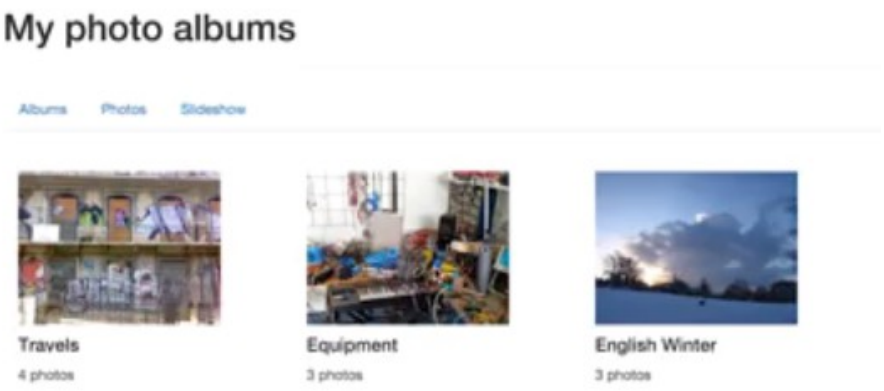
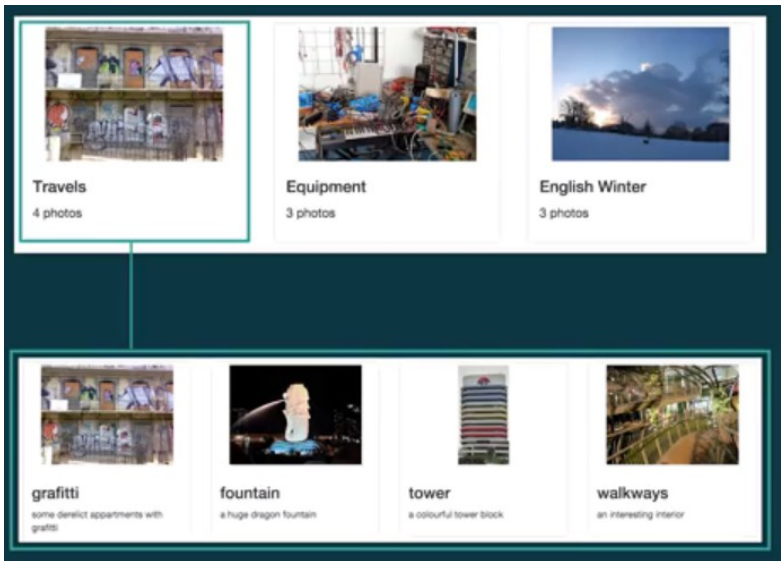
2.4.1.5: Data structure for a complete image gallery

Now you've got everything you need to build really quite a complex website. And the website that goes from having large amounts of individual bits of data whether that's individual images in a photo website, individual blog posts in a In a blog site, individual users and social media sites and allows you to display them all together on one sort of big gallery page or drill down onto individual items and display them. So, for the rest of this week I'm gonna take you through a slightly bigger example of how to do that and I'm gonna carry on with the theme of an image gallery.



- So this sort of a basic example of an image gallery, and it's structured around photo albums, so the first page you see is a bunch of albums, one called Travel, one call Equipment, one called English Winter.
- And I can click on any of these and it brings up the photos inside that album. So we've got various different locations and if I want to see them bigger I just click on that one.
- And I can see it properly and I also have the option of displaying the entire thing as a slide show here. So whichever was the last album I clicked on, it can be displayed as a slideshow.



So the content of this website is a multilevel content. As well as having, we know how to store lots of things, but we are storing lots of albums and inside each of those albums, there are lots of images.

So I'm gonna show you how to create up this website, and I'm gonna look at three basic things.

1. How to represent the content in terms of arrays and images, and arrays and objects,
2. how to create the structure in terms of templates, and then
3. the JavaScript we need to put the two of those things together.

