

Warming Up With **ember.js**

Level 3 - A Route Through the Woods

Resource Routes



Review

The name of the controller matches the route and template names

app.js

```
App.IndexController = Ember.Controller.extend({
  productsCount: 6,
  logo: '/images/test.png',
  time: function() {
    return (new Date()).toDateString();
  }.property()
});
```

index.html

```
<script type='text/handlebars' data-template-name='index'>
  <p>There are {{productsCount}} products</p>
  <img {{bind-attr src='logo'}} alt='Logo' />
  <p>Current Time: {{time}}</p>
</script>
```

Templates can use properties from a controller





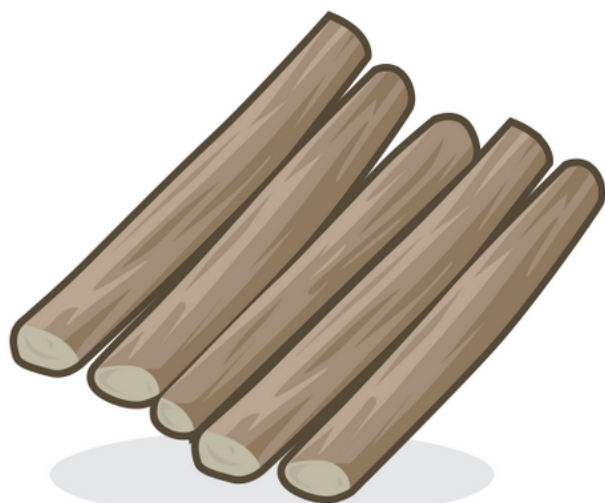
Products



Flint

Flint is a hard, sedimentary cryptocrystalline form of the mineral quartz, categorized as a variety of chert.

Buy for \$99



Kindling

Easily combustible small sticks or twigs used for starting a fire.

Buy for \$249

We'll show all of our products on a new products page

Use route for adjectives, verbs, adverbs

app.js

```
App.Router.map(function() {  
  this.route('about');  
});
```



Adjective Routes

```
this.route('onsale');  
this.route('expiring');  
this.route('new');  
this.route('internal');
```

Verb Routes

```
this.route('signup');  
this.route('ignite');  
this.route('help');  
this.route('rate');
```



Use 'Resource' Routes for Nouns

app.js

```
App.Router.map(function() {  
  this.route('about');  
  this.resource('products');  
});
```



Noun Resources

```
this.resource('tree');  
this.resource('review');  
this.resource('books');  
this.resource('contacts');
```



Can be singular or plural

The resource keyword has some additional functionality.

That difference is what this level is all about!



Resource Route Options

Like with route, we can pass in an optional path

app.js

```
App.Router.map(function() {  
  this.route('about');  
  this.resource('products', { path: '/items' });  
});
```



Resource Views

app.js

```
App.Router.map(function() {  
  this.route('about');  
  this.resource('products');  
});
```



Matches the route name

index.html

```
<script type='text/x-handlebars' data-template-name='products'>  
  <h1>Products</h1>  
</script>
```



Linking to Resources

We can link to this route with its name like any other.

index.html

```
<script type='text/x-handlebars' data-template-name='application'>
  ...
  {{#link-to 'products' tagName='li'}}Products{{/link-to}}
  ...
</script>
```



http://example.com#/products



Flint & Flame

Home

About

Products

Products

© 2013 The Flint & Flame

Credits

Rendering the products template

The link has a class of active if we're within the products route

How can we add some data to this page?

