Ember 101

WTH is Ember and what does it do?

What is Ember?

JavaScript MVC framework for building web apps

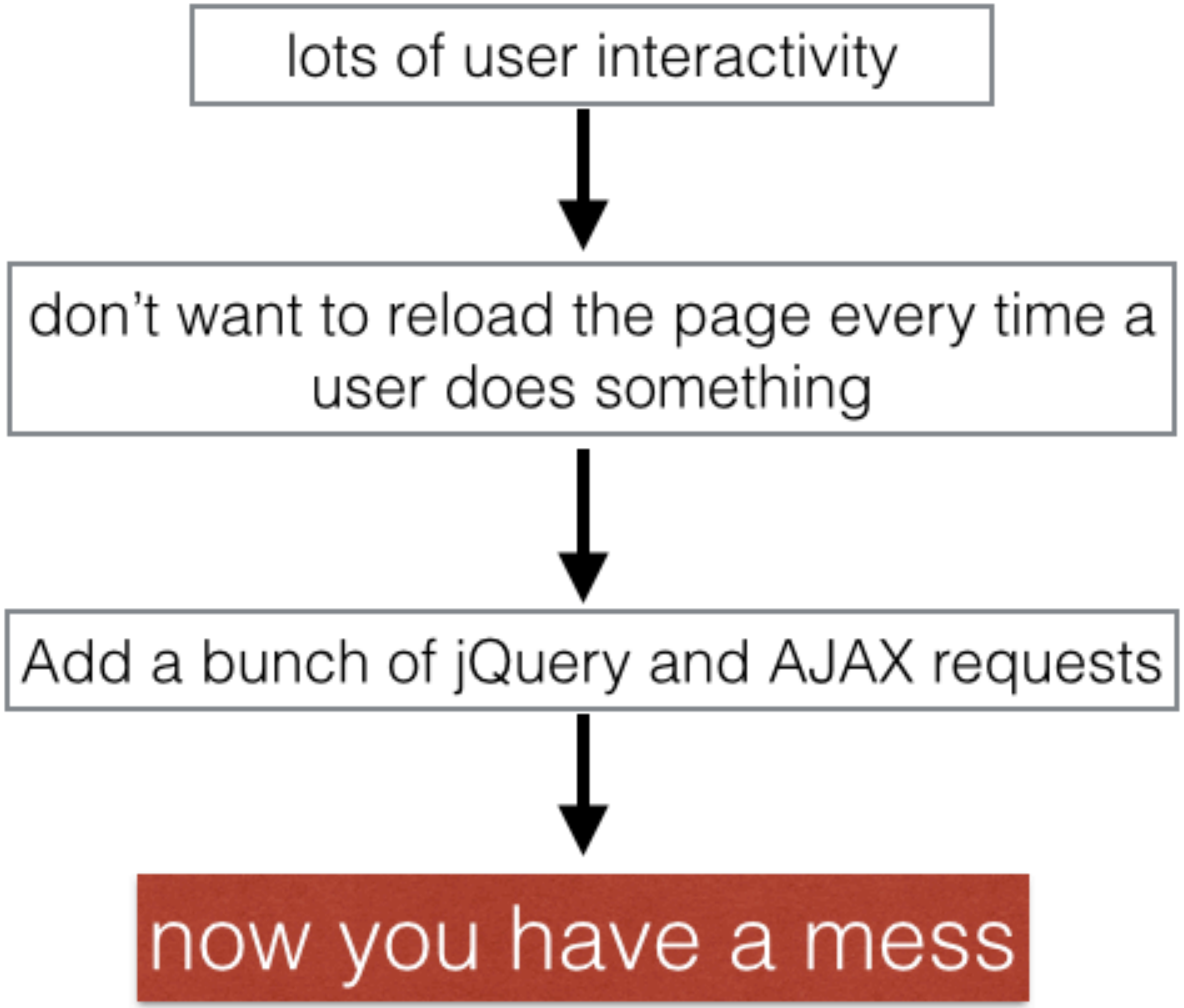
I don't know what that means.

Components of (Ember's) MVC framework

Router - translates the URL into some application state
Models - objects with properties and methods
Routes - retrieve data & render templates
Controllers - like decorators in Rails
Templates - HTML that is dynamically updated based on the application's state

Why use a JavaScript framework?

