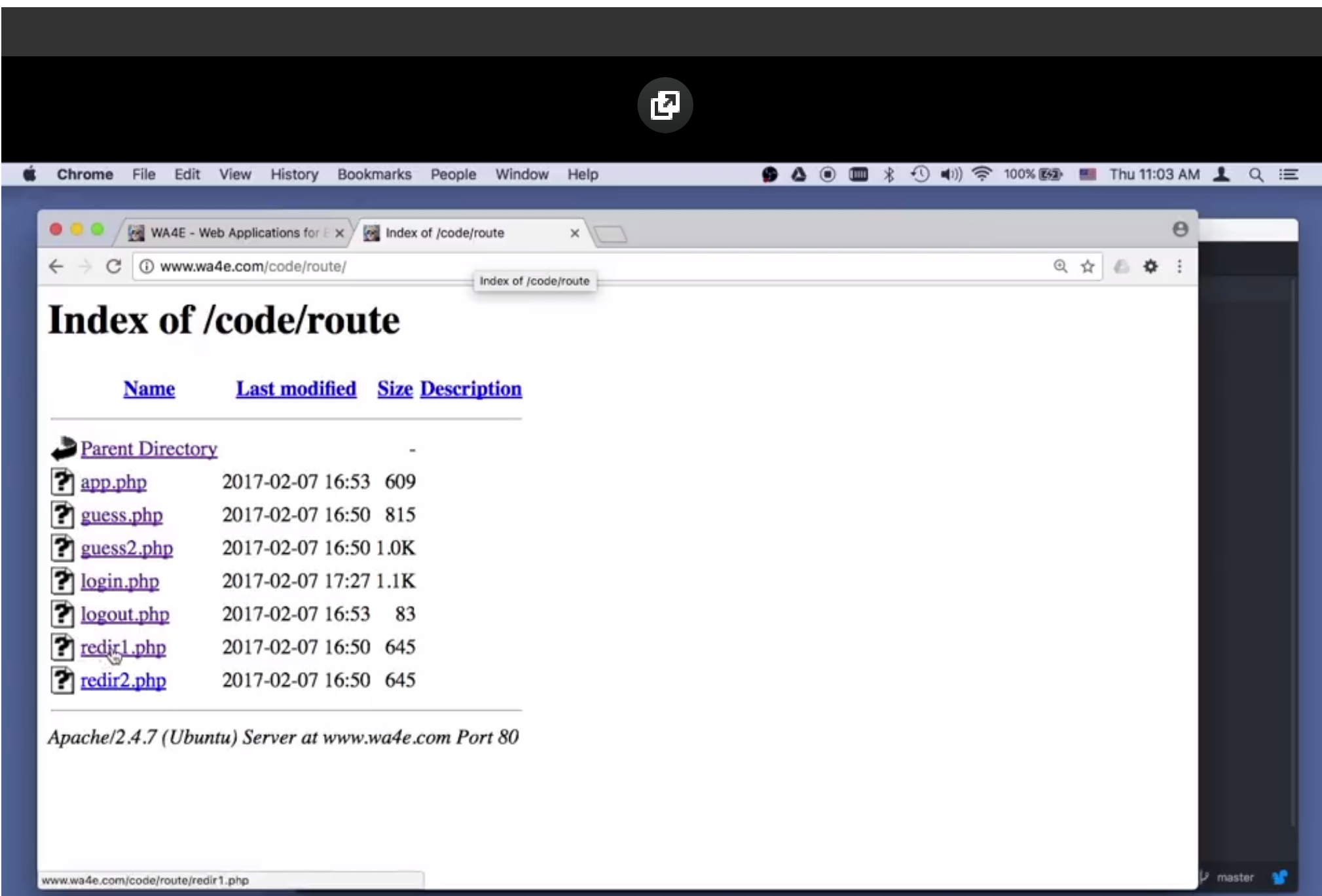


Lecture Content

	Redirect, Routing, and Authentication	9 min
	Code Walkthrough - Routing and Redirect	5 min
	POST / Refresh / Redirect	10 min
	Code Walkthrough - POST / Redirect	6 min
	Implementing Login and Logout	10 min
	Code Walkthrough - Login and Logout (3)	6 min
	Practice Quiz: Routing	11 questions

Assignments

Bonus Materials



And what we're going to do is we're going to

Code Walkthrough - Routing and Redirect

Have a question? Discuss this lecture in the week forums.

