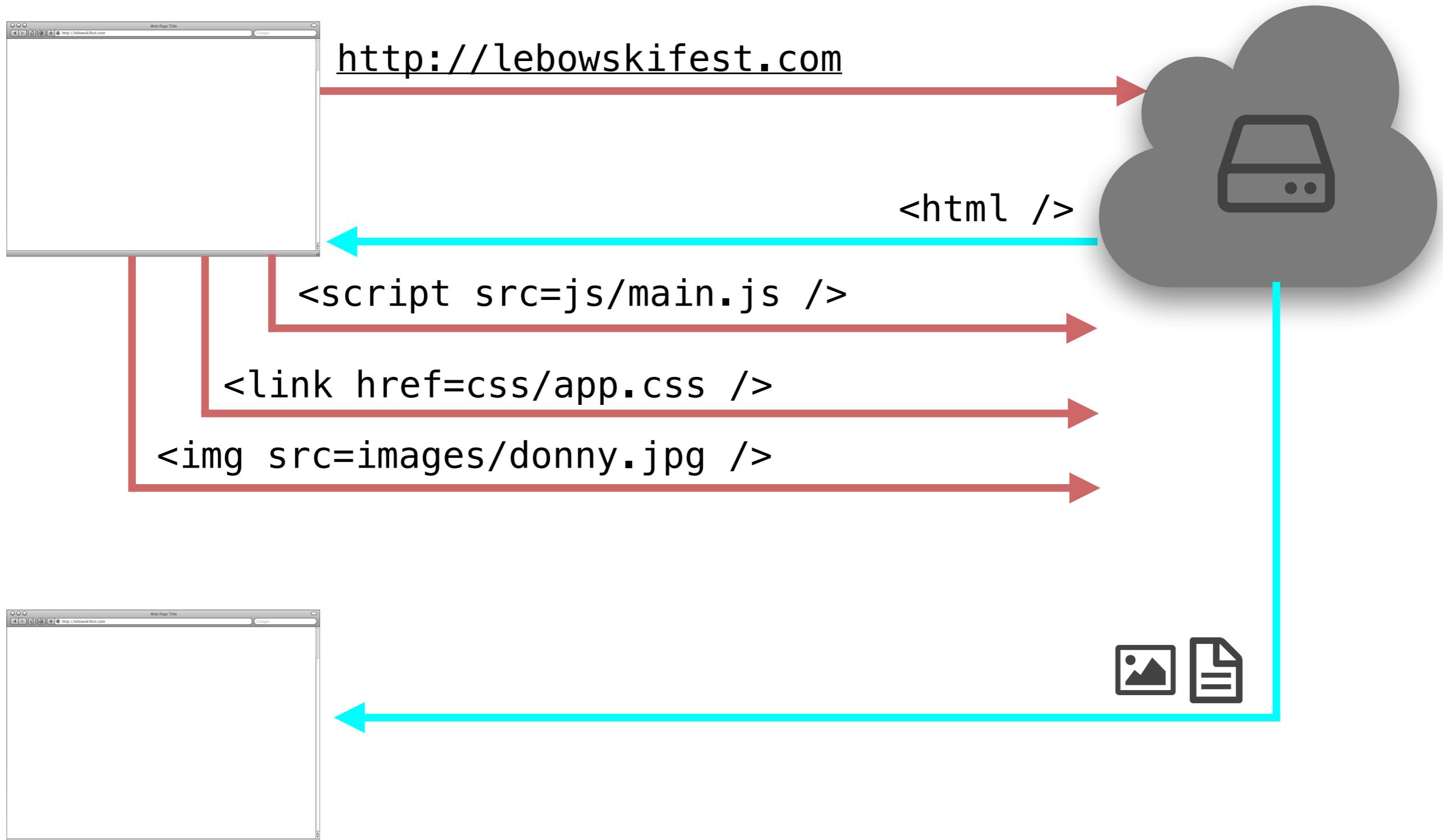
App #2



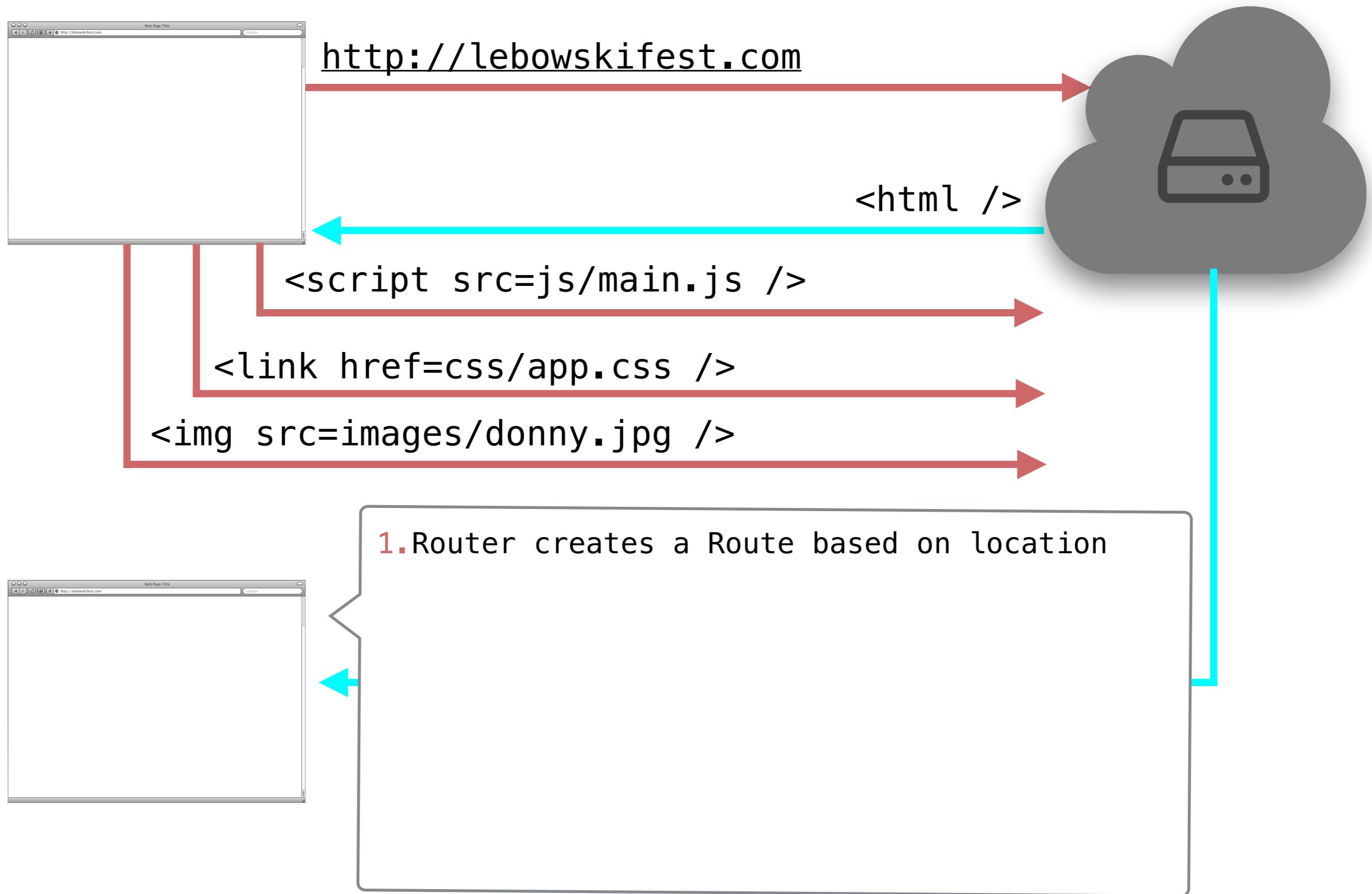
App #2



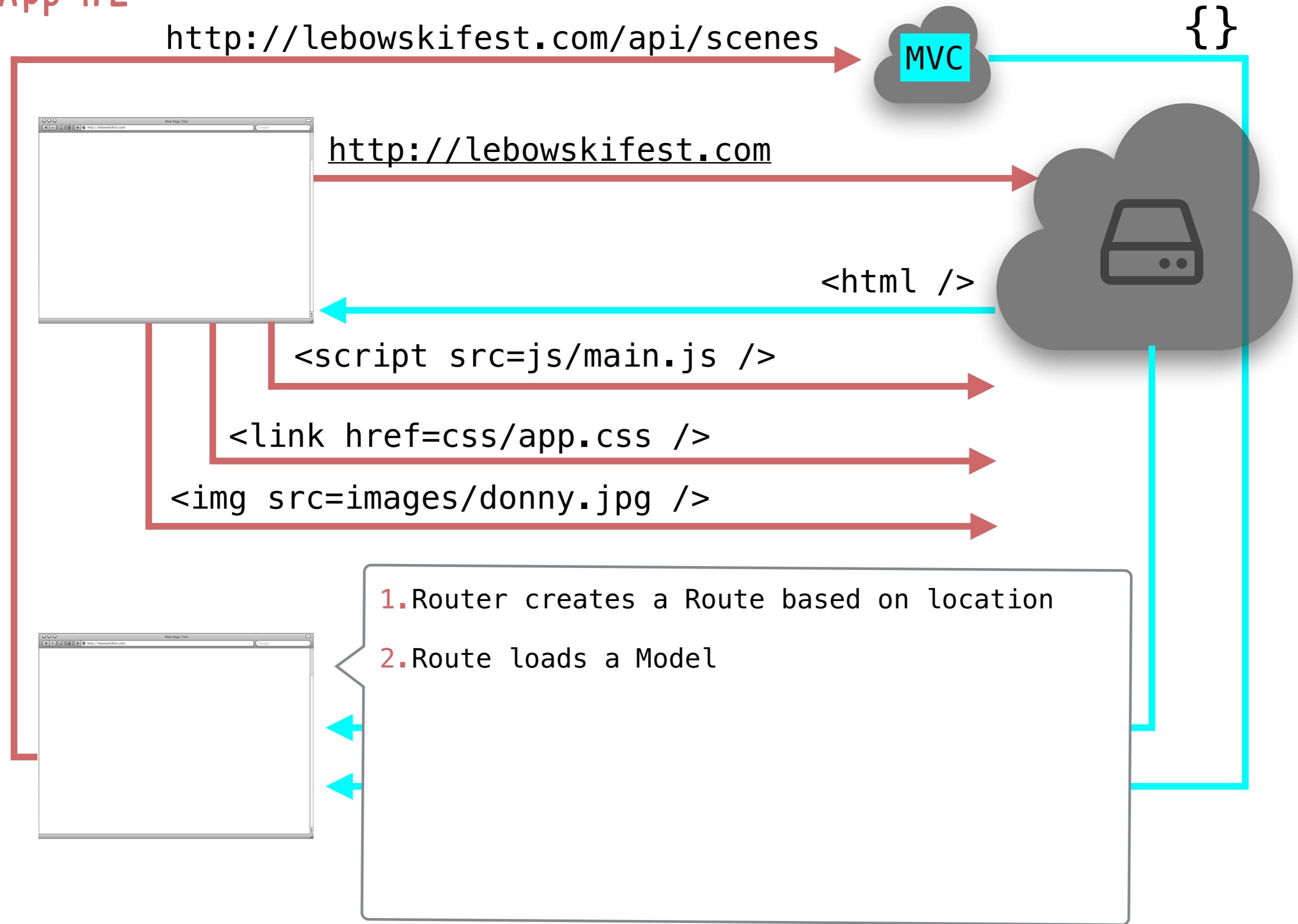
App #2



