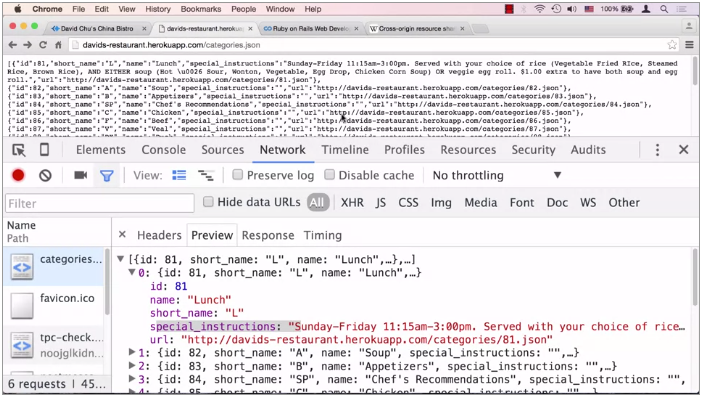
Lecture - 60



In the previous lecture, we dynamically

inserted in the contents on the main page. So there they are right here including

the menu special and the map tiles. In this lecture, we would like to

dynamically insert the menu categories when the user clicks on the menu tile. And actually I already

have this implemented so let's take a look as to what

happens when I click on this button I see that the menu categories

come up with all the pictures. And these things are also clickable. So very similar to before except now,

we actually have all the pictures for the right categories as

opposed to just place holders. So these are dynamically being filled in. Obviously at this point we need the data

to fill up these menu categories and also to point to these pictures

that are specific to each category. So how do we do that? First of all, how do we get the data? Well, there's an app that we've

deployed on herokuapp.com. And the app is called

davids-dash-restauarant.herokuapp.com/ whatever. And basically what that is is a rest API

that provides us JSON data that we need for our application. And even though the actual website is

going to be hosted on GitHub pages, so it's going to be GitHubPages.io or

something like that. The data is actually

going to sit as this app, as this restaurant,

David's restaurant app on Heroku. Now this app is actually

a Ruby on Rails app. And if you want to know more

about Ruby on Rails and how to create these apps all by yourself,

I highly suggest you go to coursera.org. And find the Become a Rails Developer or Ruby On Rails full-stack web

application specialization and this is really this course right here that

you're taking is actually part of that so if you scroll down you can see that you

can get Ruby on Rails introduction. You can go to second course and then we'll

dive a little bit deeper into Ruby and Action Pack and you could get

to MongoDB with Ruby on Rails. And this is actually a recourse

that you're taking right now, HTML, CSS and JavaScript for Web Developers

that is also part of this specialization. Even though you're not required to know

any Ruby on Rails knowledge for this course, we're certainly going to use Ruby

on Rails as our backend to get the data. So I would highly encourage you

to enroll in this specialization. Now the fact that this data is

hosted on herokuapp.com and our application is hosted, well locally

right now for development purposes. It's being hosted on our browser sync so

it's kind of a local host, or eventually we'll get hosted on GitHub. The fact that they're hosted

on two different domain names is actually a problem. And the reason that is a problem is

because every browser comes with this security mechanism that's called

same-source or same-origin rule. And basically, what that means is

is if you load JavaScript that tries to reach out through

Ajax to any other domain name other than the one that this

particular HTML page was served from, the browser will stop that activity and

will not allow it. And the reason it will stop that

is because of the following. The browser and

the JavaScripter runs in it, runs as you has the same privileges

on your computer as you do. And in some remote script

was allowed to reach out to other domain names other than

the one you actually want to. What it could do, since you're

operating behind a firewall, and the JavaScript that's executing in the

page is also executing behind a firewall, because it is your computer after all. If the JavaScript was allowed to reach out

to other domain names other than the one it was served from on the HTML page

that you want it to served from, it could potentially reach

out to internal websites and internal computers that it really

shouldn't be able to even see. So therefore, what the browsers do

is they enforce the same origin and same source rule. They don't allow that to happen. Now in that case how can

we ever get this data? The browser's going to stop us. Well, comes this new

technology called CORS, otherwise known as

Cross Origin Resource Sharing. And if you actually go, you could

Google for it, CORS Wikipedia page and you can see this explained to you

in some detail how this works. Basically the idea of CORS is

there's going to be some HDP headers that the browser will know about and will

tell the browser that it is actually safe to reach out to this

particular domain name and this is exactly what's going on in

the case of this David's Restaurant app. It's actually communicating back to a

browser and sending those headers telling us even though you really shouldn't be

able to reach out with the JavaScript anywhere else you are allowed to reach

out to me, and it is a safe thing to do. So that's point number one. That is how we're going to be able to get

our data from our server that is going to be hosted somewhere other than

Heroku or even just locally through browser sync to this Heroku app

that is going to grab our data. That's point number one. Point number two is this is kind of hard

to read what exactly this JSON is, so one quick trick you could do is that you

could open the Chrome developer tools and if you go to network tab

you can even click all. If you refresh this page right here,

you'll see that you'll get this category that's JSON,

that's the URL we're interested in. So if you click on that, you can go to

the preview tab and Chrome developer tools will be very happy to actually give

you the preview of JSON in a very nice formatted way such as you can open

up each data item and look inside. So you can see these categories

that JSON is serving as JSON that is really an array of objects. So this particular JSON string

is an array of objects. And you can see here each object

has an id, has a name, short name, special instructions for

this particular category. Okay, so now that we know where to get the

data and how we are going to get the data or at least how that is going to be

possible in part two of this lecture, we going to go ahead and execute the code,

the hx code, to pool this data, parse this data and generate the dynamic HTML in order to

insert it as our main content of our page.