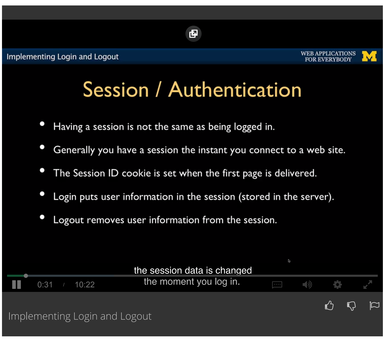
Lecture Content

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



Implementing Login and Logout

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

Interactive Transcript

English

0:08
So now we're going to actually build a fully functional and respectable login and logout. Because we have to use session, and we have to use redirect, and we have to use this concept called a flash message. So to review, sessions are not the same as being logged in. We indicate that we're logged in by sticking a little bit of data in the session. The session starts when you hit the website and the session data is changed the moment you log in. And what we do, is we look at session data, the rest of the application looks at data in the session to decide if you're logged in or not. And then logging out simply removes information from the session. logout.php is my favorite code to write all the time, okay. I guess, I'll show you how this works, right? You send in a POST to login.

0:54
It looks in the data, reads some data to see if you're logged in. If you are logged in, it changes the session that says, yes, that you're logged in. And then we do a Redirect and then we do a GET request and then we come back. And then this code somewhere else in the application, later all the rest of the GET requests or whatever, they check the session and they ask, are you logged in and they look at the session by logging in. The cookie is still there because it's the thing that determines which session is active. But basically, the fact that you're logged in is logged in equals true or something like that, we're not going to do it quite like that. But that little fact that you're logged in is stored in the session, okay. So, we're going to have a couple use cases here. We're going to have an application. And if we go right to this application, it's going to notice that you're not logged in and say, Please Log In. And so then you click on Log In, you're going to go login.php, then you're going to type an account and a password that's wrong, you'll press Log In. And that'll be a POST, that'll do a redirect. And we gotta put this message out. So we gotta POST Redirect. But you already saw in the previous example how we can pass a message into the future using session. But this one's going to be a little different. This one is called flash, meaning, that you only see it once. And then if you type in the right account and password, then you'll come to the application, except now it will know you're logged in. So that's all the scenarios that we're going to work through.

2:19
So let's take a look at the code. Let's first go down to the code that's the view code. So this is the view code, a view code. So, we're going to put that, Please Log In. This is part of the flash message which I'll come back to. Here, we have a form, nothing special, just an account and the password, method="post". You want it to be method="post" because you don't want to be bookmarking a login URLs and putting passwords in GET parameters, okay? So that's our form. Come back to that stuff in a second. So now here's the model part. Here's the model, the top part.

2:57
And so we're going to start the session. The first line, nothing before the first line, and start the session. if we have some POST data, post and password. Then what we're going to do is, we're going to unset $_SESSION["account"]. So we're going to use the fact that the account key is in the session, as our indication whether or not you're logged in or not. But what we're doing is the beginning of a login process is, we're logging out any previous user by deleting it. So unset is the PHP way of deleting a key from an array. So that says, delete the account key from the session array.

3:30
And eventually, we'll put some database and SQL in here in later things. But for now, we're just going to have a hardcoded password. If the password is right, then what we're going to do is copy the account name into the session, because the POST data is going to go away as soon as we redirect. Right, and we're going to have a message. And we're going to call that message success. And we'll call it Logged in. And then we're going to redirect to app.php. Right now, we're in login.php. We're going to redirect to app.php. If, on the other hand, the password is wrong, we're going to put an error message in the session, $_SESSION["error"], A key in the session "Incorrect password",and then we're going to redirect it back to login.php. So the fact that in this case, we're going to redirect to one file. In this case, we're going to redirect back to the same file. This is very much a controller function.

4:15
Routing, controller, deciding where to go next based on the data, based on whether your password is right, I'm deciding to send you a or b. Okay, and we also have return right after the header. So that's the model code for the login.php. Now, let's trace this little bit through here on that previous screen. So we're going to set session error. And, when we drop back into login.php, we're going to say, if it set sessions of error, we're going to print out the session, but we're going to call it red. We're going to print it out in red. And then what we're going to do, this part right here is really important, this is the flash part.