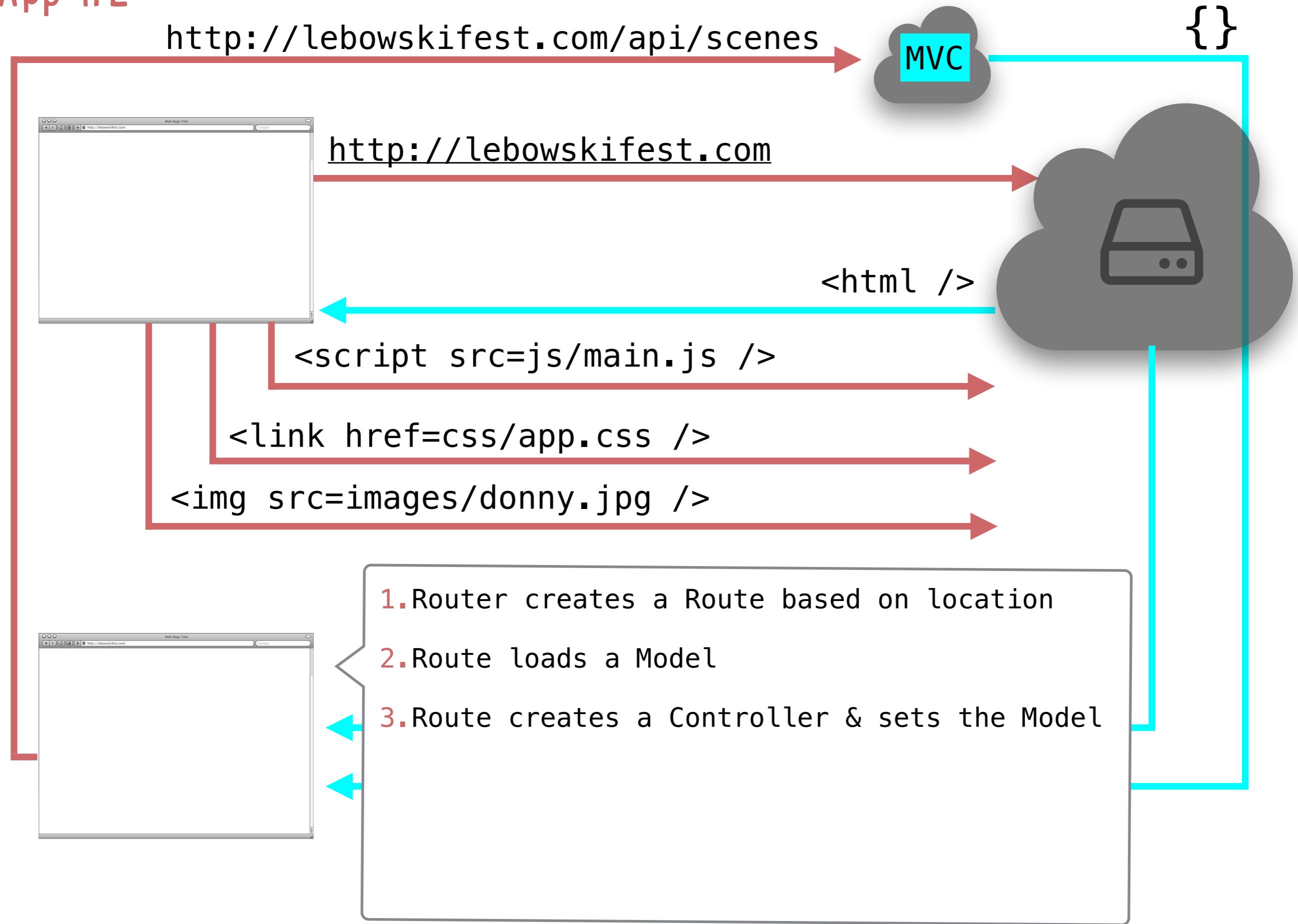
App #2



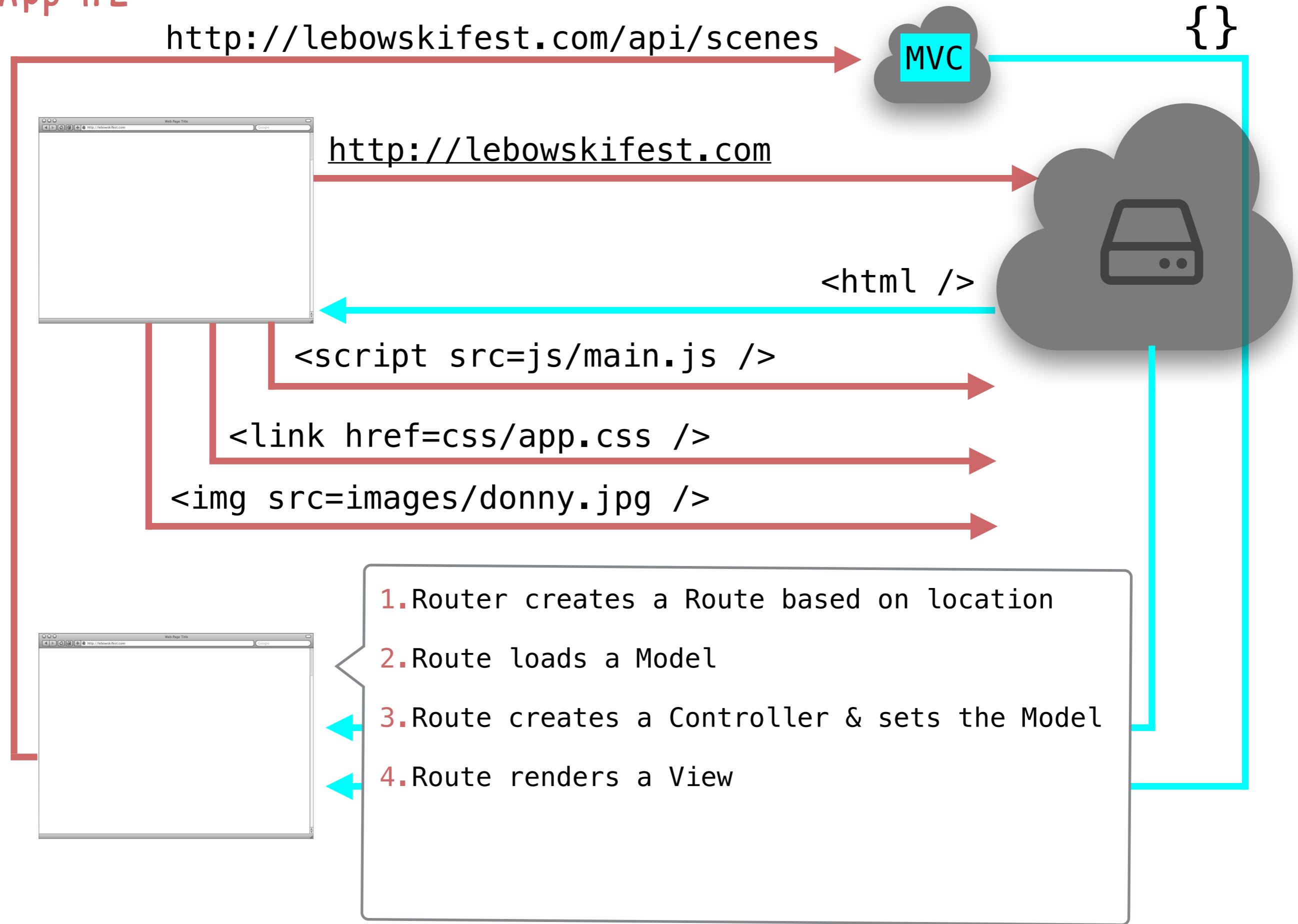
App #2



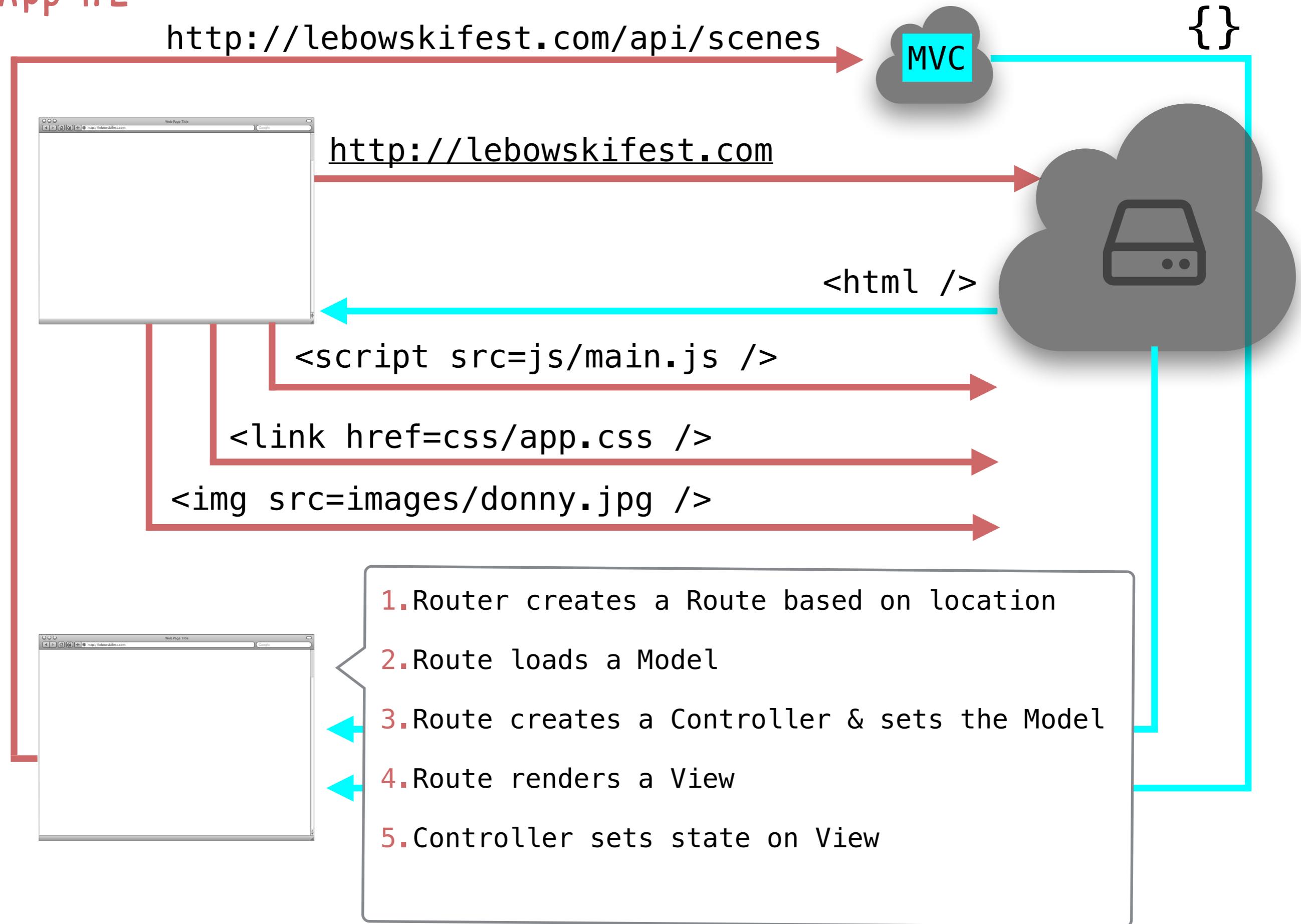
App #2