What's wrong with Rails?



lots of user interactivity

don’t want to reload the page every time a user does something

Add a bunch of jQuery and AJAX requests

now you have a mess

OK fine.

But why Ember?

Because they have the best KoolAid

Because they have the best KoolAid

lol j/k

Why Ember?

- Convention-driven

Why Ember?

- Convention-driven
- Router

Why Ember?

- Convention-driven
- Router
- Nice tools like `ember-cli`

Why Ember?

- Convention-driven
- Router
- Nice tools like `ember-cli`
- The community

That's all very nice.

So how does it work?

Directory Structure

```
$ ember new demo-app

demo-app/
├── app/
│   ├── routes/
│   ├── models/
│   ├── controllers/
│   ├── components/
│   ├── templates/
│   ├── router.js
│   ├── app.js
│   └── index.html
├── tests/
│   ├── acceptance/
│   └── unit/
└── dependencies, config, & other stuff
```

app/index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>DemoApp</title>
    <meta name="description" content="">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    {{content-for 'head'}}

    <link rel="stylesheet" href="assets/vendor.css">
    <link rel="stylesheet" href="assets/demo-app.css">

    {{content-for 'head-footer'}}
  </head>
  <body>
    {{content-for 'body'}} // this is where your templates go

    <script src="assets/vendor.js"></script>
    <script src="assets/demo-app.js"></script>

    {{content-for 'body-footer'}}
  </body>
</html>
```

Ember vs. Rails

Ember vs. Rails

```
GET  /posts
```

The Router

```
// app/router.js

Router.map(function() {
  this.resource('posts', function() {
    this.route('show', { path: ':id' });
  });
});
```

```
# config/routes.rb

Rails.application.routes.draw do
  resources :posts, only: [:index, :show]
end
```

Routes

```
// app/routes/posts/index.js

export default Ember.Route.extend({
  model: function() {
    return this.store.find('post');
  }
});

# app/controllers/posts_controller.rb

class PostsController < ApplicationController
  def index
    @posts = Post.all
  end
end
```

Models

```
// app/models/post.js

import DS from 'ember-data';

export default DS.Model.extend({
  title: DS.attr('string'),
  body: DS.attr('string')
});
```

```
# app/models/post.rb

class Post < ActiveRecord::Base
end
```

Models

```
// app/models/post.js

import DS from 'ember-data';

export default DS.Model.extend({
  // ...