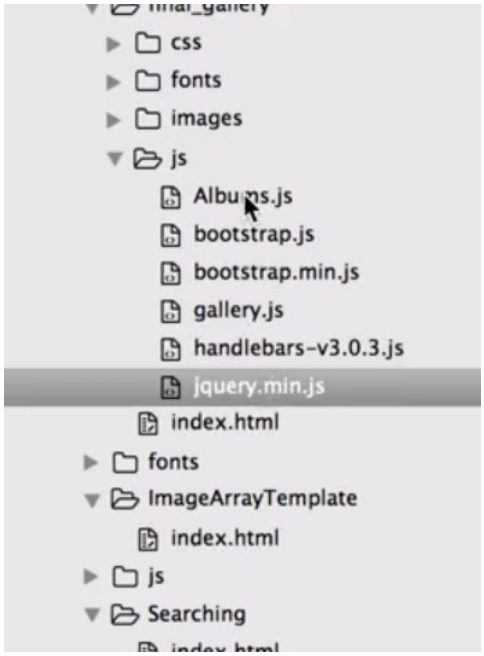
So, we need something a little more complex than the ways we've been using up til now but actually is not that much more complex, all we need is a raise inside a raise. And this is sort of an example of how that content can look.

So we start of with a big array of albums.

- So this big array contains all the albums we have on our site.
- And each of those albums is an object, which has a name,
- but inside that also has another array, which is the photos that that album contains.

So inside this inter-array of photos, again you have a a big array of objects, each with an image source, a title, a description of the kinds of metadata. So we can build quite complex dana out of objects inside arrays, which are inside other objects, which are inside other arrays.

It all sounds quite complex, but as long as you've got in your head the structural website, you can map that out into the structure of the the data itself. I want to show you how I've represented this in code, but before I start I want to show you, because this is starting to be quite a complex website, up to now I've represented everything as a single HTML file, like this index.html. But actually we want to now be able to separate it out so we don't have a really massive index.html.



So here's, I've still got my HTML page, but now inside I've got a JavaScript folder. Which as before has all my JavaScript libraries, my handlebars, jquery, but I'm now created two JavaScript files of my own. Albums, which contains all the content, and gallery, which contains the JavaScript matches between one and the other. I'm gonna go through these in turn.

> I'm gonna show you albums now then look at the HTML code in the index.html, and then go and look at what gallery it is.

Before I do that I'll just show you what else I've done with the structure, I've separated out the CSS, I've separated out the images and the fonts, so we now have a folder structure and set of files that represents all the elements of our webpage. And we can change those separately, and we don't have to deal with really massive HTML files.

And here is the actual albums.js file, which contains all the content, the data, that represents the content of my website.

```
1
2 var gallery = {
3     albums : [
4         {
5             name : "Travels",
6             thumbnail : "images/img_1.jpg",
7             photos : [
8                 {
9                     src : "images/img_1.jpg",
10                    title : "grafitti",
11                    description : "some derelict
12                },
13                {
14                    src : "images/img_6.jpg",
15                    title : "fountain",
16                    description : "a huge dragon
```

And it's basically what I've just shown you. Gallery is a big object. That has one element in there called albums. And this albums is an array, it's got a big array containing lots of objects. And each

object in the array is an entire photo album. It's got a name, it's got a thumbnail. This is the actual image you should display, when you're displaying the whole album. So that's a nice feature that you can choose which image represents the album as a whole. And then you've got all the photos in the album. Each photo in the album is, looks pretty much like the ones we've seen before, it's gotta source image file, it's gotta, a title. I haven't included author because I'm assuming this is a photo album for one particular individual. Instead, I've got a description so you can add a description to each image. And, as we can see, there quite a few photos in here. A real app would have many more. And once we've finished out this one album, we start another album which also has a name, thumbnail in the photos array.

This gives us a fairly complex representation of the data that mirrors the structure of how we think of a photo album. How we've got our app which contains a lot of albums, each album contains photos and each photo has certain properties, particularly the image file. In the next video I'll look how we can take this complex data and create multiple ways of viewing it via handlebar templates.