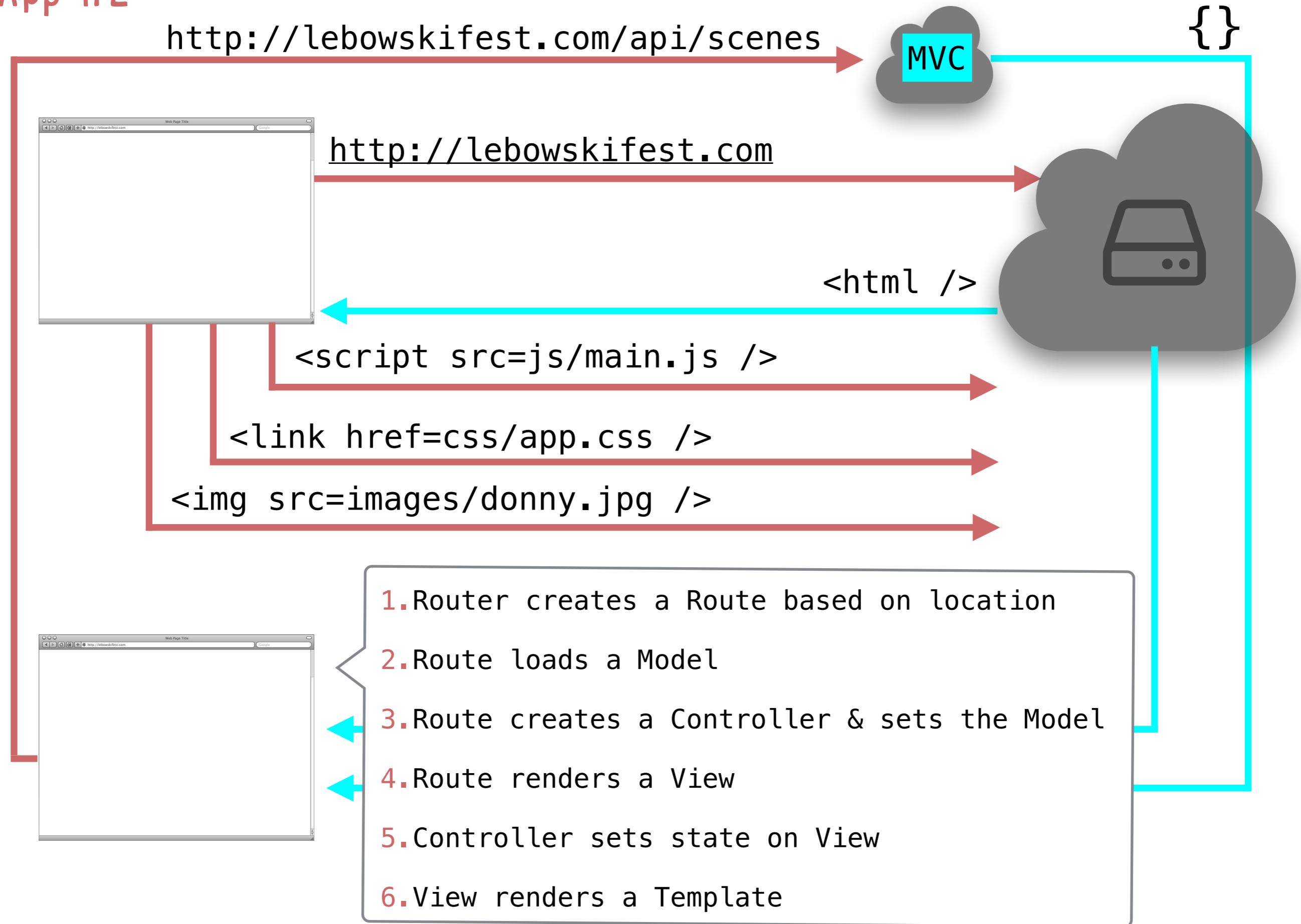
App #2



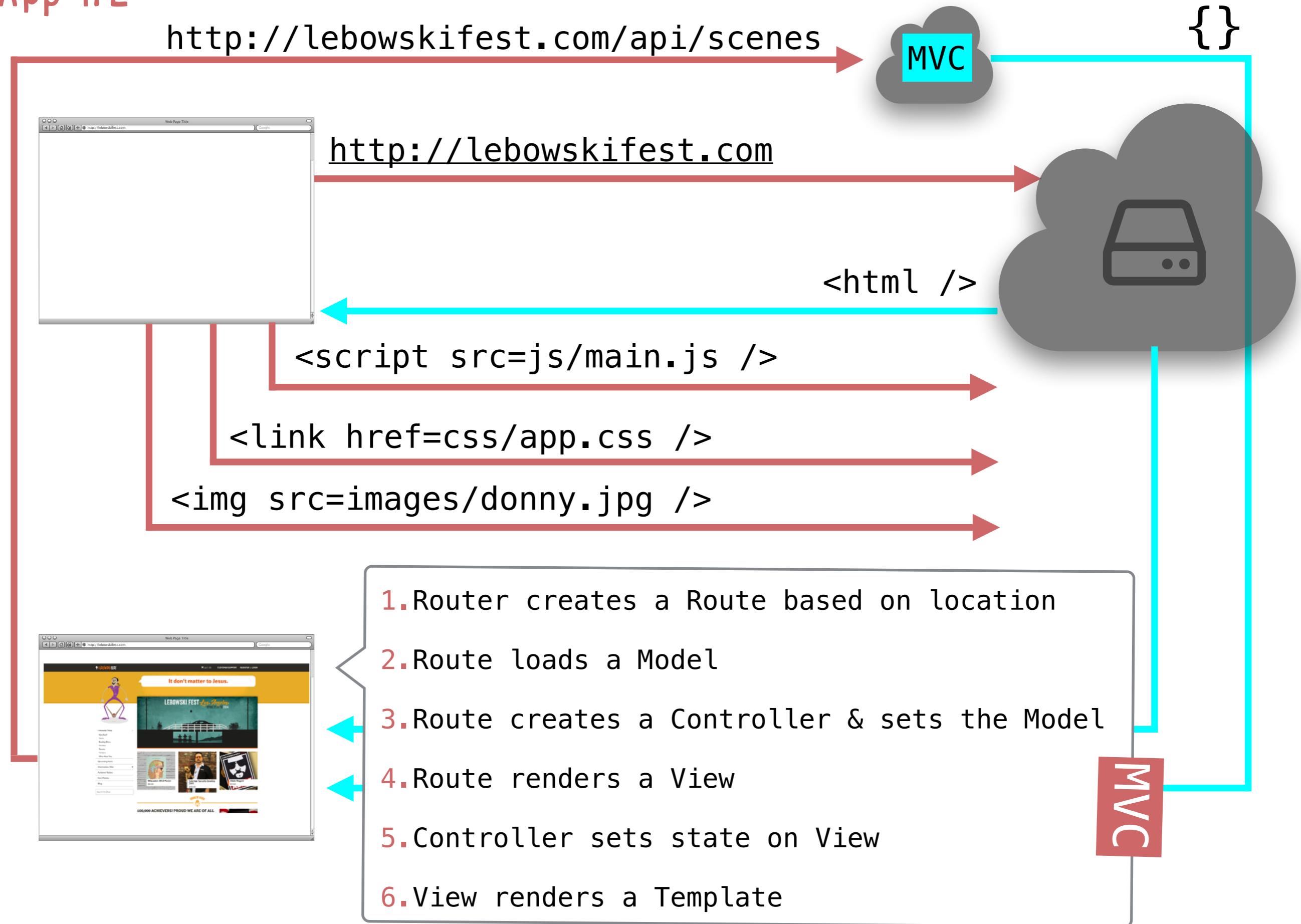
App #2



App #2



App #2





What rendering technique is faster?