Ember Route



Fetches data and passes it to the Ember Controller

Decides on what model to use for the current route

Decides which template to render to the screen

Each Ember Route will use this icon

A model could be a JavaScript
Object or an Array of objects





Ember Router

Translates a path into a route

Don't get these confused



Ember Route

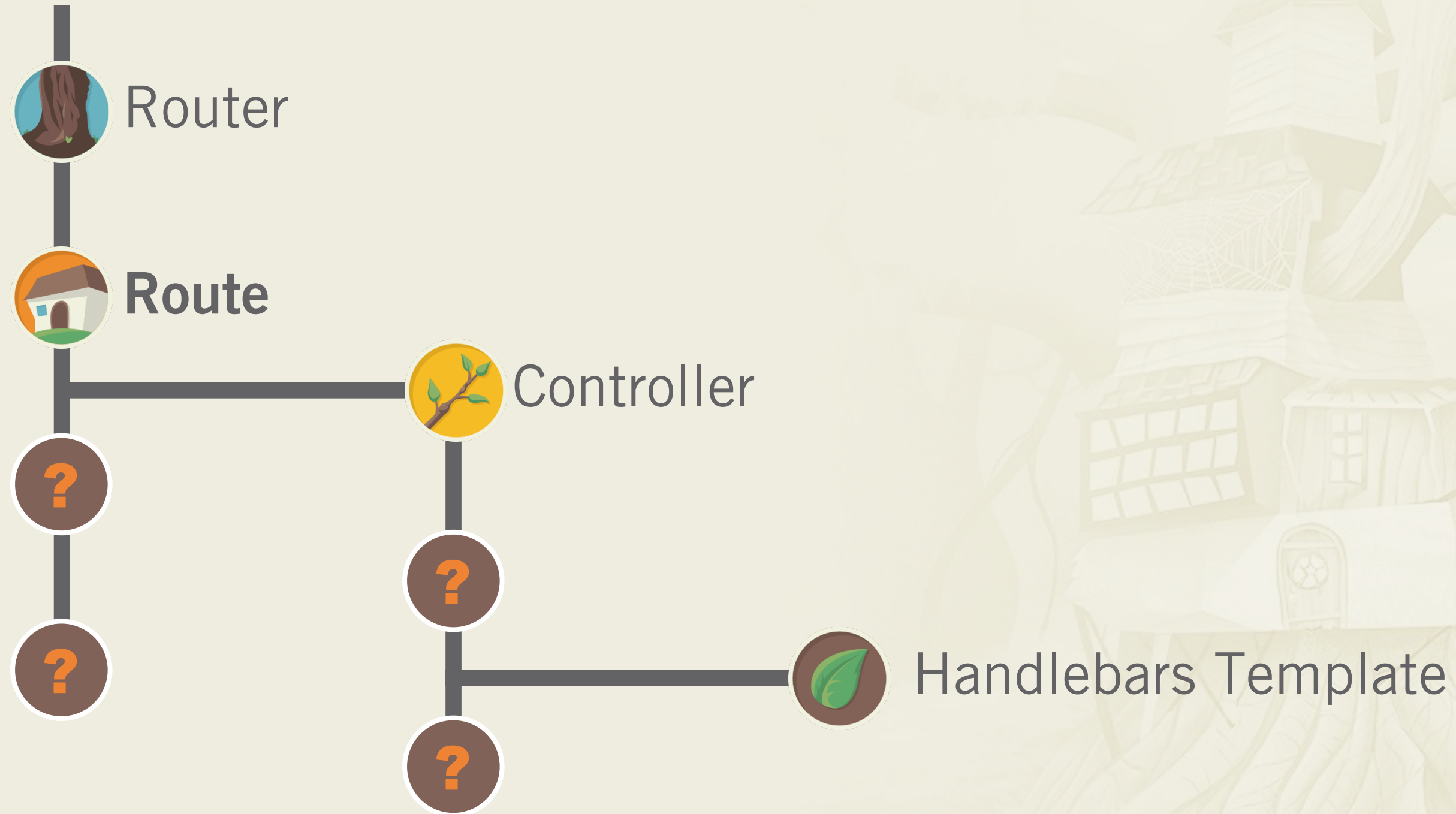
Provides data for the controller



Ember Route

Provides data for the controller

Browser Request



Router? Route? Controller?



