# Mesaninen – Spring

## Oppsett

### Felles

Sjekk ut kode herfra: git clone <https://github.com/netcompanyno/springKurs>

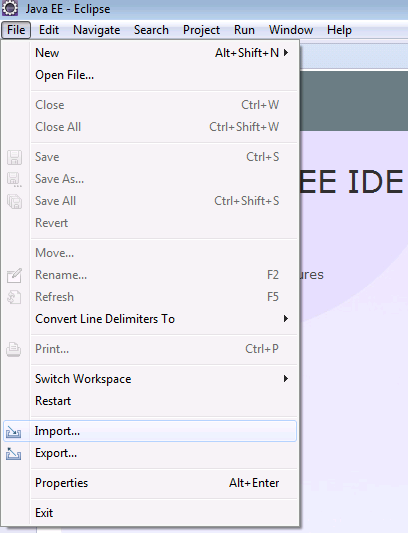
Sjekk ut riktig branch i henhold til IDE:

* git checkout eclipse
* git checkout idea

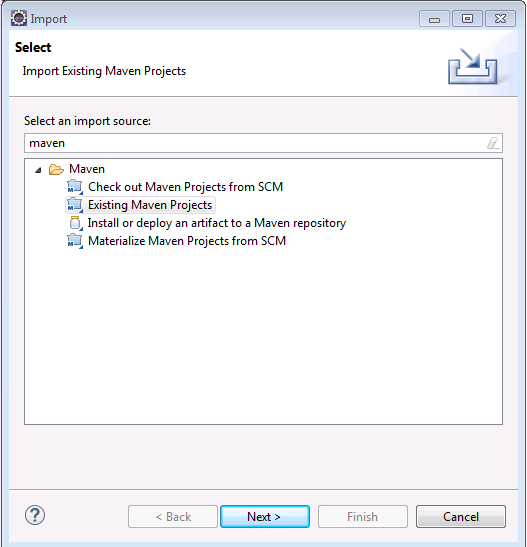
Bygg med maven: mvn clean install -DskipTests

### Eclipse

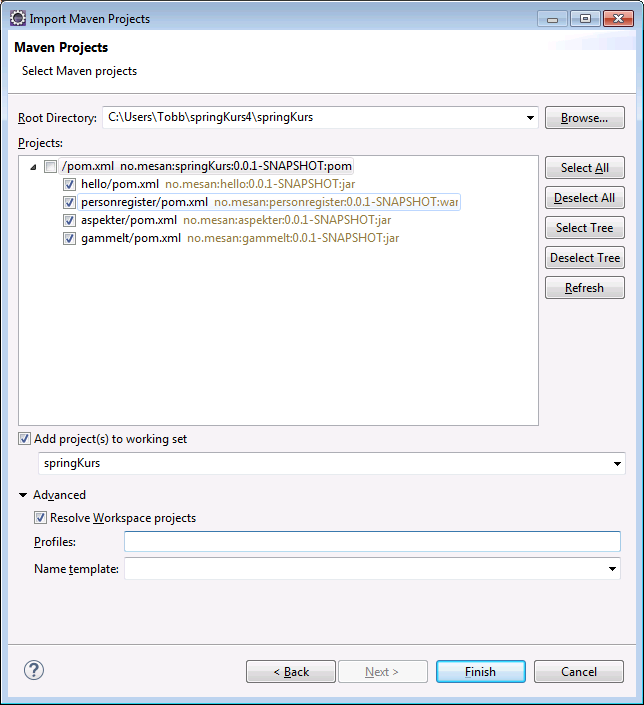
1. Velg File -> Import..



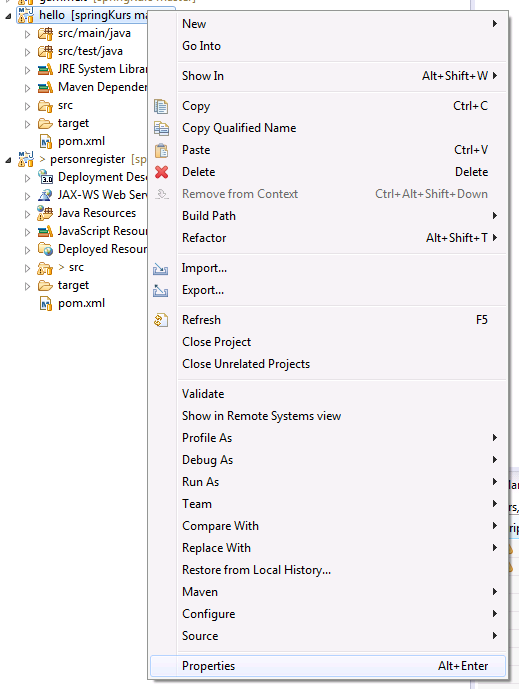
1. Velg Maven -> Existing Maven Projects