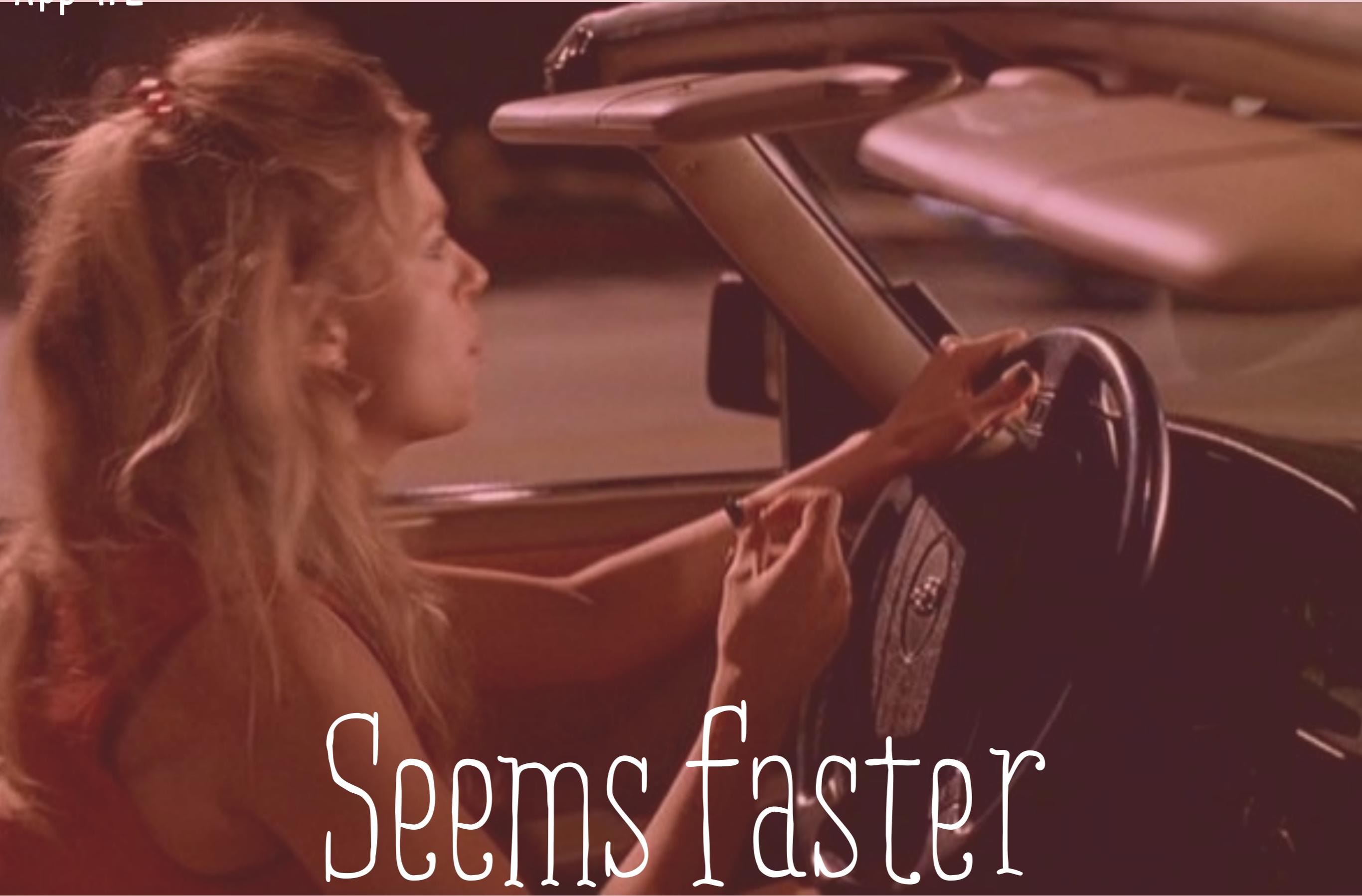
Rendering
on **THE**
server
is FASTER

App #2

A photograph of a man with long, wavy hair and a full, dark beard. He is wearing dark sunglasses and a light-colored, cable-knit sweater. He is seated in the driver's seat of a car, looking directly at the camera with a neutral expression. His right hand is raised to his chin, holding a small, dark object between his fingers. The interior of the car is visible, including the steering wheel and dashboard.

The Good

App #2



Seems faster

Rails + jQuery

Rails + jQuery

Ember.js

Ember.js

App #2



Usually leads to good UX

Add a new payment method

X

Payment Info

Name on card

Jamie | I

Jamie Wright

		01	▼	2013	▼
--	--	----	---	------	---

Billing Address

Address

Address line 1

Address line 2

Country

United States Of America

City

State

Zip

Same as shipping address

Apply

Add a new payment method

X

Payment Info

Name on card

Jamie | I

Jamie Wright

		01	▼	2013	▼
--	--	----	---	------	---

Billing Address

Address

Address line 1

Address line 2

Country

United States Of America

City

State

Zip

Same as shipping address

Apply

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

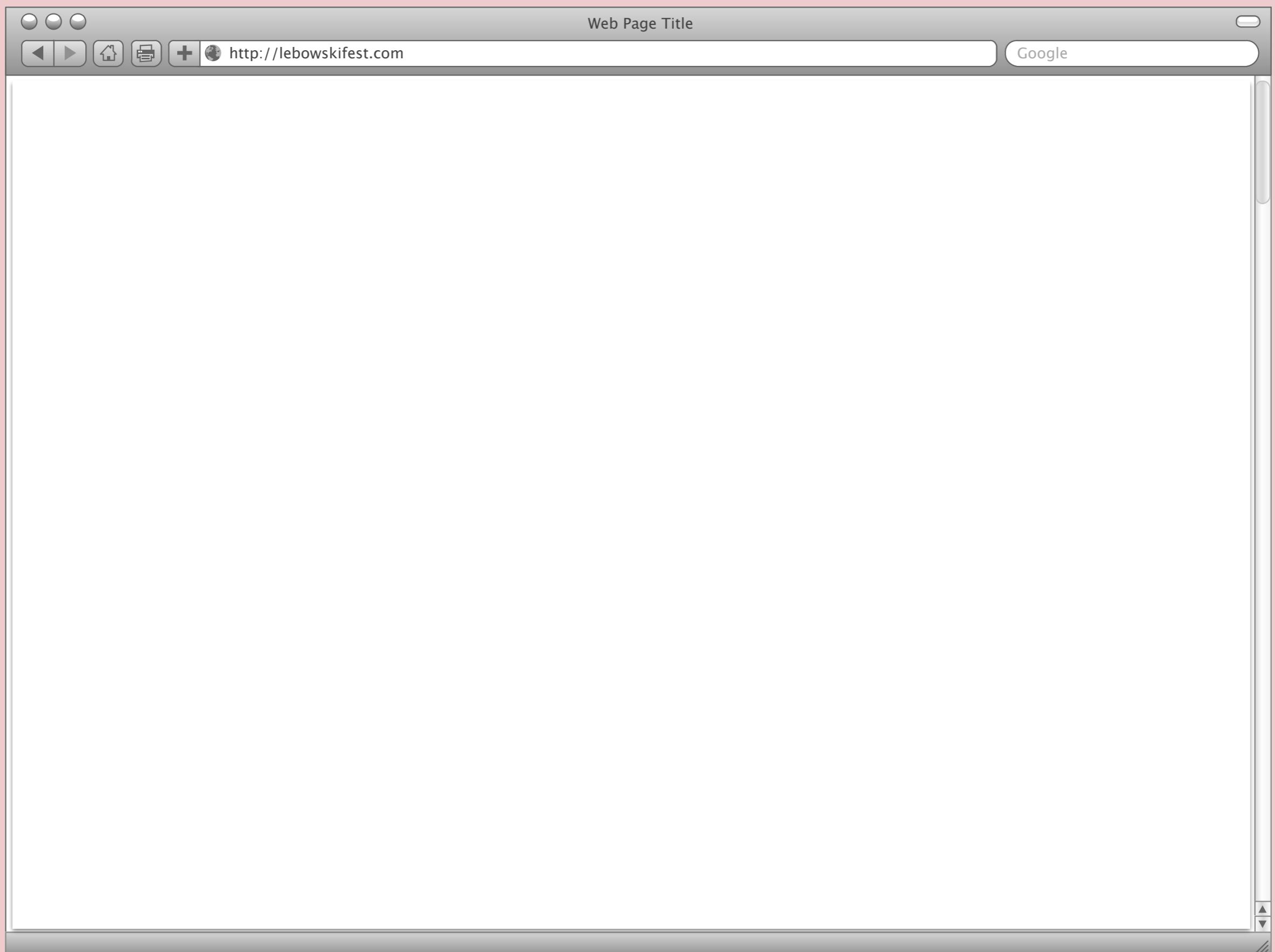
A man with grey hair and a beard, wearing sunglasses and a grey zip-up hoodie, looks upwards and to his right with a neutral expression. In the foreground, a large, out-of-focus hand holds a black marker, pointing towards a piece of paper with colorful, abstract drawings. The background is a dark room with some glowing lights.

The Bad

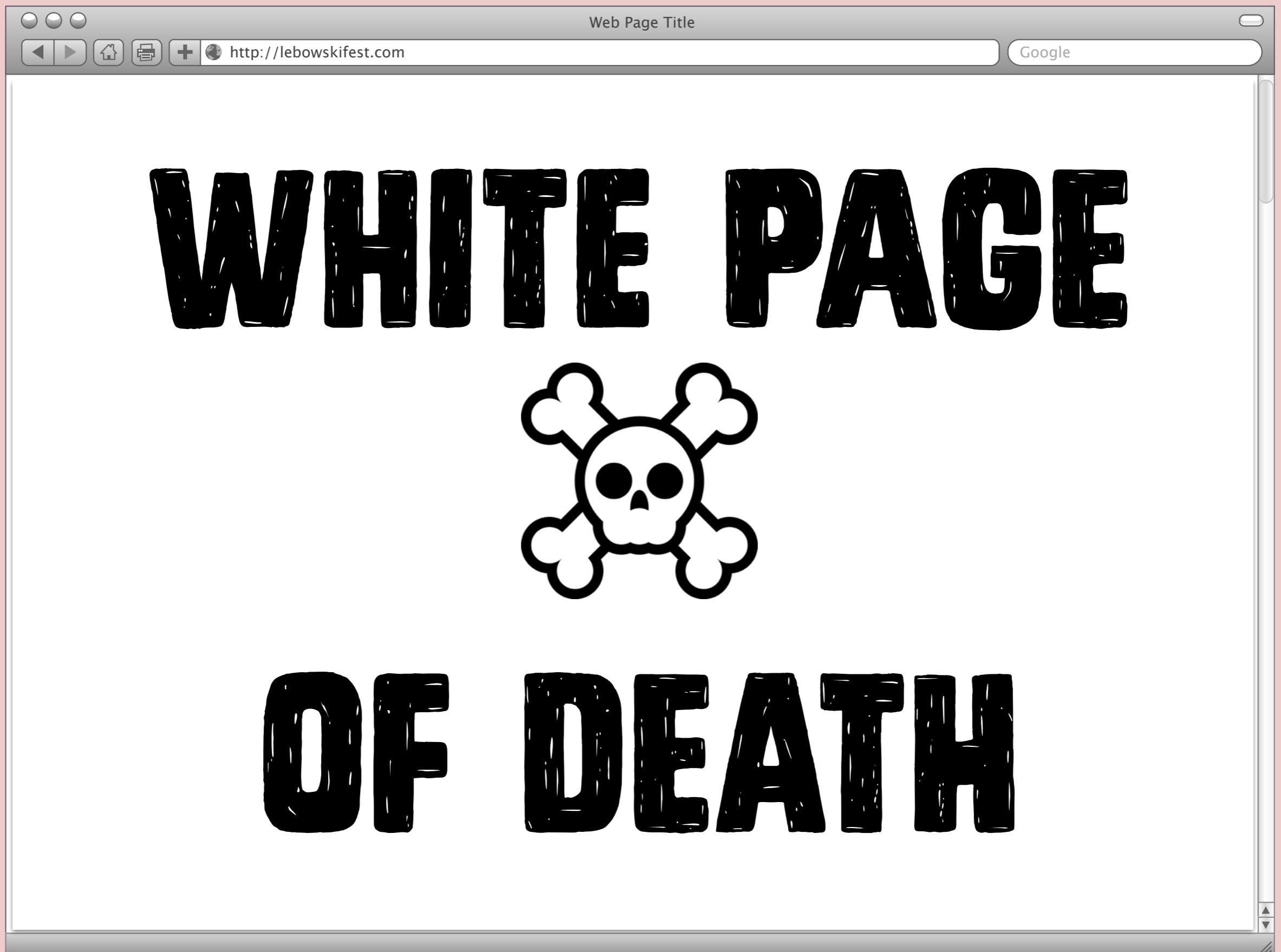
A photograph of a person with short brown hair, wearing a light-colored flight suit over a dark shirt, and a pair of aviator goggles. They are looking down at a map or document held in their hands. The background is dark and out of focus.

