

Lab 2: The guessing game

Overview

To implement a HTTP based number guessing game in Java with sockets. The purpose for this lab is to understand how request look and how they are handled. The player should guess for a number between 1 and 100 and the server needs to keep track of the guesses. The hard part in this lab is for the server to keep track of the users

A very basic working example in PHP (guess.php) is available in the lab folder (in a zip) so that you have something to compare to (this lab has nothing to do with PHP). After unzip it, all you have to do is to move it to your public_html folder on your ubuntu account at school and then go to <http://wproj-vt16.csc.kth.se/~enterYourUsernameHere/guess.php> (try this url if you got problems: <http://wproj-vt16.csc.kth.se/~csc-vahid/guess.php>) from any browser and it should work. Remember that PHP is it's own server side language and can't be used together with your java server.

Specific Requirements

1. The server has to remember the user by saving the SESSION-ID in a cookie.
2. You have to use Java sockets.
3. No javascript.
4. The server has to tell the user if the guess is too high or too low and what the new guess range is, aka "Too high, guess a new number between 50 and 75".
5. The HTML needs to be in its own separate file.
6. Your html must be valid, the easiest way to check this is to use the [WC3 validator](#).

Tips:

1. You can of course use the HTML part of the PHP example for your program but start with getting the server working first.
2. Start with creating a server that just handles one client at a time and then expand from there. It is recommended that you start working from the HttpServer.java code that was shown during the lecture. It should be very helpful with feeding HTML to the client.
3. When using the HTML file you will have to use a delimiter (you can pick any character/s you want) in the HTML so that you in the java code can split the HTML text and insert your dynamic code into the HTML.

Questions:

What is the difference between GET and POST?

What is REST?

What are the other HTTP request methods that are used in REST?

Bonus Assignment X2

You shall now create a simple client that connects to the server to play the game.

Use `java.net.HttpURLConnection` to play the game 100 times and **efficiently** guess the correct number. After that you should present the average number of guesses that was needed (hint, number of guesses needed is lower than 10).

To do this you need to read the `SESSIONID` cookie that is sent from the server and send it with every request.

You are not allowed to make any changes whatsoever to the server to accommodate these requests.

References:

<https://developers.google.com/appengine/docs/java/urifetch/usingjavanet>