Interactive Transcript

English

0:00
Hello everybody and welcome to Web Applications for Everybody. Now we're doing a bit of code walk-through. We're in the section about how things happen with routing. [And what we're going to do is we're going to](#) talk about this redirection code and it's really dumb code. It's really just a POST form that I put in a number and depending on the number, I am going to redirect to different places so. Let's first take a look at the notion of what redirecting is in general.

0:33
All right, and so the location header is basically one of the HTTP headers That you can send, you send a GET and it returns and says 302 Found. And then it says Location: where you're supposed to go. And so just as an example, if you go to www.dr, my normal URL is dr-chuck.com, but I also have drchuck.com with no dash. I got it much later, so what I do is I just automatically redirect it Using that redirect to my dr-chuck.com. So you'll notice the URL changed here. So I switched it from one to the other using this Location header. So you went to the drchuck with no dash and then you were told to go to dr-chuck with a dash and that's a redirect. And that whole notion is there. And so we have ways in PHP

1:27
to set these response headers using the header function. So echo or print sends data to the body of the response and header sends data to the header, the part that comes out first. And so let me do a view developer consoles, we can watch the network go by. So here we go, so let's take a look at redir1.php. I'm doing a session, not really using the session in this particular one, I'll use it in the next one so I probably could get rid of that.

2:02
Okay, it won't have session next time you download that source code, because it does not need a session, it really doesn't, there's no session in it. So if I POST a 1, it's really simple, if I POST a 1 I run the code redirect back to itself, because you can redirect back to the same script. And then I return. Or if I POST 2, I redirect to a location and in this case I add a GET parameter to it, which is redir2.php, which is really just a copy of the exact same thing.

2:34
So, redir2.php is just so that I can sort of be at a different place. And then if I put a 3 in, it completely redirects to some other place, right? dr-chuck.com okay? So I'll put in a 1 here and let's watch the network console. Hit Submit and so this was a POST of the value 1 to this URL. And what it gave me back is a 302 found which is not the same as a 200. A 200 means good news, 302 means, like, I know what's going on but you're at the wrong place, you need to go here. So what happens is immediately, without us even seeing a blink of the eye, the browser stops talking to this URL, and switches to that URL. And then it sends GET request to that URL, and so that's what we actually see. [COUGH] Now I'll refresh this so we only have one thing in there. And if I say 2, you will see that I do a POST back to redir1 because that's where the POST goes, but then redir1 is going to say go somewhere else. And the place I want you to go is redir2.php? parm=123. So then it goes and instantly grabs this zip, it grabs it with a GET request, so that's the screen that we're seeing okay. And so if I type a 3, remember redir1 and redir2 are the exact same code. If I type a 3, it is going to go to location www.dr-chuck.com. So let's put a 3 in clear the network. Put a 3 in and watch what happens, boom. Now we're at dr-chuck.com. So there's all kind of crap that comes out, HTML and CSS and whatever. So let's just look at the first two requests. I scrolled all the way back up. The first one I did a POST to redir2.php with a value of 3 and it responded 302, go away. And it said, here's the place you're supposed to go. So it immediately, my browser immediately saw this and then did a GET request to dr-chuck.com. It of course retrieved all that HTML and then it said, I gotta bring a picture and yadda, yadda, yadda, yadda. So that's all these other 71 requests are the rest of that. The redirect is what happened here from this POST, from this POST with this redirect to this GET, okay. [MUSIC] Okay, so that gives you a quick overview of the redirect and come back, we'll talk about some of the ways that we can use the redirect. [MUSIC]

Downloads

Lecture Video mp4

Subtitles (English) WebVTT

Transcript (English) txt

Would you like to [help us translate](#) the transcript and subtitles into additional languages?