Error handling

App #2

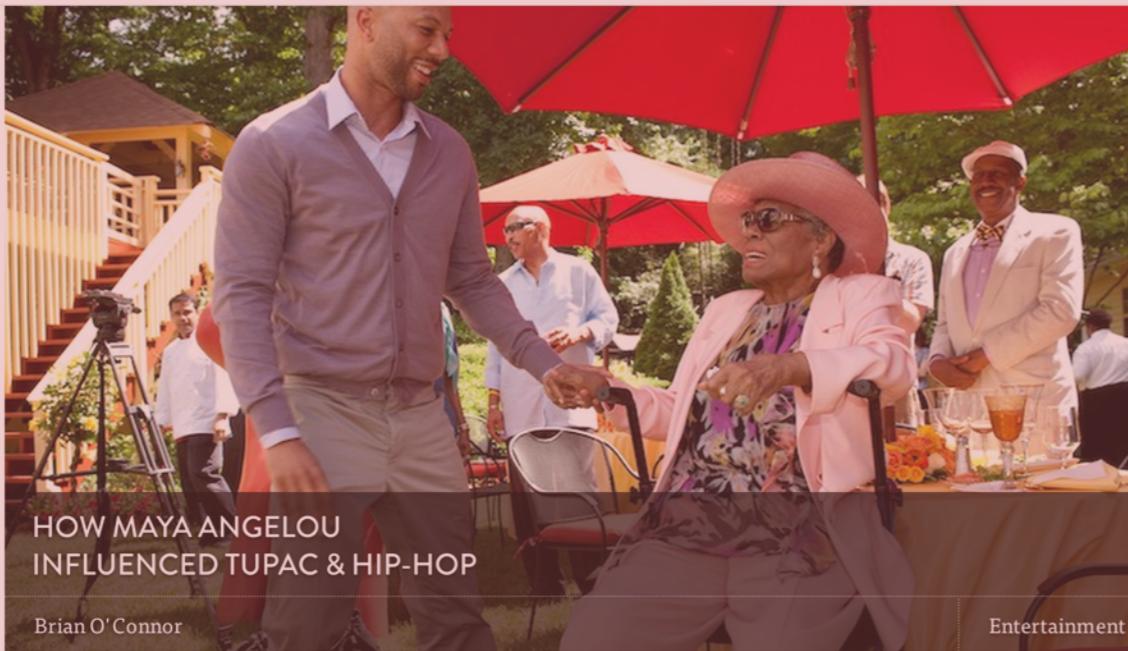


App #2





Jump to crawl



HOW MAYA ANGELOU
INFLUENCED TUPAC & HIP-HOP

Brian O'Connor

Entertainment



THIS IS WHAT SPELLING BEES DO
TO CHILDREN

Kaitlin Reilly

Entertainment



DO YOU NEED TO CHANGE YOUR EBAY
PASSWORD?

Lark Turner

45 minutes ago



MEET YOUR NIPPLE RIGHTS ACTIVIST

Aly Semigran

2 hours ago

MUST READS

THIS STAR ISN'T ON TWITTER, BUT YOU CAN
CALL HER

[policymic.com](#)



UCSB SHOOTING IGNITES DISCUSSION ON
MISOGYNY

By Emma Cueto



HOW TO SURVIVE FESTIVALS WHEN YOU
HATE FESTIVALS

By Rachel Semigran



9 STYLE LESSONS FROM SUMMER
ROBERTS

By Shorey Andrews



14 THINGS EVERY WOMAN HAS THOUGHT
DURING SEX

[womenshealthmag.com](#)



