

# Oblig4 del1 – Spring MVC

*Sist oppdatert av Lars-Petter Helland 12.10.2024*

For å unngå opphoping av arbeid i slutten av semesteret, har vi delt opp Oblig4 i 2 innleveringer (del1 og del2). Dette er oppgaveteksten for del1.

Del2 vil være en fortsettelse/utvidelse av denne, og inneholde ting som databasetilgang, innlogging, tilgangskontroll, sikker passord-håndtering, og (forhåpentligvis) deployment.



**Innleveringsfrist for Oblig4 del1 er fredag 25. oktober.** Vi har som mål å rette innleveringene og godkjenne innen 2 uker etter frist.



Øvingen skal gjennomføres i grupper på **2-4** studenter.

**NB!** Det er nytt gruppesett for hver innlevering.  
Gruppesettet for Oblig4 er «**DAT108 Oblig4 gruppe x**».



**Innleveringen er en zip i Canvas.**

Zip-en skal hete **Oblig4\_gr17.zip** for gruppe 17 osv. (dere forstår) ... Denne skal inneholde:

1. **Et pdf-dokument** med:
  - opplisting av hvem som er med i gruppen
  - eksempelskjernbilder fra kjøring
2. **Et Maven Spring Boot-prosjekt** med løsninger på programmeringsoppgavene.

## Ressurser til øvingen

Siden DAT108 ikke primært er et HTML-kurs, har vi samlet HTML-koden fra skjermbildene i oppgaveteksten pluss et lite stilsett i **oblig4-del1-maler.zip**. Dere slipper derfor å bruke tid på å gjette dere frem til hvilken HTML-kode som ligger bak web-sidene i oppgaveteksten.

## Oppgaven: Web-applikasjon Påmelding til fest

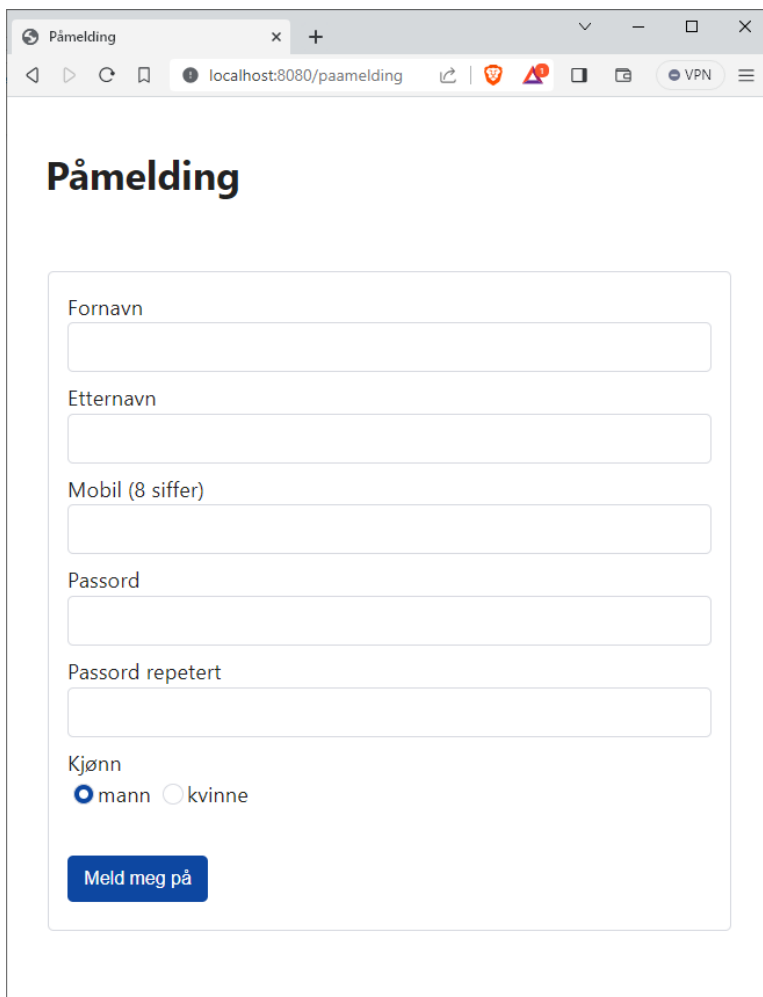
Det skal lages en enkel web-applikasjon som kan brukes til påmelding til fest.