  author: DS.belongsTo('user', { async: true }),
  comments: DS.hasMany('comment', { async: true })
});
```

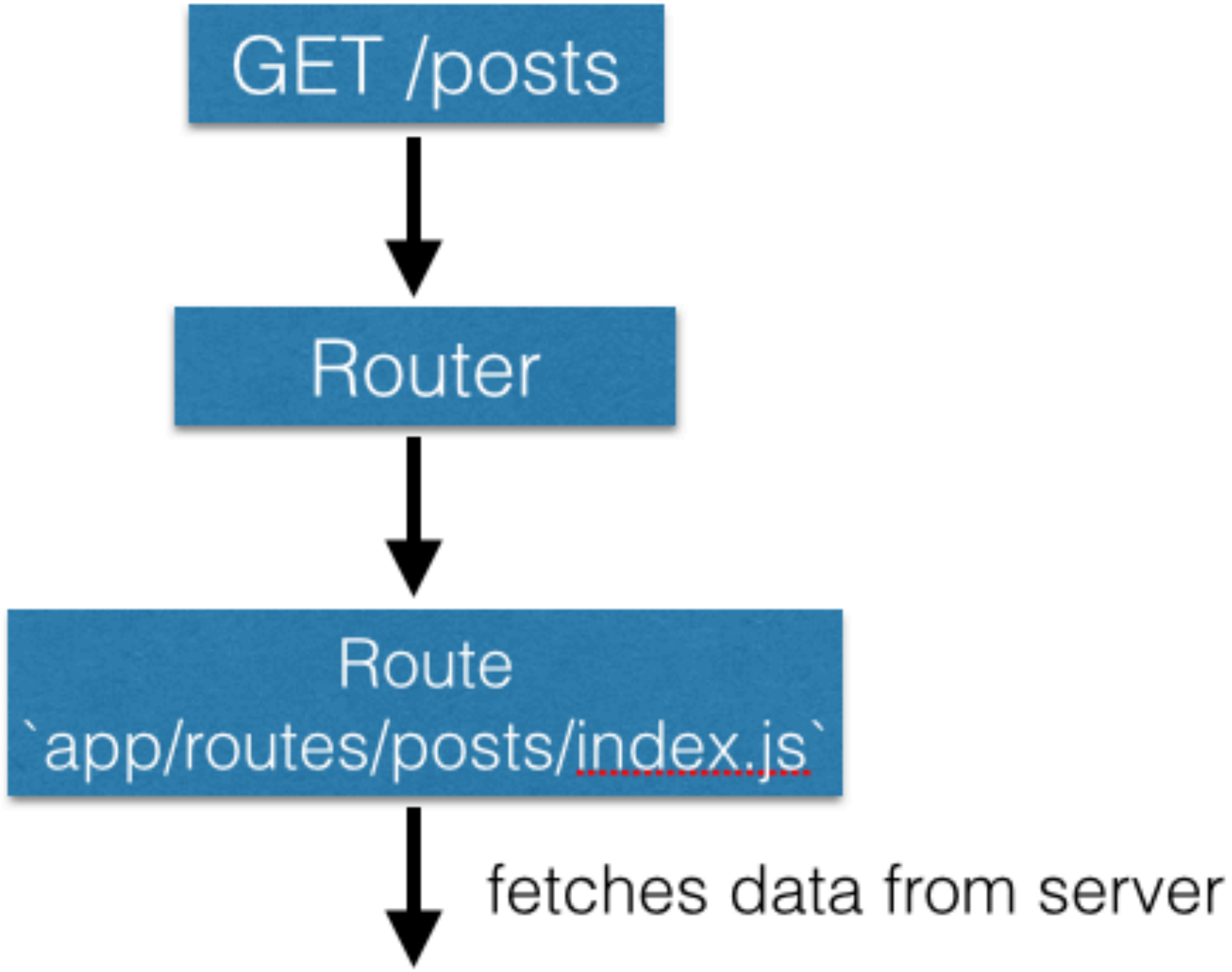
```
# app/models/post.rb

class Post < ActiveRecord::Base
  belongs_to :author, class_name: 'User'
  has_many :comments
end
```

Models

```
// app/models/post.js

import DS from 'ember-data';

export default DS.Model.extend({
  // ...

  commentCount: function() {
    return this.get('comments').length;
  }.property('comments')
});
```

```
# app/models/post.rb

class Post < ActiveRecord::Base
  # ...

