

## Øving Intro HTTP

Tomas Holt, Institutt for informatikk og e-læring ved NTNU

Opphavsrett: Tomas Holt og Stiftelsen TISIP

Lærestoffet er utviklet for faget IFUD1042 Applikasjonsutvikling for mobile enheter

---

### Oppgave

Oppgaven går ut på å lage et enkelt spill. Det skal være laget slik at brukeren skal tippe et tall, i et gitt tallområde. Brukeren skal få tre forsøk. Hvis ikke det er gjettet riktig innen det 3 forsøket (dvs. man må tippe riktig på et av de tre først), må man begynne på nytt. Det er en web-tjener som styrer tippingen. Klienten skal sende tallet til tjeneren som så vil returnere om tallet er riktig eller ikke. Om du er interessert i å se tjenerkoden (tallspill.jsp) så kan du laste ned denne fra samme lokasjon som denne øvingen (du trenger ikke å kikke på tjenerkoden for å løse øvingen).

Din oppgave blir å programmere Android-klienten. Før brukeren får spille må han oppgi navn og kortnummer (tanken er at vi skal kunne trekke noen kroner for å la folk spille, men kortnummer burde selvsagt vært kryptert...). Disse opplysningene skal sendes til tjeneren som vil ta vare på opplysningene. Parametrene (som sendes til tjeneren) må hete hhv. *navn* og *kortnummer*!

Når dette er gjort vil det komme en streng fra tjeneren. Dette er enten "Oppgi et tall mellom x og y!" eller "Feil, du må registrer navn og kortnummer før du kan tippe (via start nytt spill)". I førstnevnte tilfelle kan brukeren nå få tippe et tall mellom x og y som er bestemt av tjeneren. Navnet på parameteren som sendes til tjeneren må være *tall*! Hver gang brukeren tipper så vil han få en av disse beskjedene fra tjeneren:

- Tallet <tall> er for lite!Tallet <tall> for stort!
- <Navn> du har vunnet <beløp> som kommer inn på ditt kort <kortnummer>

Alt som står i <> er varierende strenger, avhengig av hva klienten har oppgitt.

Under ser du hvilke skjermbilder jeg har brukt for å lage applikasjonen. Merk at all tekst bortsett fra det første skjermbildet er strengen som kommer fra tjeneren.

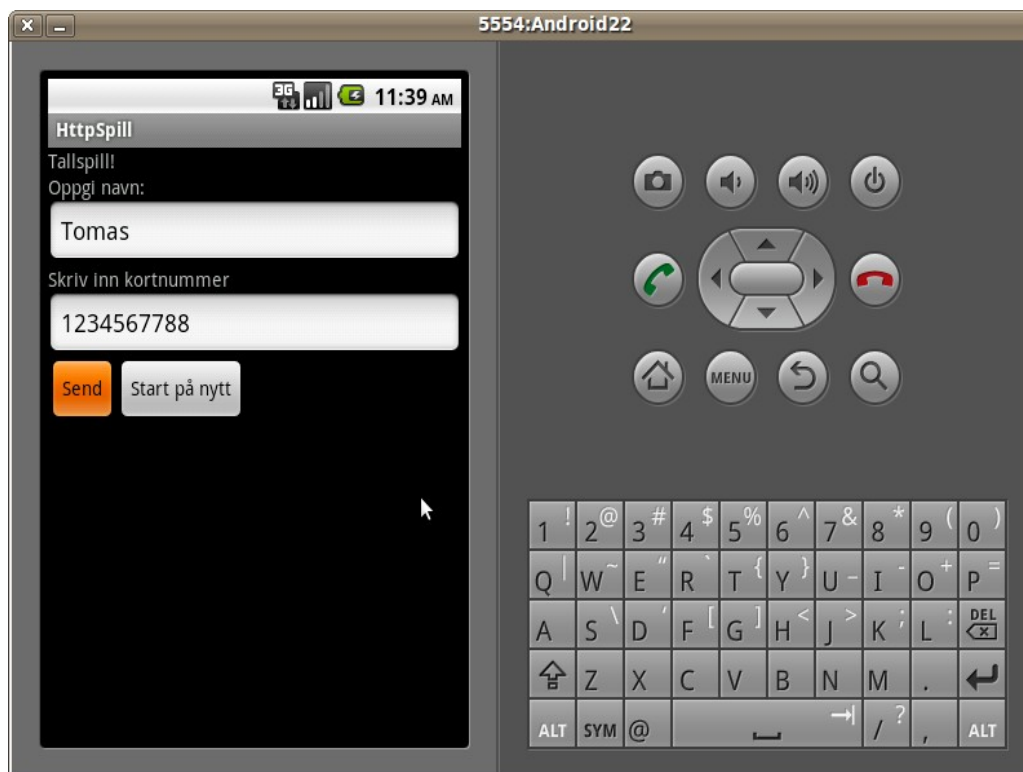


Illustration 1: Oppstartbilde, nytt spill.