Her er brukstilfellene i grove trekk:

1. En bruker skal kunne melde seg på festen.
2. En bruker skal kunne se listen over påmeldte. (I del2 vil det være krav om innlogging)

### Påmelding

Skjema for påmelding skal se ca. slik ut:



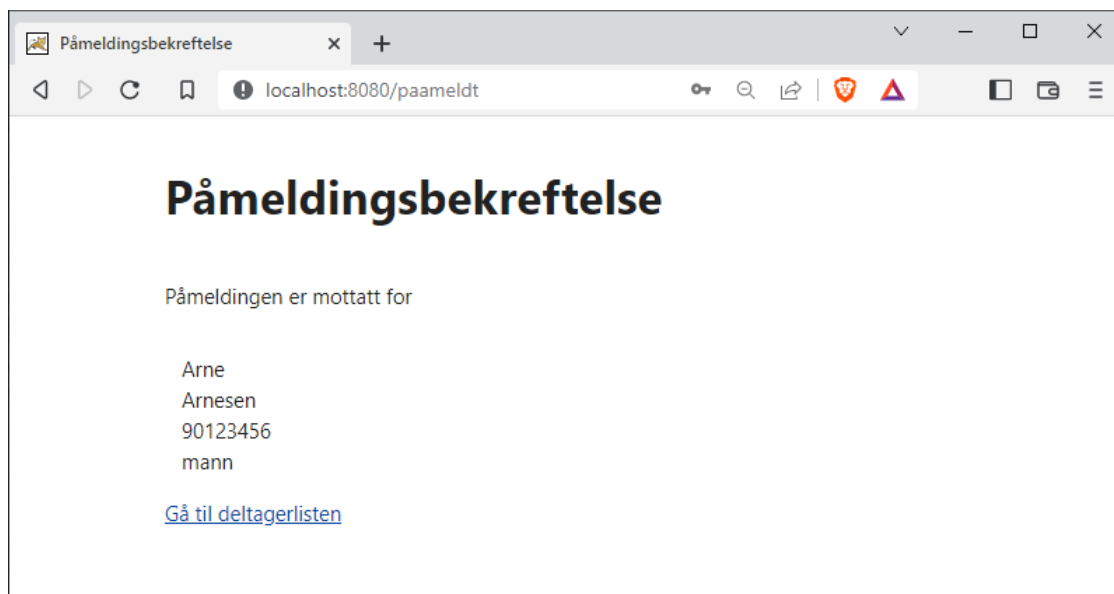
The screenshot shows a web browser window with the title 'Påmelding' and the URL 'localhost:8080/paamelding'. The form contains the following fields and controls:

- Fornavn: Text input field.
- Etternavn: Text input field.
- Mobil (8 siffer): Text input field.
- Passord: Text input field.
- Passord repetert: Text input field.
- Kjønn: Radio buttons for 'mann' (selected) and 'kvinne'.
- Meld meg på: Blue button.

Alle felter er obligatoriske, og bør valideres på klientsiden (i nettleser) via HTML/CSS/JavaScript. Det er opp til dere hvor mye dere gjør ut av dette. Poenget er å gi god hjelp til bruker ved utfylling av skjemaet + sikre at skjemaet helst ikke kan sendes av gårde før visse minimumskrav er oppnådd.

All brukerinput SKAL også valideres på tjeneren. Mer detaljer om valideringsregler senere i oppgaveteksten.

Hvis alt er OK får man en bekreftelse på at påmeldingen er registrert, slik:



### Se deltagerlisten

Når man trykker på lenken "Gå til deltagerlisten" kommer man til deltagerlisten (denne vil ha litt ekstra innhold i del2):

Kjønn	Navn	Mobil
♀	Anne Panne	234 56 789
♂	Arne Arnesen	901 23 456
♂	Lars-Petter Helland	123 45 679
♂	Per Persen	123 45 678
♂	Per Persen	123 45 675
♂	Per Viskelær	345 34 534
♀	Xx-x Xxx	123 21 378

Deltagerlisten skal være sortert stigende på fornavn, deretter etternavn.

