

Lab 1: Gissa talet jag tänker på

Översikt

I den här labben ska ni implementera ett HTTP-baserat gissningsspel i Java eller Python, baserat på vanliga sockets. Syftet med denna labb är att förstå hur HTTP-kommunikation ser ut och hur den hanteras. Spelaren ska gissa ett nummer mellan 1 och 100 och servern ska därpå hålla koll på hur det går för spelaren. Den svåra biten i den här labben är hur servern ska hålla koll på flera användare simultant.

För att ni ska kunna se vad vi förväntar oss för resultat så har vi en exempelsida här:

<http://wproj.csc.kth.se/~vahid/guess.php>.

Specifika krav

1. Servern kommer ihåg användaren genom att spara ett SESSION-ID i en kaka.
2. Använd Javas [ServerSockets](#) eller Pythons [socket](#), ni har fått exempelkod för att hjälpa er komma igång i ert gits-15 repo. För Python har vi skrivit en wrapper då den råa socketen är lite bökig att ha att göra med.
3. Ingen javascript.
4. Vid en gissning ska servern svara "För högt" eller "För lågt" och informerar användaren att välja ett nytt tal baserat på vad användaren gissat på tidigare, t.ex. "För högt, gissa på ett nytt tal mellan 50 och 75".
5. Ni ska spara HTML:en i egen fil **guess.html** som läses av er server som med hjälp av ett unikt påhittat ankare sätter in de värden som ändras mellan gissningar. Se exempelvis [moustache](#) för inspiration (notera dock att ni inte får använda ett bibliotek).
6. HTML:en måste vara valid. Enklaste sättet att kolla det är att använda [WC3 validator](#).
7. När användaren gissar rätt ska följande ske:
 - a. Sessionen avslutas, vilket inkluderar att sessionskakan invalideras.
 - b. Användaren får en sida med information om att gissningen var korrekt med antal gjorda gissningar.
 - c. Det ska finnas en knapp (länk) så att användaren kan starta en ny omgång.

Tips:

1. Du kan använda dig av HTML:en från vår exempelsida genom att högerklicka och välja "Visa källkod", fokusera dock på att få igång servern först. För att hjälpa er så har vi gett er skelett i Java och Python.
2. Starta med att servern hanterar bara en klient åt gången och expandera därifrån.

Frågor:

Vad är skillnaden mellan GET och POST?

Nämn de andra 3 mest använda HTTP metoderna.

Vad är REST? Vad ställer det för krav på HTTP-metoderna?

Bonusuppgift X1

Lös uppgiften i ett annat språk. Antingen det språket du inte valde ovan eller något helt annat.