Router

Used to define routes our application accepts.

```
this.resource('products');
```



Route

Responsible for getting data from external resources

```
App.ProductsRoute = Ember.Route.extend({ });
```

created by Ember if not defined



Controller

Decorates the model, provides property values

```
App.ProductsController = Ember.Controller.extend({ });
```

created by Ember if not defined



A Very Basic Model

app.js

```
App.PRODUCTS = [  
  {  
    title: 'Flint',  
    price: 99,  
    description: 'Flint is...',  
    isOnSale: true,  
    image: 'flint.png'  
  },  
  {  
    title: 'Kindling',  
    price: 249,  
    description: 'Easily...',  
    isOnSale: false,  
    image: 'kindling.png'  
  }  
];
```

This could be pulled from an API

app.js

```
App.ProductsRoute = Ember.Route.extend({  
  model: function() {  
    return App.PRODUCTS;  
  }  
});
```

The model property should return an object or an array.



Putting it All Together

The Route is responsible for fetching the model. Ember will set this model on the Controller, and that's what'll be used in your Handlebars Template.



Router



Route



Controller



Template

index.html

```
<script type='text/x-handlebars' data-template-name='products'>
  <h2>Products</h2>
  {{model}}
</script>
```



Rendering the Model

index.html

```
<script type='text/x-handlebars' data-template-name='products'>
  <h2>Products</h2>
  {{model}}
</script>
```



```
<h2>Products</h2>
[object Object],[object Object]
```

Model is an array, so we'll need to loop over it!



Looping Over the Array of Objects

index.html

```
<script type='text/x-handlebars' data-template-name='products'>
  <h2>Products</h2>
  {{#each product in model}}
    <h2>{{product.title}}</h2>
  {{/each}}
</script>
```



```
<h1>Products</h1>
<h2>Flint</h2>
<h2>Kindling</h2>
```

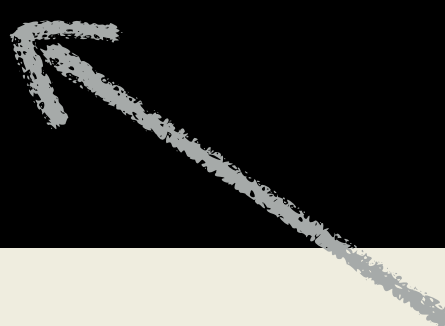


A Simpler #each Loop

#each without extra options will look for a model property.

index.html

```
<script type='text/x-handlebars' data-template-name='products'>
  ...
  {{#each}}
    <h2>{{title}}</h2>
  {{/each}}
</script>
```



Within the #each loop, you can access properties on that product



A Dash of Style

index.html

```
<script type='text/x-handlebars' data-template-name='products'>
<h1>Products</h1>
<ul class='list-unstyled col-md-8'>
  {{#each}}
    <li class='row'>
      <img {{bind-attr src='image'}} class='img-thumbnail col-md-5' />
      <div class='col-md-7'>
        <h2>{{title}}</h2>
        <p class='product-description'>{{description}}</p>
        <p><button class='btn btn-success'>Buy for ${{price}}</button></p>
      </div>
    </li>
  {{/each}}
</ul>
</script>
```

Output a few more properties and add some styling





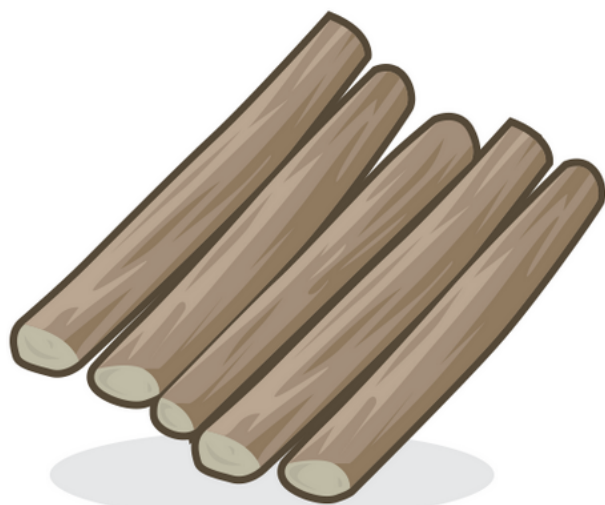
Products



Flint

Flint is a hard, sedimentary cryptocrystalline form of the mineral quartz, categorized as a variety of chert.

Buy for \$99



Kindling

Easily combustible small sticks or twigs used for starting a fire.

Buy for \$249