

Lab 2: Gissa talet jag tänker på

Översikt

Ni ska implementera ett HTTP baserat gissningsspel i Java, baserat på sockets. Syftet med denna labb är att förstå hur request ser ut och hur de är hanterade. Spelaren ska gissa på ett nummer mellan 1 och 100 och servern behöver hålla koll på hur det går för spelaren. Den svåra biten i den här labben är hur servern ska hålla koll på alla användare.

En väldigt enkelt exempel i PHP (guess.php) finns tillgänglig i en zip så att ni har något att jämföra med (den här labben har inget med PHP att göra). Efter att ni har unzippat den så är allt ni behöver göra att flytta den till er public_html map på er ubuntu-konto på skolan och sedan gå in på <http://wproj.csc.kth.se/~enterYourUsernameHere/guess.php> (try this url if you got problems: <http://wproj.csc.kth.se/~csc-vahid/guess.php>) . Ifall det inte fungerar så kolla på kurs-sidan då nya servrar behövs skapas och konfigureras för deltagare för aktuell kurs varje år. Kom ihåg att PHP är sitt eget server-språk och kan inte kombineras med er Java-server.

Specifika krav

1. Servern kommer ihåg användaren genom att spara en SESSION-ID i en kaka.
2. Du använder dig av rena Java sockets.
3. Ingen javascript.
4. Servern svarar "För högt" eller "För lågt" och promptar användaren att välja ett nytt tal baserat på användarens tidigare gissning., t.ex. "För högt, gissa på ett nytt tal mellan 50 och 75". Där 49 och 76 har varit användarens tidigare gissningar.
5. HTML:en är sparad i en egen fil. Hitta på en egen delimiter för att sätta in era värden.
6. Din HTML måste vara valid. Det enklaste sättet att kolla det är att använda [W3C validator](#) .
7. Du ska se till att detta händer när användaren gissar rätt:
 - a. Sessionen avslutas.
 - b. Användaren får en sida med information om att gissningen var korrekt.
 - c. Antal gissningar användaren gjort.
 - d. Det ska finnas en knapp (länk) så att användaren kan starta en ny omgång.

Tips:

1. Du kan använda dig av HTML-delen från PHP-scriptet för ditt program, men fokusera på att få igång servern först.
2. Starta med att servern hanterar bara en klient åt gången och expandera därifrån. Det rekommenderas att du utgår från koden som finns i föreläsningsanteckningarna (HttpServer.java)

Frågor:

1. Vad är skillnaden mellan GET och POST?
2. Vad är REST?
3. Vad är de andra HTTP metoderna som används i REST?