CS 240: cURL for Chess Transcript

[00:00:01] **INSTRUCTOR:** Now, let's see how you could use cURL to develop and debug your chess server. Let's try to call some of the web API functions on your chess server and see what happens. So the first thing I want to do is try to retrieve a list of games in the server by calling the slash game function with a get request.

Start visual description. Text on screen reads:

Curl localhost:8080/game

(“success”:false, “message”:”Error: Invalid auth token”)

End visual description.

[00:00:25] And so if we try to do that, of course I'm running my chess server locally here so I can call it. So if I try to get the list of games, you'll see that it actually fails the first time I try to call it.

[00:00:43] The reason it failed, as you might guess, is that I haven't logged in. So I sent a request to the server that didn't have an auth token in it, and so the server says, well, you haven't logged in, you're not allowed to get a list of the games.

[00:00:56] And so really what we need to do is, or what I need to do is log in. So what I'm going to do there is I'm going to say, okay, let's call the login function. So I'm going to need a post request for that.

[00:01:12] And the URL for logging in is slash session. So we're creating a session, and then I need to send a request body that has the username and password in it. I happen to know there's a username Bob in my server, so I'm going to log in as Bob.

Start visual description. Instructor types in command line: curl -X POST localhost:8080/session -d ‘{“username”: “bob”, “password”:”bob}’ End visual description.

[00:01:40] And so that should do it, hopefully. Okay, so you can see here that the login succeeded, and it returned an auth token. And so I'm going to need that auth token so I can go back and get the list of games like I wanted.

[00:02:00] Let's go back to the game. Like we said, this by default is doing a GET, but I'll just go ahead and put that in there. I need to add an HTTP header to the request, so that should be enough to get back the list of games, and there it gave me back the list of two games that I have in the server.

Start visual description. Instructor types in command line: curl -X GET localhost:8080/game -H "Authorization: " End visual description.

[00:02:38] So you can see that with cURL you can just call your web API functions just like your chest client would. It's a great way to debug it. Now let's try to do some other stuff. Let's see, let's say we could register a user.

[00:03:03] Let's go ahead and create a game. So let's use cURL. I need to use a POST request. And we're going to use the same URL to get the game list, except we're going to call it with a POST. I'm going to create a name, so we have to provide a name for the game.

Start visual description. Instructor types in command line: curl -X POST localhost:8080/game -d '{ "gameName": "speed" }' -H "Authorization: " End visual description.

[00:03:38] I think that's all we need. We're also going to need the HTTP header for the authorization. So you can see I was able to create a game. If we go back and get the list of games, you can see that my new game is in the list.

[00:04:05] So that just gives you a flavor of what you can do with cURL for the chess project. I highly encourage you to play around with curl and get familiar with it and learn how to use it, and it will be very valuable to you as you work on the chess project.