## Validering av input ved påmelding av ny deltager

Validering av input ved påmelding av ny deltager skal gjøres både i nettleser/klient og på tjener.

**Validering på tjeneren** (Java/Spring) er for å sikre at dataene som mottas og lagres er gyldige. Her må man også f.eks. sjekke om en deltager med gitt mobil (unik id) allerede er påmeldt.

TIPS: Manuell testing av validering på tjeneren kan f.eks. gjøres ved å strippe HTML-en litt og kommentere ut JavaScript-et under testing. Testing kan også gjøres med JUnit, se senere.

## Valideringsregler

Regler for gyldig brukerinput ved validering på tjenersiden:

- **Fornavn** skal være 2-20 tegn og kan inneholde bokstaver (inkl. æøåÆØÅ), bindestrek og mellomrom. Første tegn skal være en stor bokstav.
- **Etternavn** skal være 2-20 tegn og kan inneholde bokstaver (inkl. æøåÆØÅ) og bindestrek (IKKE mellomrom). Første tegn skal være en stor bokstav.
- **Mobil** skal være eksakt 8 siffer, ingenting annet. Et tilleggskrav ved påmelding er at mobilnummeret IKKE må tilhøre en allerede påmeldt deltager. Alle mobilnumre i deltagerlisten skal være unike!
- **Passord** bør ha en viss minimumslengde. Krav utover dette bestemmer du selv.
- **Repetert passord** må være likt passordet.
- **Kjønn** må være "mann" eller "kvinne".

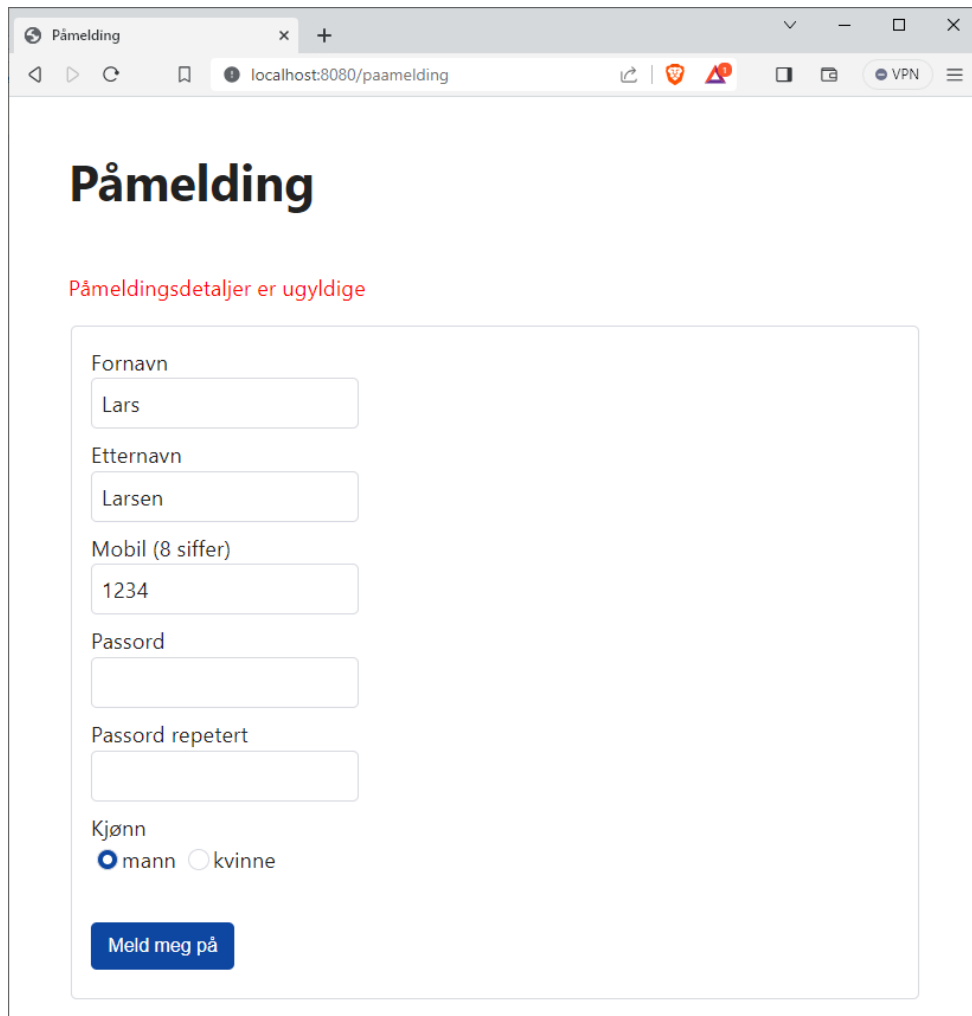
## Validering av brukerinput og feilmeldinger på klientsiden