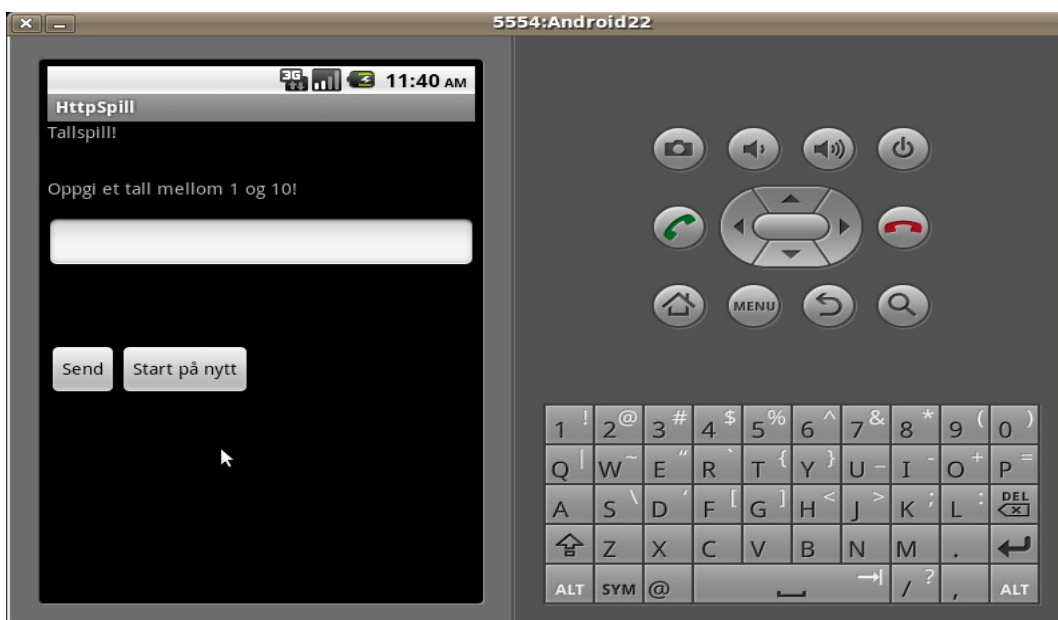


Illustration 2: Sender det første tallet, tjener angir hva som er gyldig tallområde.

