



## Ember Organization

### Description

Today in class we learned a lot about the organization of files in Ember.

If you have an ember project created in ember-cli and are looking at it.

(ember new <appName>)

**app** – holds all the folders and files to display your app with the exception of the public folder which holds any images or fonts in a folder you create called 'assets'.

**In app** – There are many folders but 3 primary files already generated which are extremely important:

*app.js* – which holds code that basically holds your app together connecting it to the resolver and initializing it so it can run.

*index.html* – Your basic html file where static html can stay

-the head section holds all the connections to dependencies so don't mess with it!

-the body section has special placeholders inside handlebars, `{{ }}`, which are embers

special way of introducing dynamic code. DO NOT DELETE THESE!

-there are also script statements in the body that connect to any vendor or ember dependencies. Leave these alone and let them do their thing.

**router.js**: an extremely important file that sets up the app. When a route is added through: ember generate route <appName> then a special line of code appears in this file that tells ember what the name of your app is, which ember uses to set up its entire file organization. The line of code is this:

```
this.index('<appName>', { path: '/' });
```

<appName> represents the name of the app and must be used in naming several other crucial files in ember, including, templates/appName.hbs, routes/appName.js, models/appName.js, controllers/appName.js, etc. All these folders are in the root app folder.

### Other Crucial folders in app –

All the folders in app are important but the following are crucial:

**templates:** holds the file that is used to render the app. This determines how your app will look on the screen. Handlebars {{ }}, are used to contain dynamic data. The file name will be appName.hbs. (appName is whatever you choose to name the app, duh)

**styles** – hold the css styles for your app. You can put them all in app.css. More later about how to include things like sass.

**routes** – A route is not the router!!! Get this in your head. It's very important. There is one and only one router.js or router. It is to change urls. a route represented by a file in the route folder named something like appName.js represents a particular application state. Interestingly enough this is an object (not a function like in Backbone).

**models** – Represented by appName.js. The model handles ajax requests and holding your data for your application

**helper** – Represented by appName.js

**controllers** – Represented by controllers/appName.js. The controller has the same basic function as a view in Backbone (if you are familiar with this). It mediates between data in routes and the templates. This is where you would manipulate the data before sending it to the template to display it on the page/screen.

components – Not sure what this is for yet. I am sure I will find out soon.

adapters – Not sure about this either, but it has a built in file application.js that has a default adapter that can be changed if needed.

### Category

1. Coding
2. Ember
3. Ember CLI