  def comment_count
    comments.length
  end
end
```

Recap



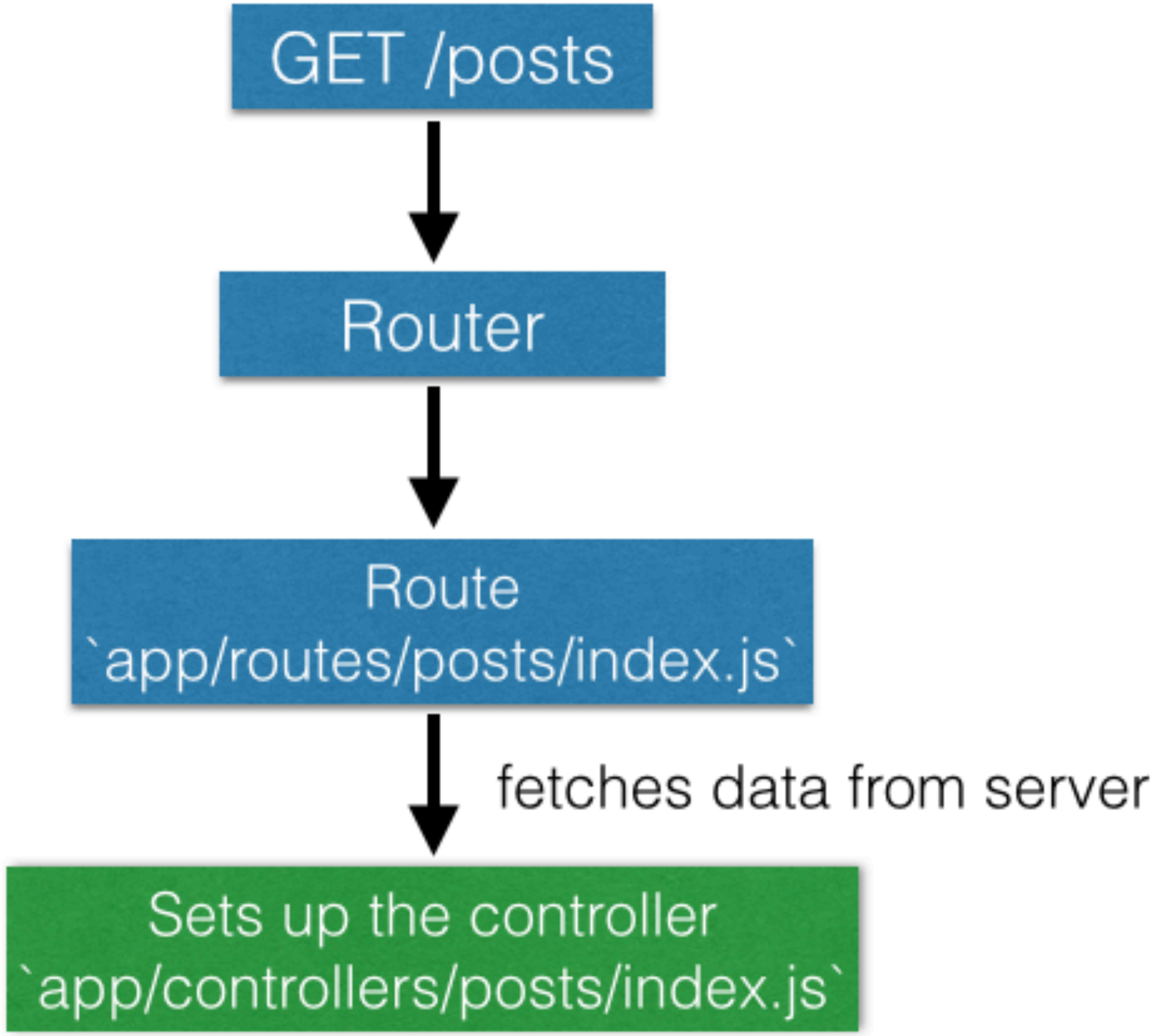
GET /posts

Router

Route
`app/routes/posts/index.js`

fetches data from server

Recap



GET /posts

Router

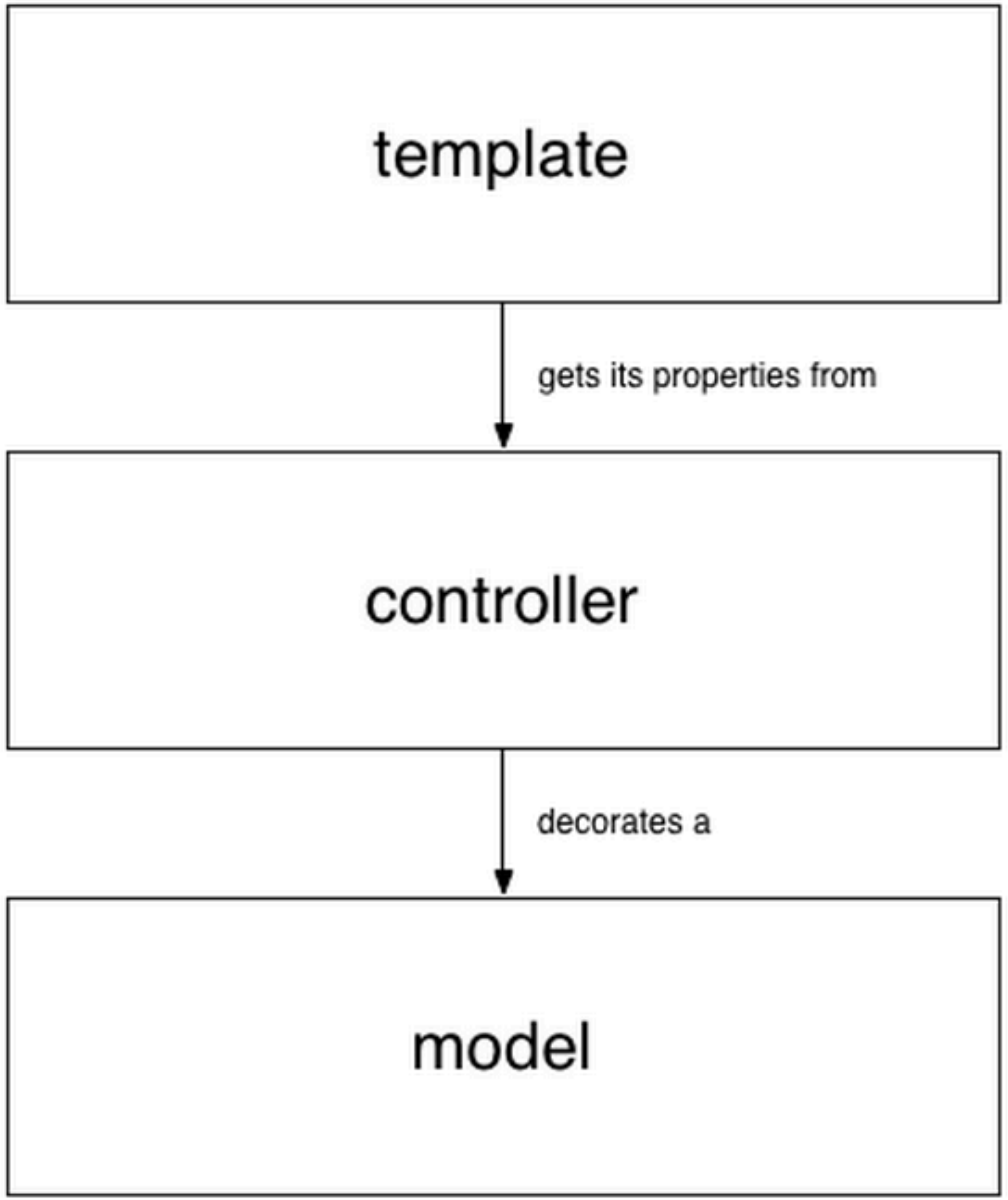
Route
`app/routes/posts/index.js`

fetches data from server

Sets up the controller
`app/controllers/posts/index.js`

Controllers

