

Lecture Materials

- Cookies7 min
- Sessions10 min
- Sessions Without Cookies4 min
- Code Walkthrough - Cookies and Sessions10 min
- Practice Quiz: Sessions17 questions

Assignment

Bonus Materials

Sessions Without Cookies

WEB APPLICATIONS FOR EVERYBODY

No Cookies for You!

Session is empty

Click This Anchor Tag!

Click This Submit Button!

Our Session ID is: s6ek00I1md7n2v8smpfo7b3lk0

Array ([value] => 0)

nocookie.php

And it's really cool for PHP.

Sessions Without Cookies

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

Interactive Transcript

English

0:09

So I just got done talking with you about how PHP uses cookies and sessions. But there's a really cool feature in PHP that has to do with sessions that don't require cookies. Now it's a kind of a narrow application. Probably 99% of the applications just use sessions. But the applications that I work on personally have to do with teaching and learning.

0:28

So you might have the same application and in one course, you're a teacher and in another course, you're a student. And you might want to have both of those courses open up into two tabs at the same time, whereas there's one cookie for both of those tabs for one website. So if you want to have a different session in one tab than the other tab, you can't use cookies for that and that's why I know and like sessions without cookies.

0:51

So basically most applications, like I said 99% of them, just use cookies and don't worry about it. Just don't think about it, but if you really need to function with an iframe or have more than one session active, away you go. And it's really cool for PHP. So here's a little bit of code and I'll sort of walk you through this. How we can actually make sessions work without cookies. Okay, so here's some magic bits and you can go Google how you do this. But this basically says we're not going to use cookies for our sessions and we're going to transform the output automatically to make it as easy as possible for doing this. We do a session start the same as we ever do, and there's some screen there, and this code is pretty much the same as before where we're going to have sessions of value. If it's empty, we're going to set it to zero. If it's below three, we're going to add one to it. And then if it's above three, we're going to destroy it and restart it. So it's the same, zero, one, two, three, zero, one, two, empty, zero, one, two, three.

1:47

And so that's what it is and we got both a form that we're going to submit and a href that we're going to submit, to click back and forth. So this is how this works. So if we take a look at this, we want to make it so that we're passing this identifier. We still end up with a session identifier. The place that it's stored on disc is the exact same thing. But we need to pass this from one request to another. And so we either have a form right here that we can submit or we can click on the anchor tag. But we have to somehow pass from this request to the next one. And so if we take a look at the view source of this, we can see the two tricks that are happening. So the first thing that happens is we put out href= nocookie.php and then PHP added ?PHPSESSID. So if you click on this anchor tag, that's the click it's going to go to and so the session ID is being passed as a getParameter. So that's how you do it for anchor tags that are going back to the same application. For the form, so we have a hidden field right here that is added by PHP. So if you'll notice, we didn't put this out, added by PHP, which is the session ID. And it's that same ID, the same identifier, that identifier and that identifier. I don't know why it's wrong there, but it's okay.

3:04

That should be the same there. So there's our session, and away we go. So if we hit a form, we send in post data, and session start, whether it's a get or a post, session [SOUND] start [SOUND] looks for those things and away you go, okay? So then what happens is if we click the button, in this case I click the GET request, okay? And we've added one in the session, so it's now one and away you go. And you can see the only kind of compromise that you have is on GET request, you see the session ID. On a POST request, you don't because it's hidden, but there's no way to hide it from a GET request. So that's basically how you do cookieless [SOUND] sessions. And that's just a whoosh by. You can read up on how that works on the Internet. Session ID isn't in everything. It's not automatically added. But basically hrefs in forums, they get added automatically. We'll learn sometimes while we're talking to URLs that are not just hrefs or forms. But basically the URL, we've revealed that and if you like copy and paste the URL, you got other ways to have to protect the session because it's revealed to the user and they might bookmark it. And so if you're going to do these cookieless sessions, there is few more things that you have to learn. Okay, but for 99% of you, we're going to use cookies to do sessions. Sessions and cookies work really well in PHP with a dollar underscore cookie, the set cookie call, the dollar underscore session, and the session start, and the session destroy make things happen really easily. And so up next, we're going to learn a lot more about what we might do with these sessions in PHP.

Downloads

Lecture Video mp4

Subtitles (English) WebVTT

Transcript (English) txt

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