Som tidligere nevnt kan dere selv velge hvordan og hvor mye dere legger i dette.

## Validering av brukerinput og feilmeldinger på tjenersiden

Mens validering i nettleseren handler mest om brukervennlighet, handler validering på tjeneren mest om sikkerhet og robusthet. Man kan i teorien tenke seg at data er sendt inn som har omgått valideringen i nettleseren.

Ved ugyldige data ved påmelding skal påmeldingsskjemaet presenteres på nytt med en enkel feilmelding, f.eks. "Deltager med dette mobilnummeret er allerede påmeldt" og (for andre feil) "Påmeldingsdetaljer er ugyldige" (se nedenfor).



The screenshot shows a web browser window with the title "Påmelding" and the address bar displaying "localhost:8080/paamelding". The page content includes a large heading "Påmelding" and a red error message "Påmeldingsdetaljer er ugyldige". Below the error message is a registration form with the following fields and options:

- Fornavn: Text input containing "Lars"
- Etternavn: Text input containing "Larsen"
- Mobil (8 siffer): Text input containing "1234"
- Passord: Text input (empty)
- Passord repetert: Text input (empty)
- Kjønn: Radio buttons for "mann" (selected) and "kvinne"
- A blue button labeled "Meld meg på"

Dere kan velge å gi mer spesifikke feilmeldinger om dere ønsker det, men det er ikke et krav.

Tanken er at nettleser/klient skal gjøre en komplett validering med hjelp til bruker, og at ugyldige data i requesten da kan antas å skyldes "hacking".

## Arkitekturkrav

Spring MVC og Spring Boot skal brukes som rammeverk.

### MVC, EL og JSTL

ALLE forespørsler inkl. "redirects" skal gå til Controllere.

All HTML skal genereres fra JSP-sider! Bruk JSP Expression Language (EL) og JSP Standard Tag Library (JSTL).

For å sikre mot direkte forespørsler til JSP-sider skal alle disse plasseres under mappen /webapp/WEB-INF, der de er utilgjengelig mot direkte forespørsler fra webleser.

### PRG

GET/POST skal brukes korrekt.

Hvis forespørselen er en POST skal det ikke gjøres en direkte videreformidling til view, men i stedet en "redirect" til controlleren for dette viewet (Post-Redirect-Get).

Redirect bør også gjøres ved GET når man skal omdirigeres til en "annen" side, f.eks. ved ugyldige data ved registrering.

### Lagring av påmeldte deltagere

Data i applikasjonen er kun de påmeldte deltagerne. Her er nok det mest nærliggende å kun ha én klasse, **Deltager**.

```
public class Deltager {  
    private String mobil;  
    private String passord;  
    private String fornavn;  
    private String etternavn;  
    private String kjonn;  
    ...  
}
```

Legg inn 5-10 hardkodete deltagere i en List<Deltager> i applikasjonen for å simulere at vi har registrerte deltagere.

(I del2 skal data lagres i en database, og det blir også krav til korrekt og trygg lagring av brukernes passord.)

## Testkrav

Innleveringen SKAL inneholde litt **enhetstesting** med JUnit, minimum 5 ulike "asserts" om hvordan ulike ting i applikasjonen er forventet å virke.

Ting man kan teste kan f.eks. være inputvalideringen ved påmelding (deklarativ / programmatisk)

## Krav til deployment

**Nei, ikke i del1.** Det er tilstrekkelig at applikasjonen kjøres på egen maskin, og at virkemåten dokumenteres i innlevert pdf via kopier av skjermbilder.

Lykke til!

Lars-Petter