4:57
It means that, once we've shown the user the login error message, we're going to delete it from the session. So if they hit refresh, they won't see it. So a flash is defined as a message that's shown to the user once. It's flashed, and then it's gone. The moment you see it is the moment that you delete it, okay.

5:16
You're going to be really tired of this code by the end of this class, where you're going to be like, you gotta push the flash code in. [LAUGH] Partway through, we just write a function that does this code. You might want to even write a function that does it all the time. Where you just put the error in a session and redirect, and then it comes out. But you then throw the error away, or success. Difference between success and error is you just have the success message green, and you delete it in the same flash pattern, right? And so that's how, if you type the wrong id and password, and you come here, you see this flash message, but then if you hit refresh on this page, it goes and looks like this. This is not there on this page because the very moment that you showed this, you also deleted it from the session, so that's how flash works. POST-Redirect-Flash, right? So POST detects an error, puts the message into session, redirects the GET back to itself, sees it and then deletes it. By the way, we have app.php and login.php, session is shared across all of them, right? So it's also a way to send data from one file to another. It doesn't have to send data, just back to the same file.

6:22
So if we take a look at how this flash works, we get a POST.

6:27
And if something goes wrong, we put the message in the session, we redirect, and then the browser does a GET request. And then in that GET request, it pulls out the session, from the session it pulls the error message out of the session, and then it responds to the GET request. Now with that message unread, but then it also has wiped it out of the session. And now in the future, if you do like a refresh, You see no error message, because that's gone out of the session. The session kind of goes along with all things, but this one deleted that and so now, you don't see it anymore. So, let's look at app.php now. One thing you'll notice about app.php is that it has no model code up here. And that's perfectly okay. because we're not posting to app.php, we're posting the login.php so there's no model. This is all view basically.

7:22
So the first thing it does is, it comes in and it checks to see if there is a success message. If there is a success matches, it prints it out in green and then deletes it.

7:31
That is how we see this little message right here.

7:35
Now, this last little bit is how we check to see if we're logged in. Now remember, we got session start up here. Sorry, there's a session start up here somewhere.

7:46
If there was none, none of this would be working, that's up here.

7:49
If we have the account variable or account key as set in the session, if it's not set, then we tell them that they have to login. If it is set, then we say, welcome to our cool application, and then we put a link into logout.php, so that we can logout. Okay, so this is the success, we go with the good password goes in to login.php. login.php puts the account in the session and it puts the success message in the session. And then it redirects to app.php. And app.php both looks at the logged in message, and prints out the nice little message, and it also checks the account and says, welcome. By knowing that account is there it says welcome. Now later, the flash message is gone but the account stays in session. So the account sort of lives in the session the whole time. So every time we go to app.php, no matter how many times I refresh this, it knows that you stayed logged in. Because it's not a flash message, it's just a piece of the session. And unless you get rid of it explicitly, it just stays there forever, okay. So this application, it doesn't have to just be app.php, it could be any number of files and they just look to see if account is set, and if it is, we assumed that you're logged in.

9:10
Now, my favorite code to write in any application is the logout code, because, [LAUGH] all you do is you start the session, you wipe out all the key value pairs, and then you redirect back to the application. So if you press this Log Out, it actually goes to logout.php, does this code, and then redirects back to app.php. And app.php comes in and checks, is the account set in session? And the answer's no, because you just wiped it all out. So it tells you to log in. So that's how you kind of take the complete step of logging in and logging out.

9:47
Okay, so that's app.php.

9:51
So in this, we sort of have zoomed through the mechanism of redirect, just HTTP concept can redirect anywhere. We talked about redirecting with POST-Redirect-GET. The way you have to, the fact that you can't, you don't want to hit refresh after a POST. We talked about how you can put a message in the session that lasts really only until the next request response cycle. And then how you have long-term, long-lived data that you put into the session to support things like login and logout.

Downloads

Lecture Video .mp4
Subtitles (English) WebVTT
Transcript (English) .txt

Would you like to help us translate the transcript and subtitles into additional languages?