source: http://guides.emberjs.com/v1.10.0/controllers/



template

gets its properties from

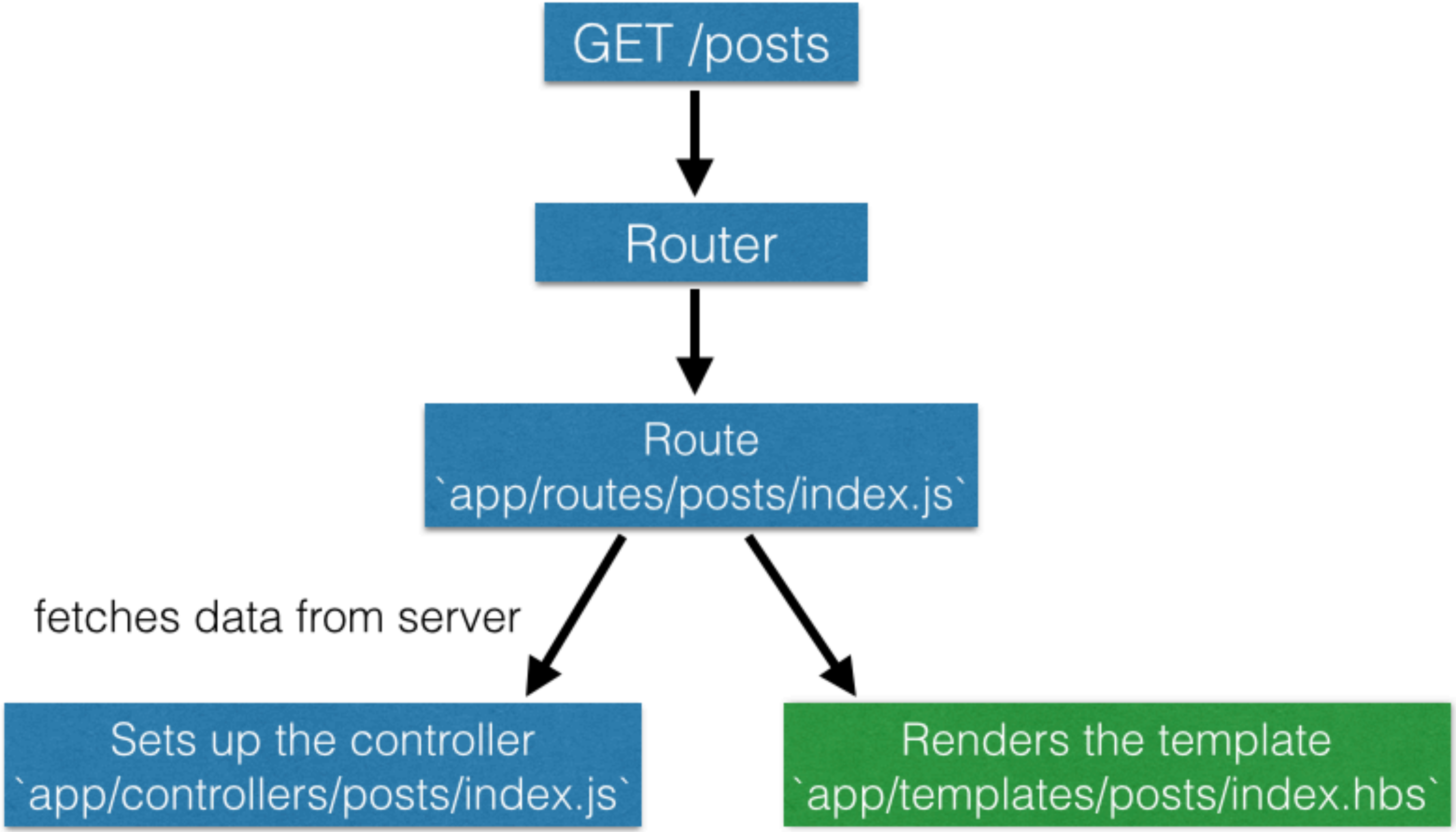
controller

decorates a

model

Moving on.

Templates!



GET /posts

Router

Route
`app/routes/posts/index.js`

fetches data from server

Sets up the controller
`app/controllers/posts/index.js`

Renders the template
`app/templates/posts/index.hbs`

Templates