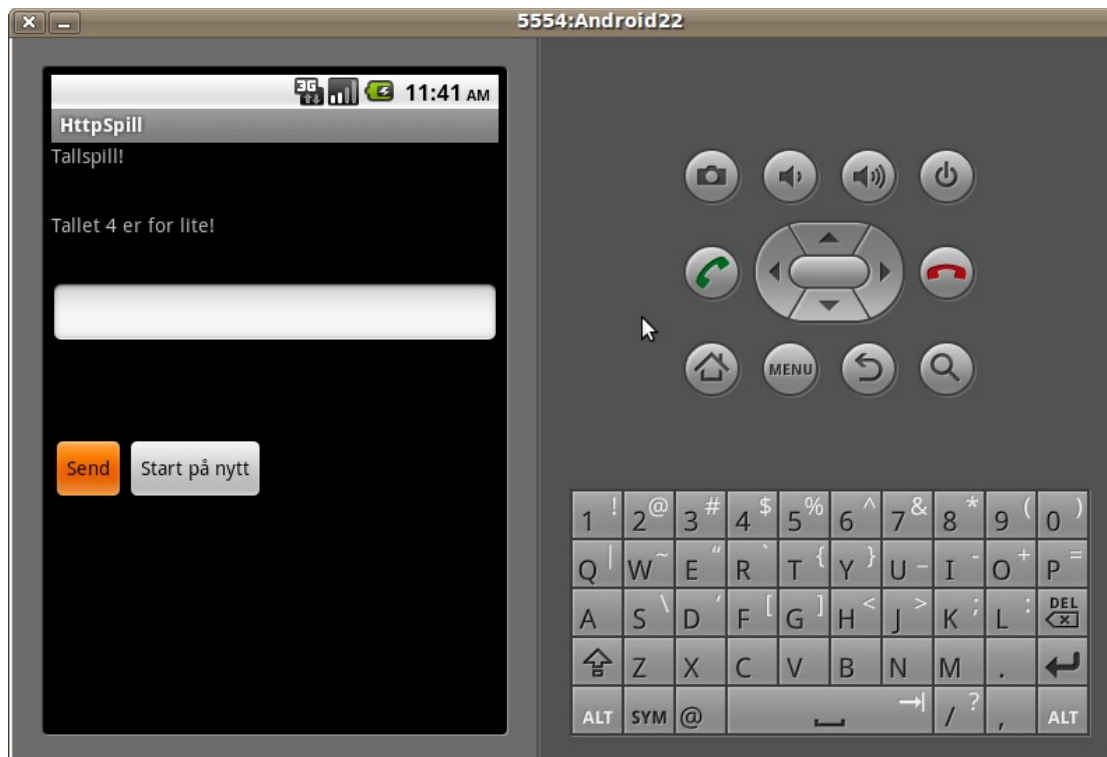


Illustration 3: Får beskjed om at tallet 4 er for lite

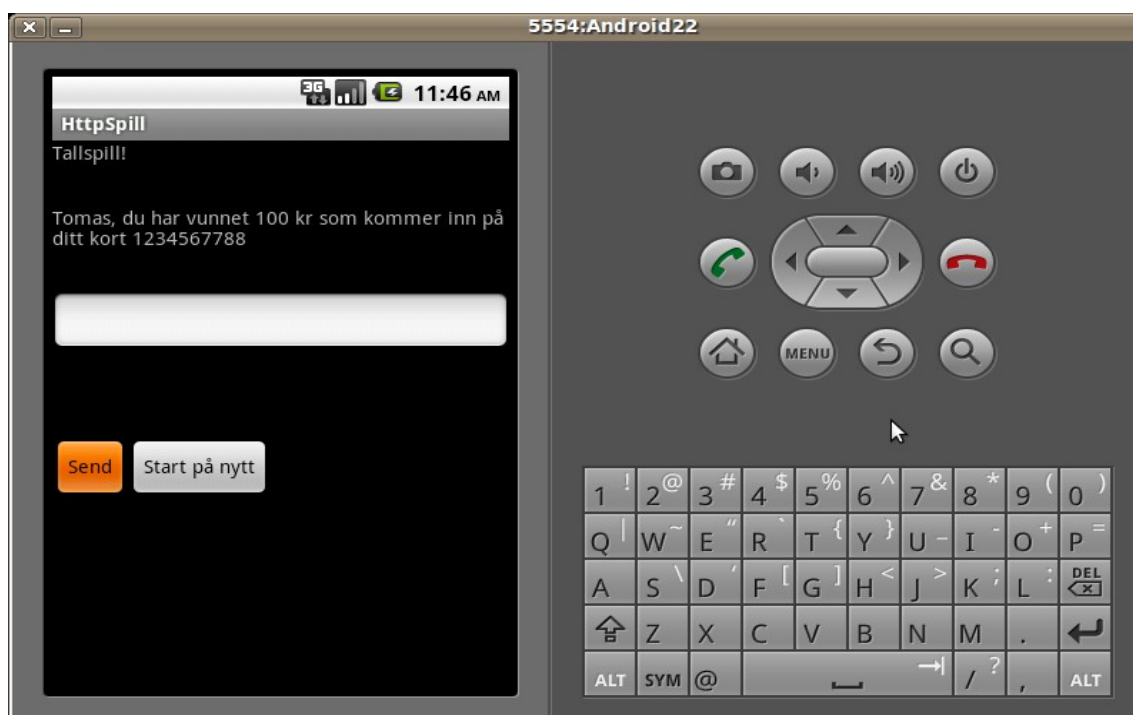
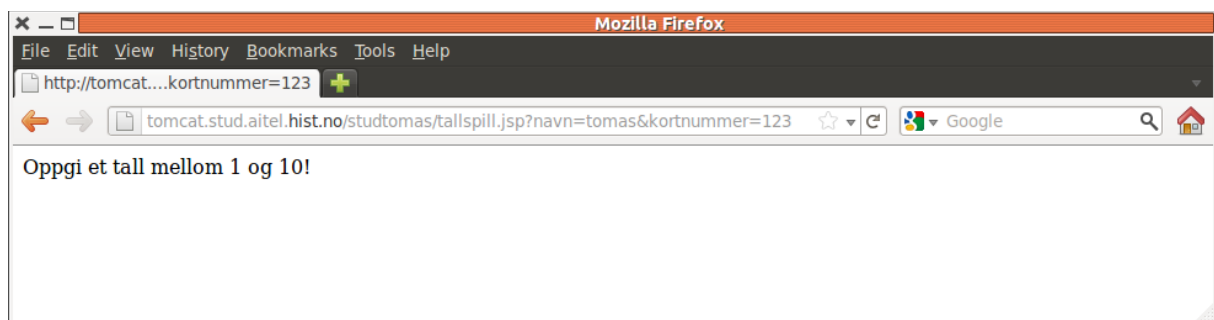


