**152. Understanding Location Strategies;**

* Instructor: Almost done.
* We covered a lot in this module.
* This has been a very long section of course but it has been so important because routing is such a key task in any application.
* Now, if you have a look at our application, we get a couple of routes in there slash users slash servers and much more.
* Now, here's one important thing.
* *It works fine here on our local setup, but actually this is not something you should take for granted.*
* *If you have a route like this, your domain slash servers and you're hosting this on a real server somewhere in the web, this might not work out of the box because their routes, the URL is always, this is how the web works is always parsed handled by the server first.*
* So by the server which hosts your application.
* *Now here on the local environment in our development environment we're also using a development server but this server has one special configuration*.
* Your real life server also has to have and I will come back to this in the deployment section *the server hosting your Angular single page application has to be configured such that in a case of a 404 error, it returns the index html file So the file, starting and containing your Angular app.*
* **Why? Because as I mentioned, all your URLs are parsed by the server first, so not by Angular, by the server.**
* Now, if you have slash service here it will look for a slash service route on your server on the real server hosting your web app.
* Now, chances are you don't have the route here because you only have one file there.
* Index html containing your Angular app and you want angler to take over and to parse this route but it will never get a chance if you a server the server hosting your app decides no item noted route.
* Here's your 404 error page.
* Therefore, you need to make sure that in such a case your web server returns the index html file.
* If for some reason you can't get us to work or you need to support very old browsers which are not able to parse paths like this in the client, which Angular does, then you have an alternative approach to using these nice URLs which look like old URLs in the web.
* ***You can fall back to our older technique which was used a couple of years ago using a hash sign in your routes.***
* Let's see how we enable it and then I will explain what it does.
* **App-routing-module.ts:**
* *You can enable it in your app routing module where you register your routes.*
* Here with the four route method you can pass a second argument at JavaScript object to this method, configuring the setup of your routes.
* So besides the routes itself, now we can configure them and one important configuration is to *use hash configuration, which you can set to true.*

Text

Description automatically generated

* *The default is false which is why we didn't have to pass it.*
* If we do this and we save that, let's see what happens.

Graphical user interface, text, application, email

Description automatically generated

* You out of the box, see that? Now we have this hashtag in our URL, and if I click on servers, you see there is this hashtag in between and that's no buck that belongs here because ***this is hash mode routing.***

Graphical user interface

Description automatically generated

* **Hash Mode Routing**
* What this hashtag will do is it informs your web server ***"Hey, only care about the part in this URL before this hashtag.***
* " So all the part thereafter will be ignored by your web server.
* Therefore, this will run even on servers which don't return the index html file in case of 404 errors because ***they will only care about the part in front of the hashtag.***
* That's how it works by default and the part after the hashtag can now be parsed by your client by Angular.
* So this is the solution if you can't get the other approach to work.
* However, I will say that definitely you should try to use the more prettier routes using the HTML history mode as it is called with the normal slash routes without the hashtag.
* So this is how you would enable it.
* I'm going to duplicate this so to comment this out to have it there for reference, but I will fall back to the other mode, which gives us cleaner routes which really look a lot nicer and more like you're used to seeing routes from our web apps and this is the last piece of the section.