```
<!-- app/templates/posts/index.hbs -->

<div class="posts">
  <h3>All Posts</h3>

  <ul>
    {{#each post in model}}
      <li>
        {{link-to post.name 'posts.show' post}}
      </li>
    {{/each}}
  </ul>
</div>
```

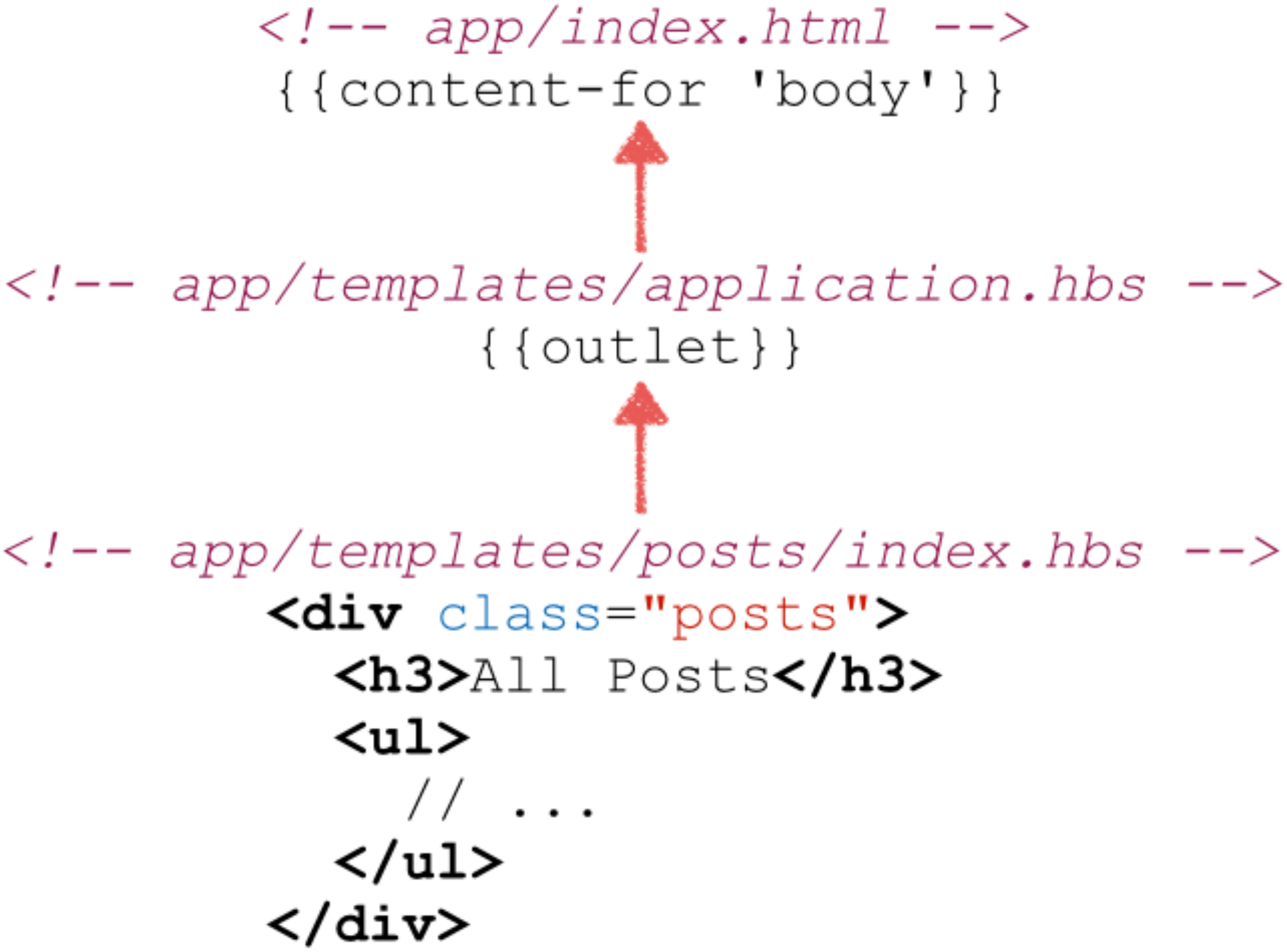
Templates