GRADUATION CEREMONIES THAT WENT
HORRIBLY WRONG

By Samantha Sabin



88 VINTAGE SLANG TERMS TO BRING BACK
IMMEDIATELY

By Tori Telfer



App #2



App #2



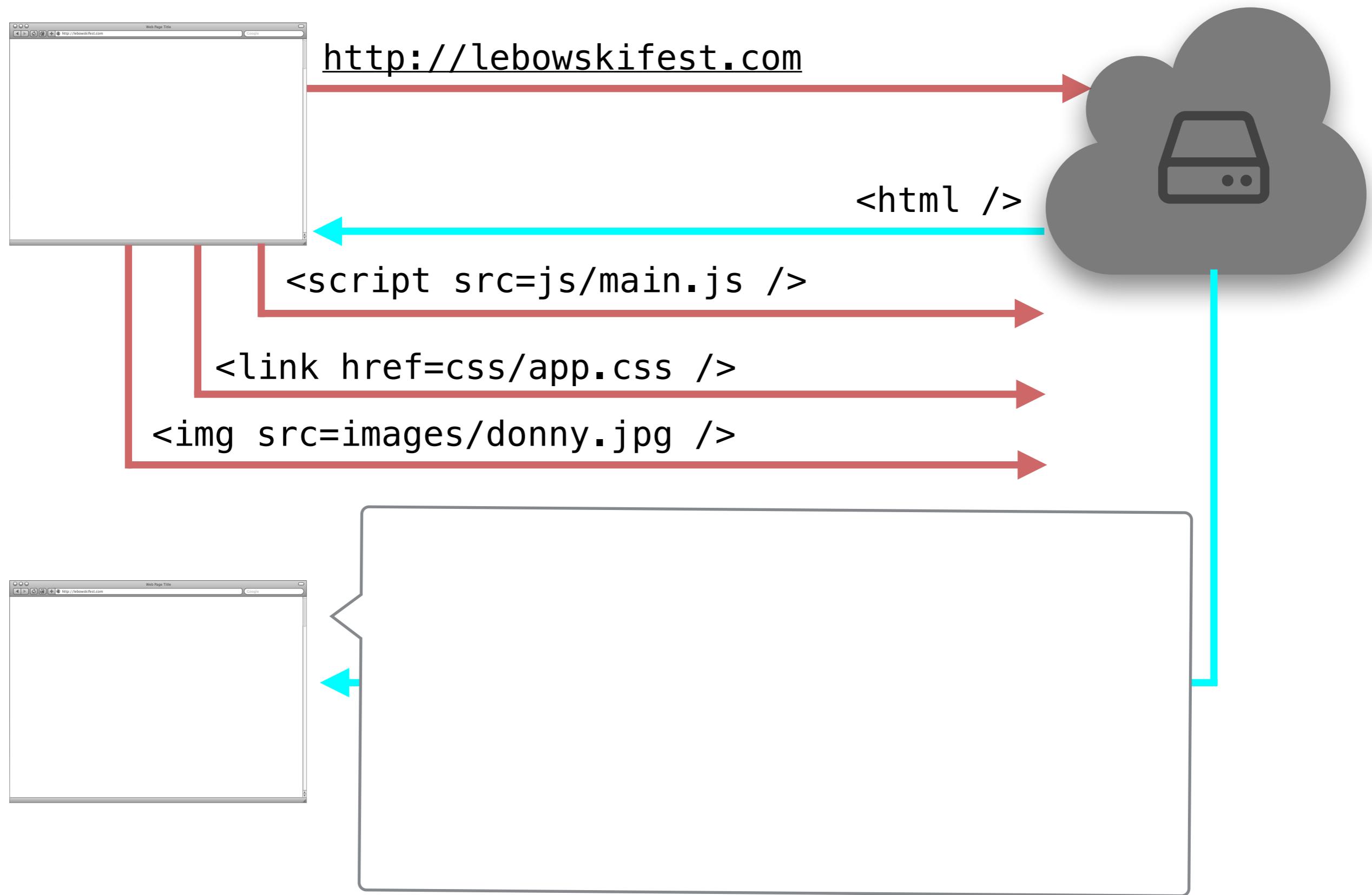
Progressive Enhancement

The rug

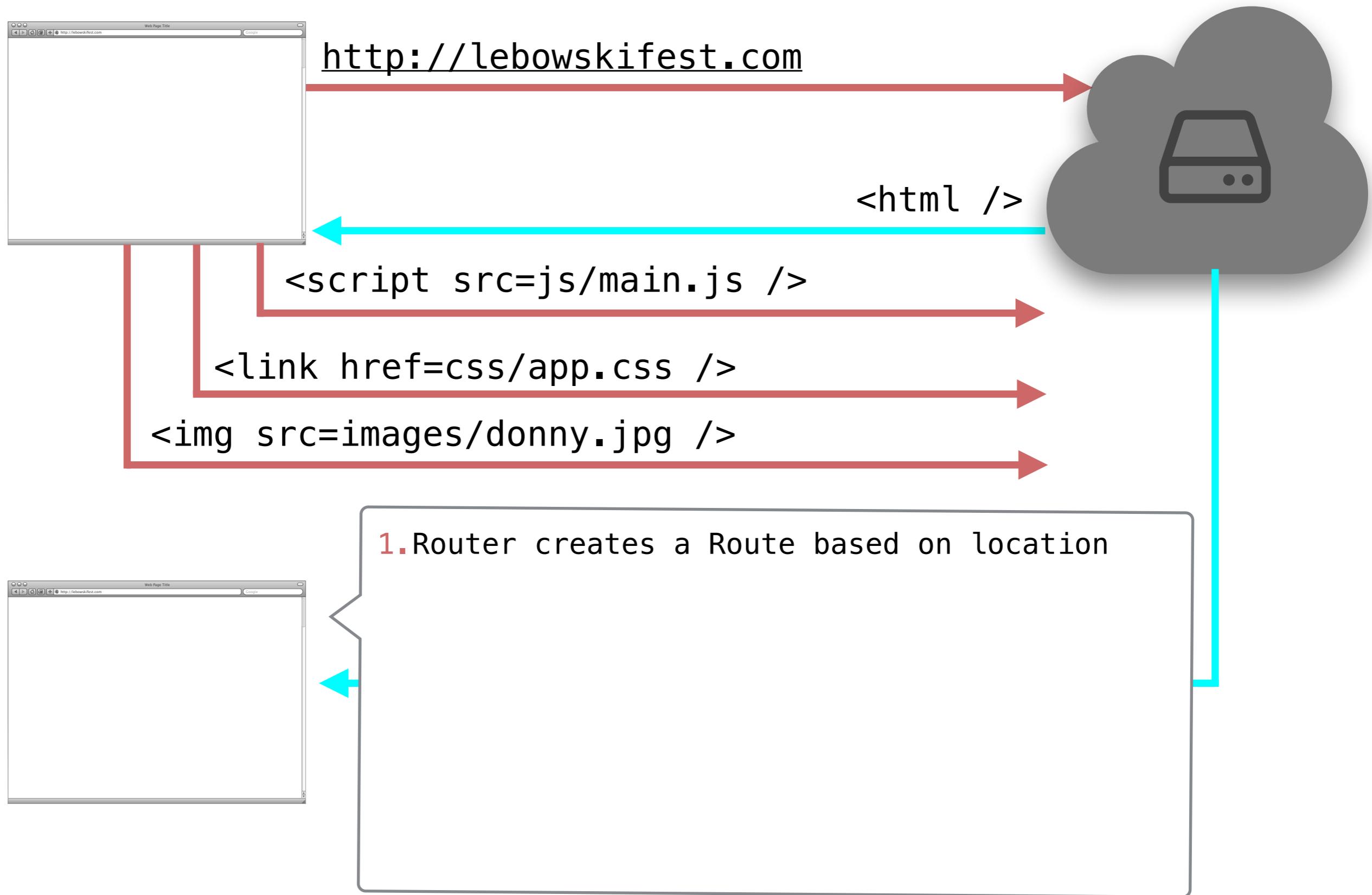


Ember

App #2



App #2

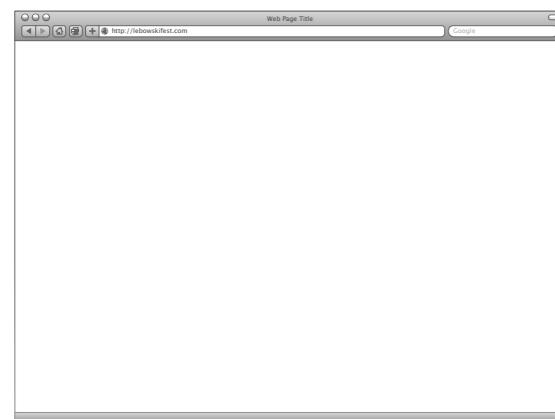


App #2

`http://lebowskifest.com/api/scenes`

MVC

{ }



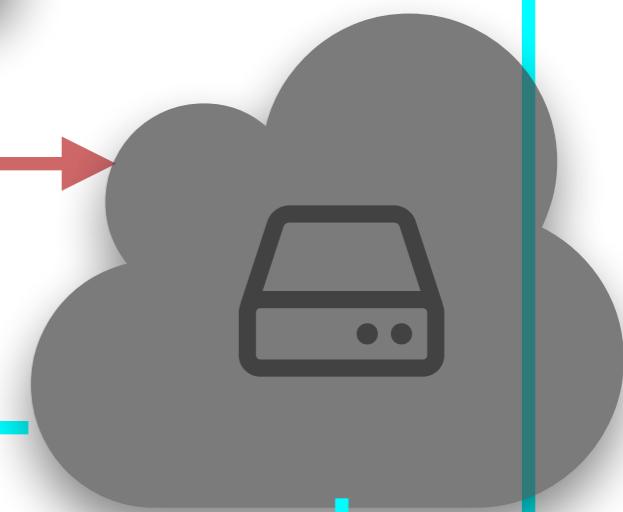
`http://lebowskifest.com`

`<html />`

`<script src=js/main.js />`

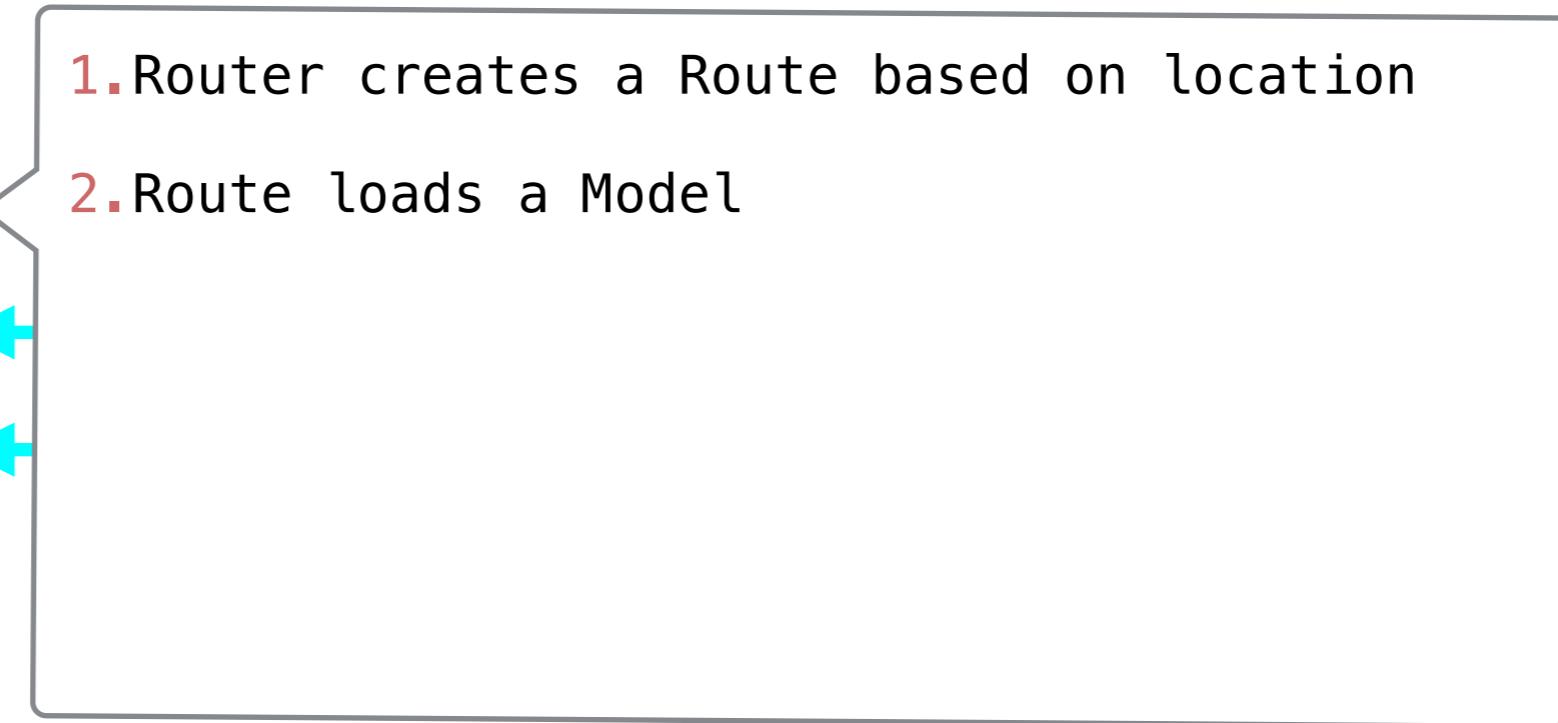
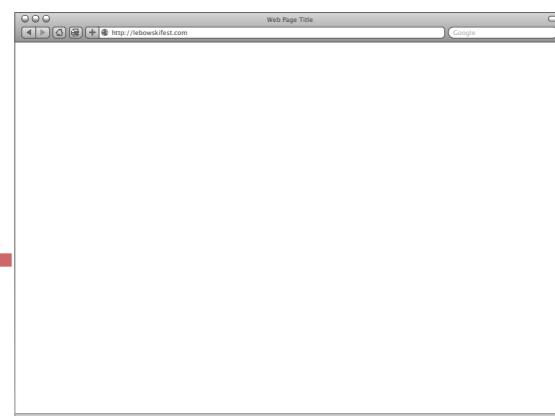
`<link href=css/app.css />`