Illustration 4: Til slutt treffer vi riktig tall og vinner!!

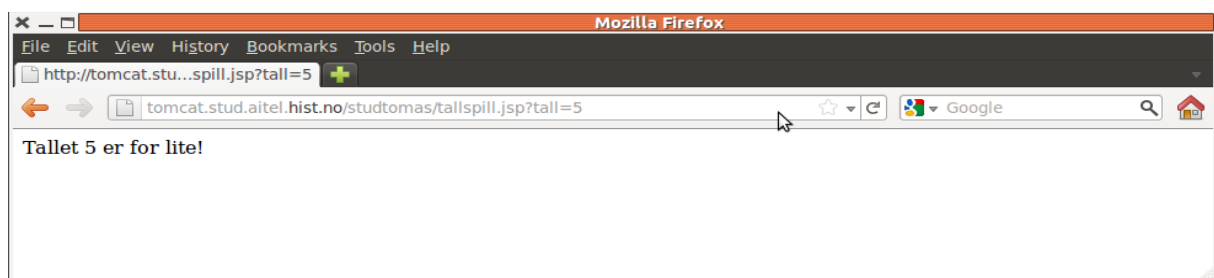
Ting å tenke på når du lager applikasjonen:

- Husk at tjeneren må kunne huske tilstanden til klienten (ellers vet den ikke hvor mange ganger klienten har tippet, eller hvilket navn/kortnummer som ble oppgitt i starten).
- Navnet til en person kan inneholde mellomrom og særnorske tegn (du må gjøre noe for å ta høyde for dette).
- Selve nettverkskommunikasjonen trenger ikke å foregå i en egen tråd i denne øvingen.
- Adressen til tjeneren (JSPen som styrer spillet) er `http://tomcat.stud.iie.ntnu.no/studtomas/tallspill.jsp`
- Sørg for å skrive ut mest mulig til loggen, slik at du vet hva som foregår. Spesielt er det nyttig å vite hva som sendes ut på nettverket og hva som kommer tilbake fra tjeneren.
- Fancy grensesnitt er ikke noe poeng i denne øvingen (få det til å fungere). Du trenger heller ikke å gjøre noe sjekk på data som brukeren skriver inn. Tjeneren gjør noen enkle sjekker.

Husk at det også er mulig å test løsningen med en web-leser. Du må da selv sørge for å fylle inn parametrene i adressefeltet. Dette kan se slik ut:



Her kan du se at jeg har sendt parametrene `navn=tomas` og `kortnummer=123` til tjeneren. Tjeneren returnerer så at jeg må oppgi et tall mellom 1 og 10. Ved å endre innholdet i adressefeltet til å sende parameteren `tall=5` får jeg nå følgende respons fra tjeneren: Tallet 5 er for stort (se under).



Hvis jeg fortsetter vil jeg etter hvert få beskjed om at jeg har brukt opp antall sjanser eller at jeg har tippet riktig. Merk at for at dette skal fungere så må nettleseren din akseptere cookies.

### En liten kommentar til øvingen (skal ikke implementeres i øvingen):

Om man ønsker å lage denne typen løsninger «in real life» så vil selvsagt kryptering (bruk HTTPS i stedet for HTTP) og bruk av XML- eller JSON-data også være meget aktuelt. Mer informasjon kan f.eks. finnes på <https://developer.android.com/training/articles/security-ssl.html> og <http://developer.android.com/training/basics/network-ops/xml.html>