```
<!-- app/templates/posts/index.hbs -->
<ul>
  {{#each post in model}}
    <li>{{link-to post.name 'posts.show' post}}</li>
  {{/each}}
</ul>
```

```
<!-- app/views/posts/index.html.erb -->
<ul>
  <% @posts.each do |post| %>
    <li><%= link_to post.name, post %></li>
  <% end %>
</ul>
```

How are templates rendered in a single-page application?

Outlets



```
<!-- app/index.html -->
{{content-for 'body'}}
```

```
<!-- app/templates/application.hbs -->
{{outlet}}
```

```
<!-- app/templates/posts/index.hbs -->
<div class="posts">
  <h3>All Posts</h3>
  <ul>
    // ...
  </ul>
</div>
```

That's it!

Now you know Ember

Action handling

Creating a new post

```
// app/routes/posts/new.js

export default Ember.Route.extend({
  model: function() {
    return this.store.createRecord('post');
  },
  actions: {
    save: function() {
      var _this = this;
      var model = this.currentModel;

      model.save().then(function() {
        _this.transitionTo('post.show', model);
      }, function() {
        // Failed!
      });
    }
  }
});
```

Creating a new post

```
<!-- app/templates/posts/new.hbs -->

<form {{action 'save' model on='submit'}}>
  <div>
    {{input name="name" value=model.name placeholder='Enter post name'}}

    {{#each error in model.errors.name}}
      <p>{{error.message}}</p>
    {{/each}}
  </div>

  <div>
    {{input name="body" value=model.body placeholder='Enter post body'}}
  </div>

  <div>
    <input type="Submit" class="button">
  </div>
</form>
```

Components

Reusable UI elements

Topbar navigation

```js
// app/components/topbar-nav.js
export default Ember.Component.extend({
  tagName: 'nav',
  classNames: ['topbar-nav']
});
```

```hbs
<!-- app/templates/components/topbar-nav.hbs -->
<ul>
  <li>My Awesome Blog</li>
  <!-- sign up / in / out links -->
</ul>
```

```hbs
<!-- app/templates/application.hbs -->
{{topbar-nav}}

{{outlet}}
```

Ember 2.0

Resources

Ember guides: http://guides.emberjs.com/v1.11.0/

DockYard `ember-cli` **/** `rails-api` **tutorial: http://reefpoints.dockyard.com/2014/05/07/building-an-ember-app-with-rails-part-1.html**

Ember CLI 101 book: https://leanpub.com/ember-cli-101

Frontside podcast: https://frontsidethepodcast.simplecast.fm/

Check out the code for some apps built in Ember:

Questions?