``

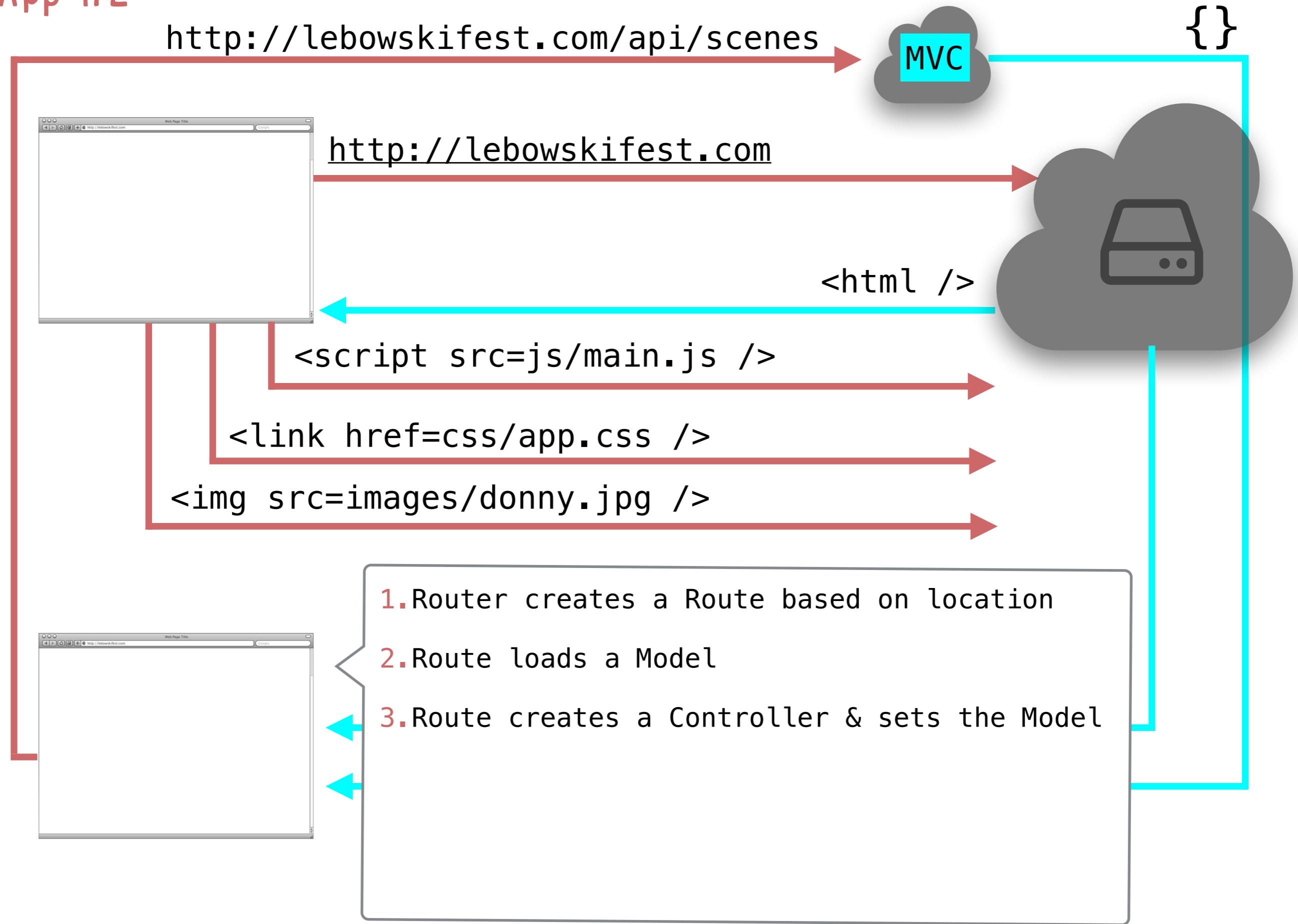


1. Router creates a Route based on location

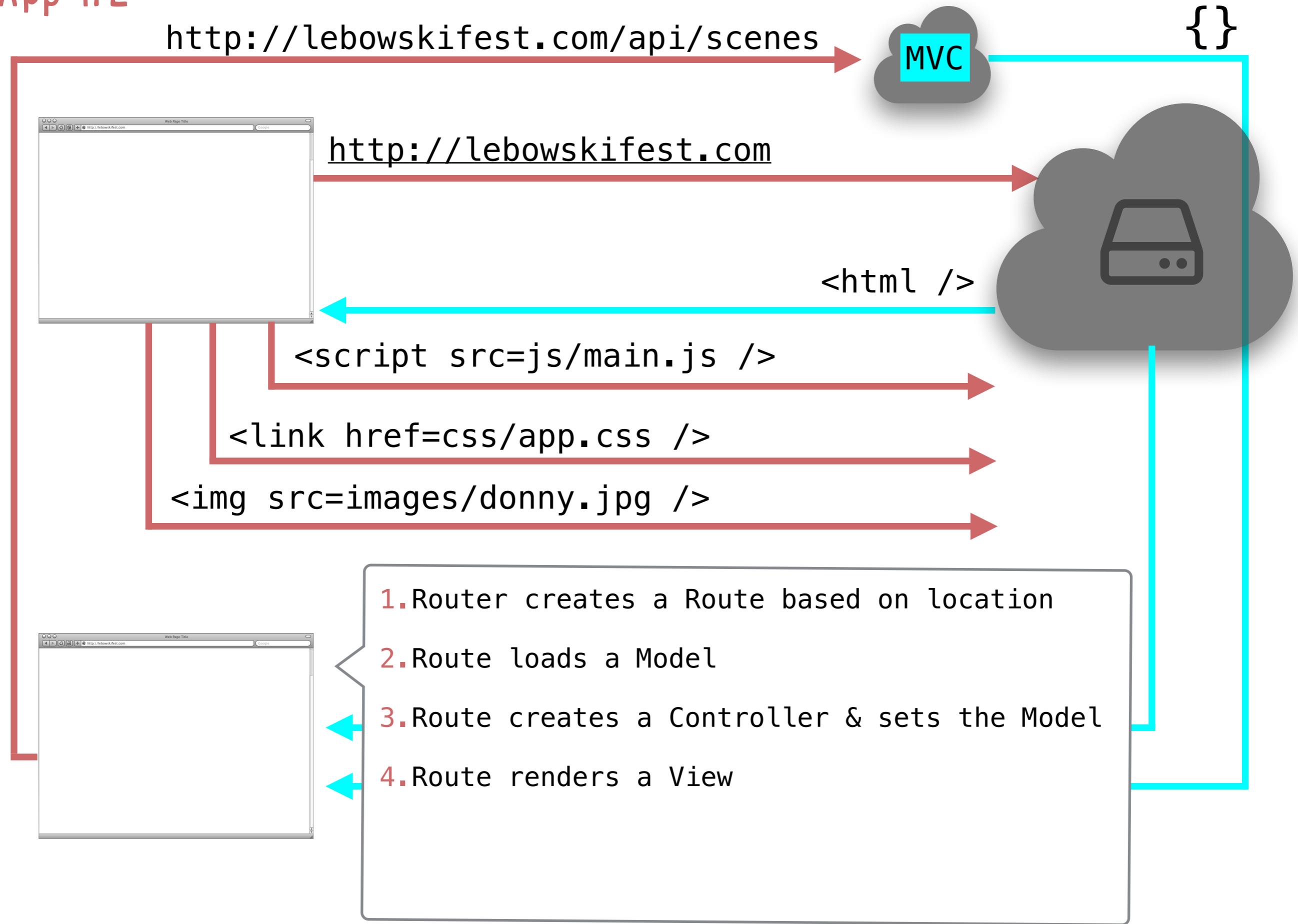
2. Route loads a Model



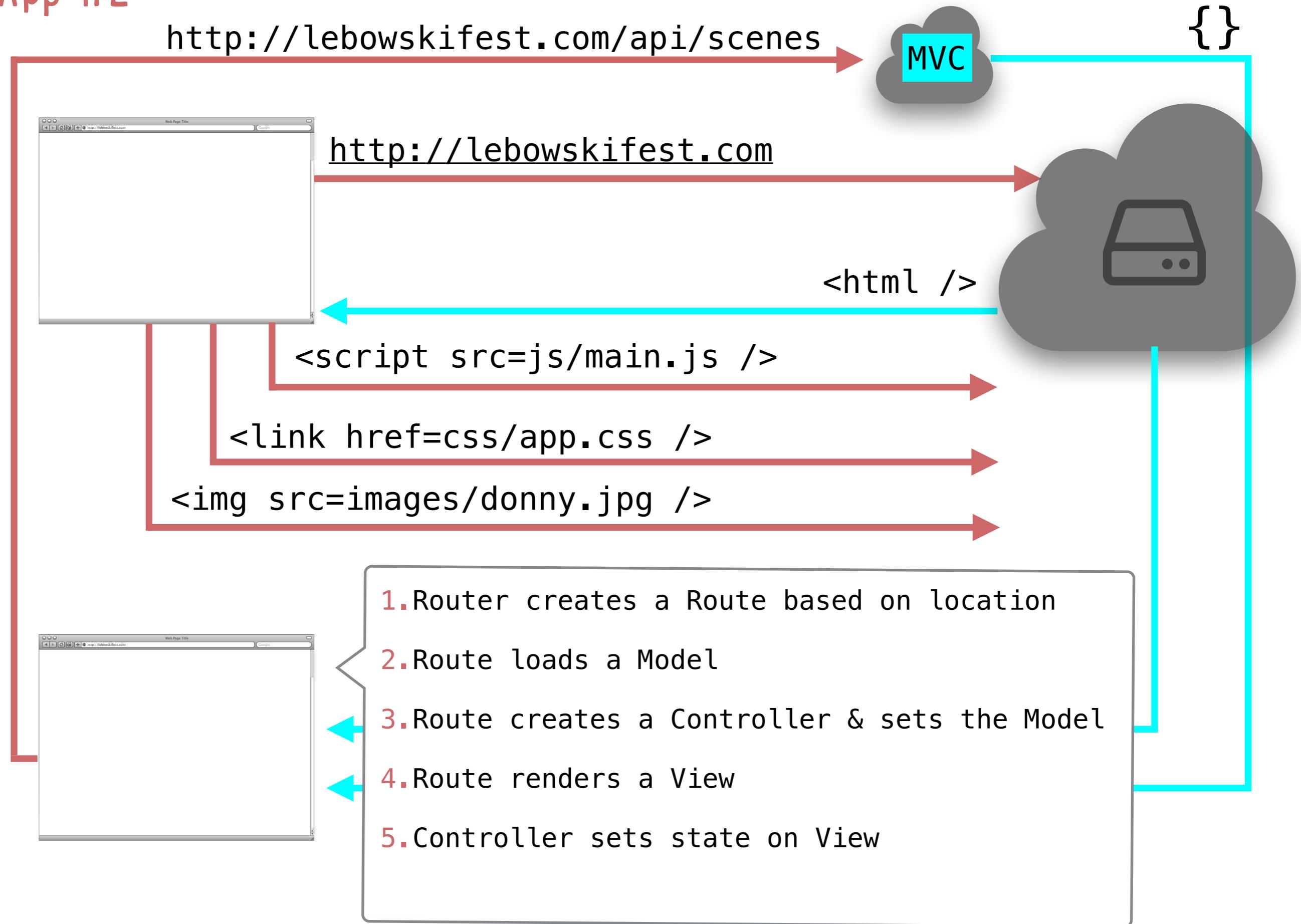
App #2



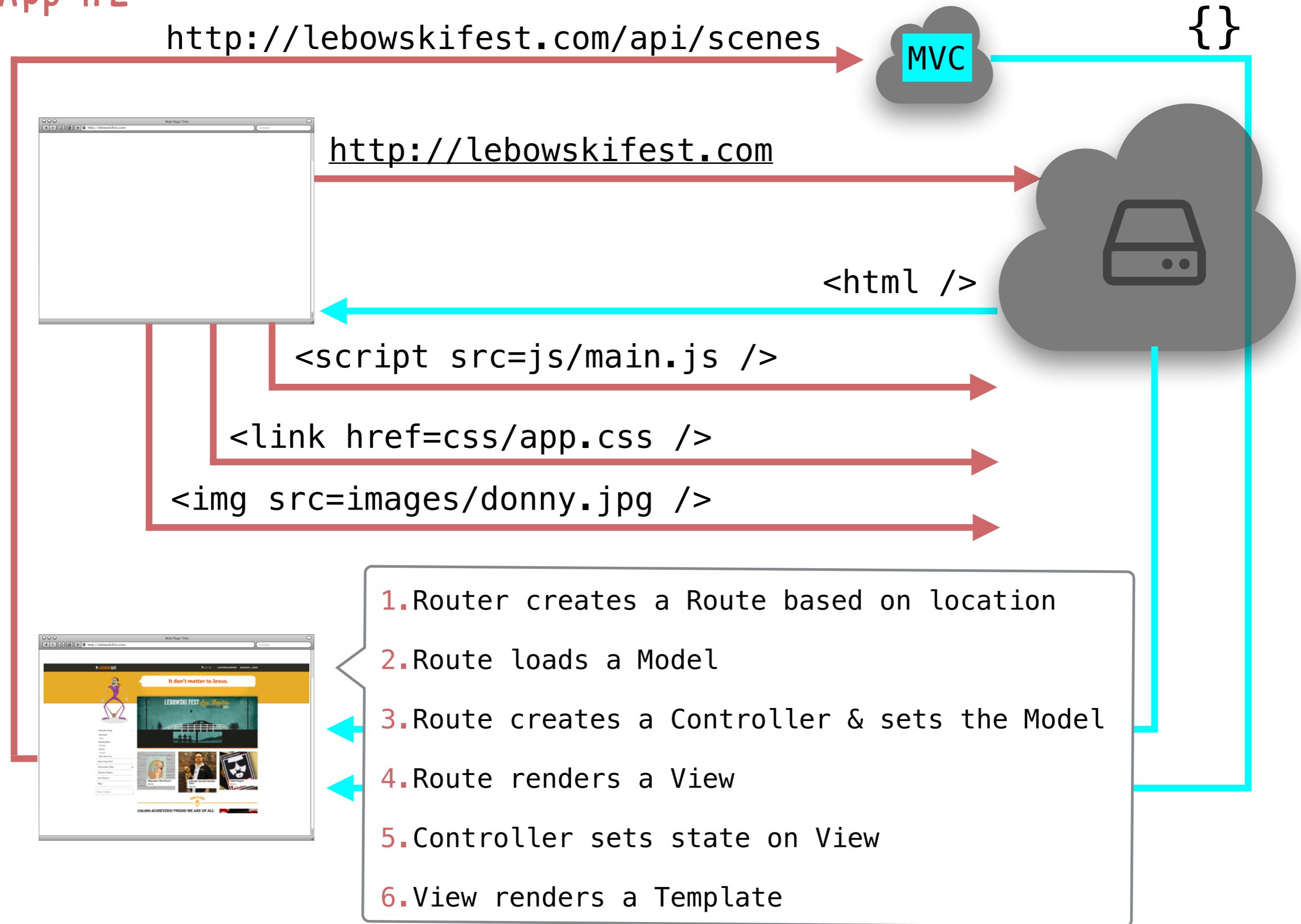
App #2



App #2



App #2



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>
```

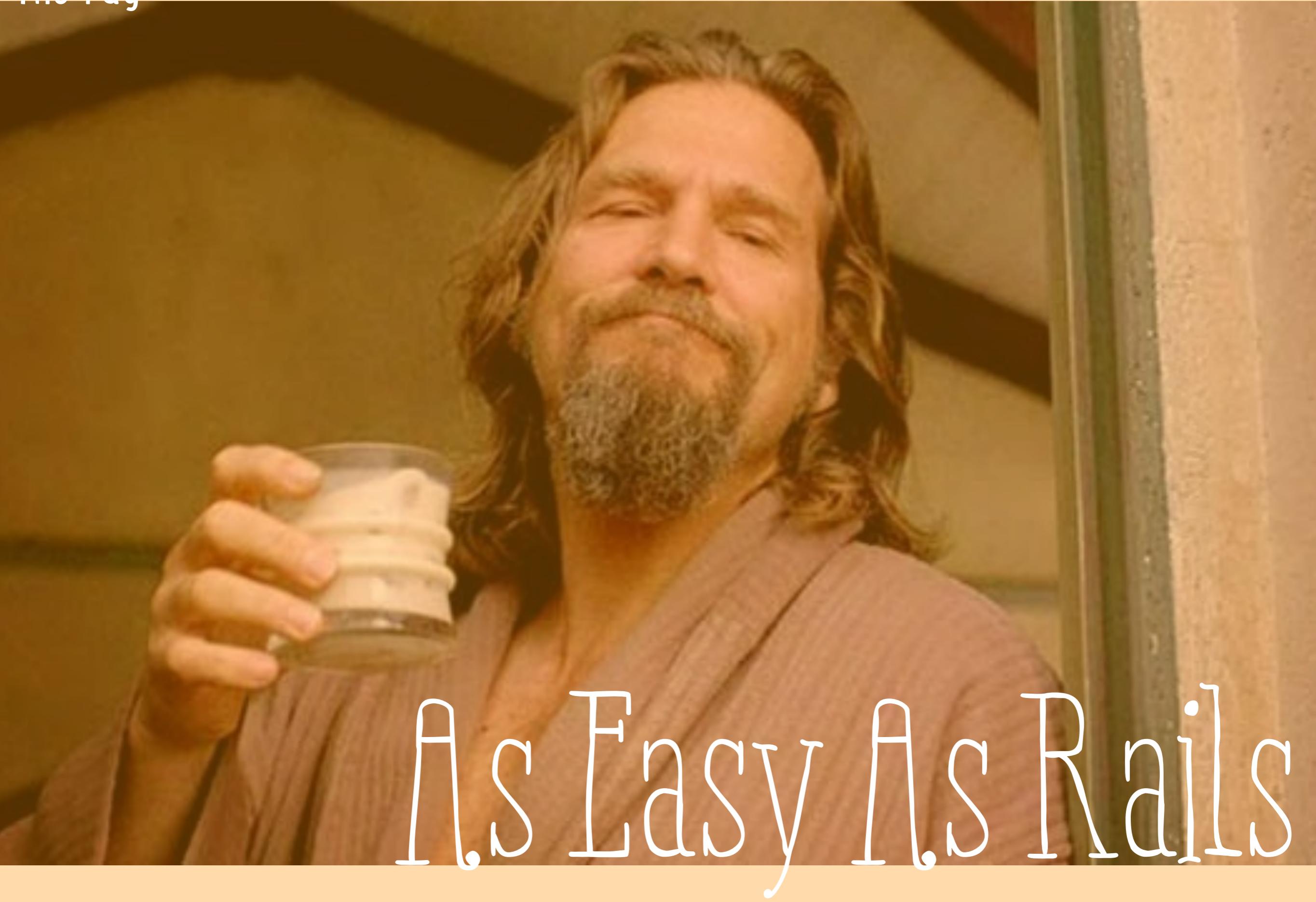
Templates

The rug



Conventions

The rug



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

The rug

\$ 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

The rug

```
$ tree dist
```

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

Building

The rug

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

Deployment

The rug

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

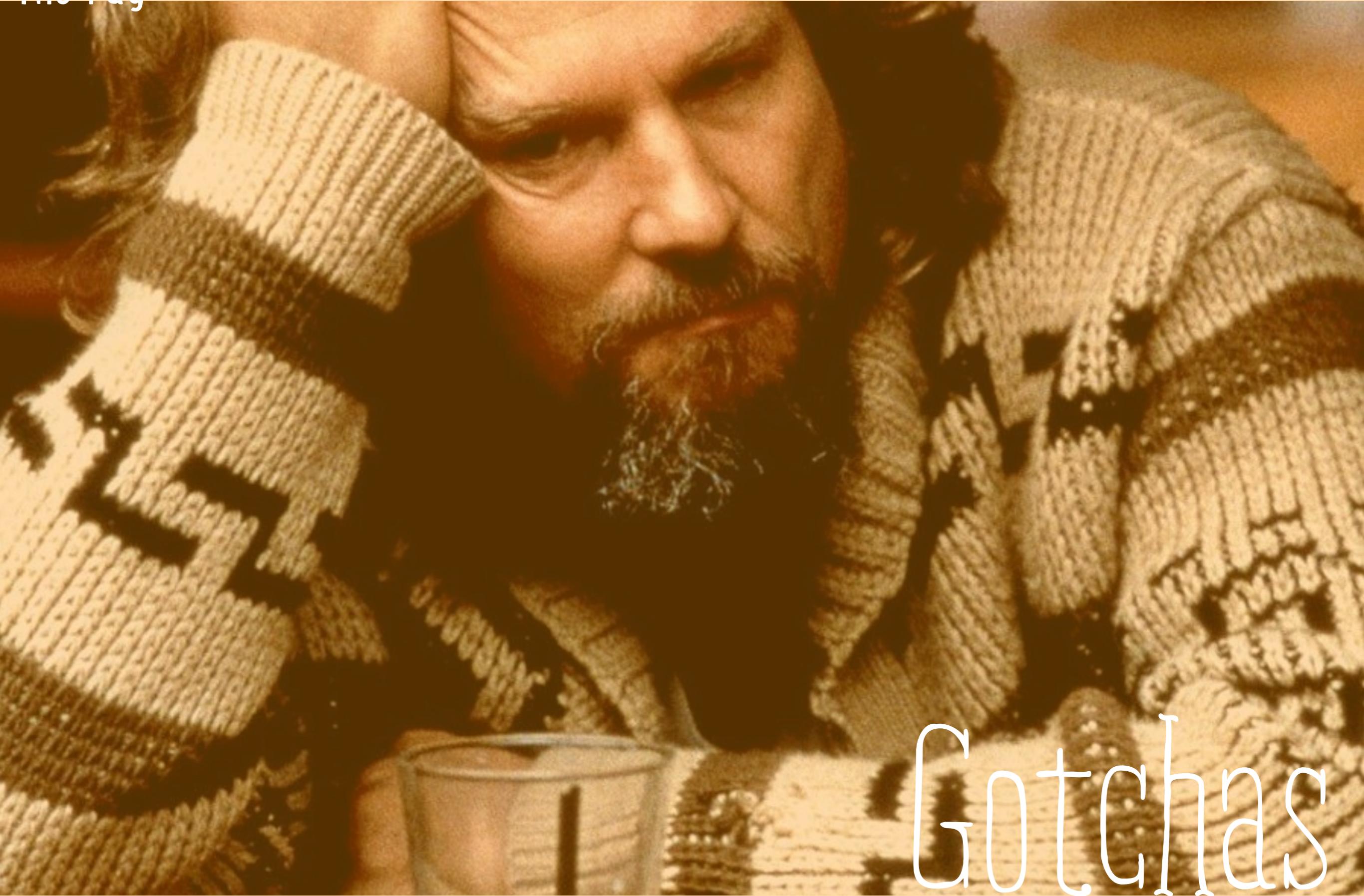
Deployment

The rug

```
$ git push heroku master
```

Deployment

The rug



Gotchas

The rug



Going against the grain

The rug



Handlebars

The rug



Documentation

The rug



StackOverflow

Thanks,
Jim





Questions?



Why Ember and not *?