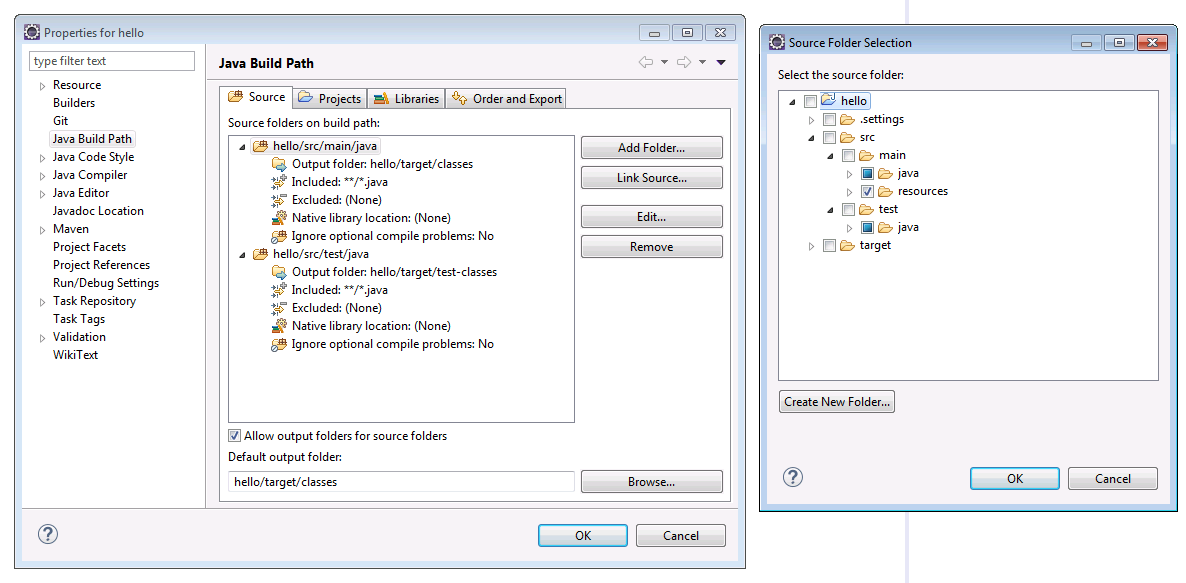
1. Finn springKurs-mappen, fjern avhuking av modul springKurs



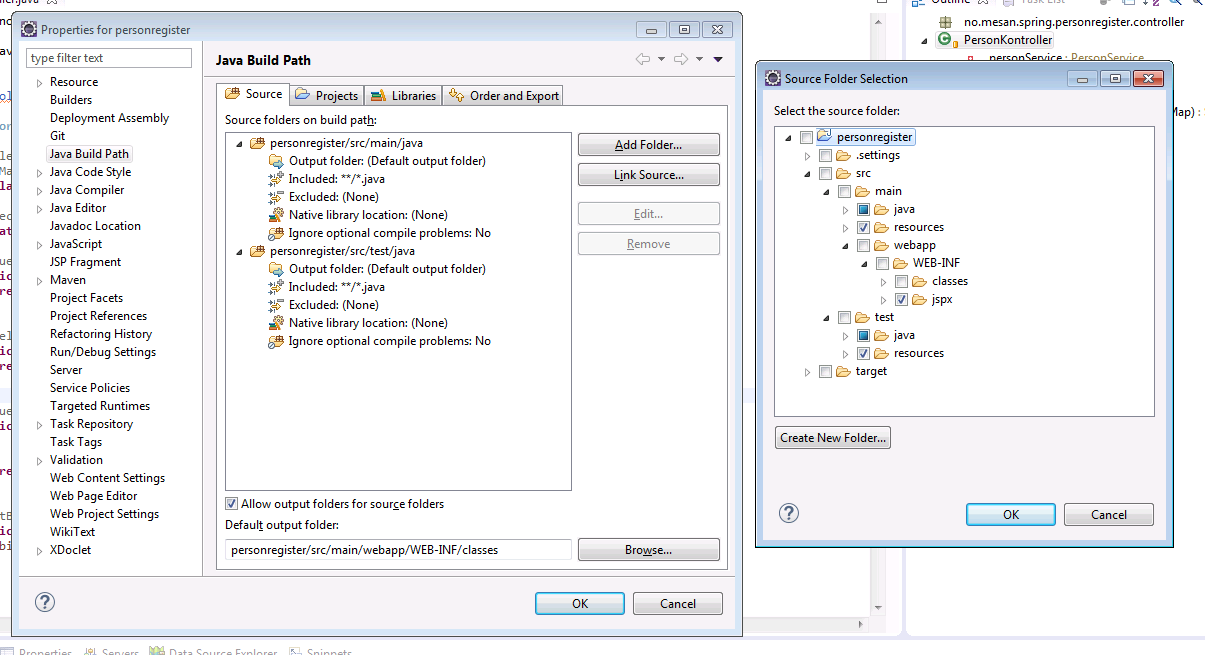
1. Høyreklikk på modulen «Hello» og velg properties



1. Velg Java build path -> source -> add folder og huk av for src\main\resources-mappen



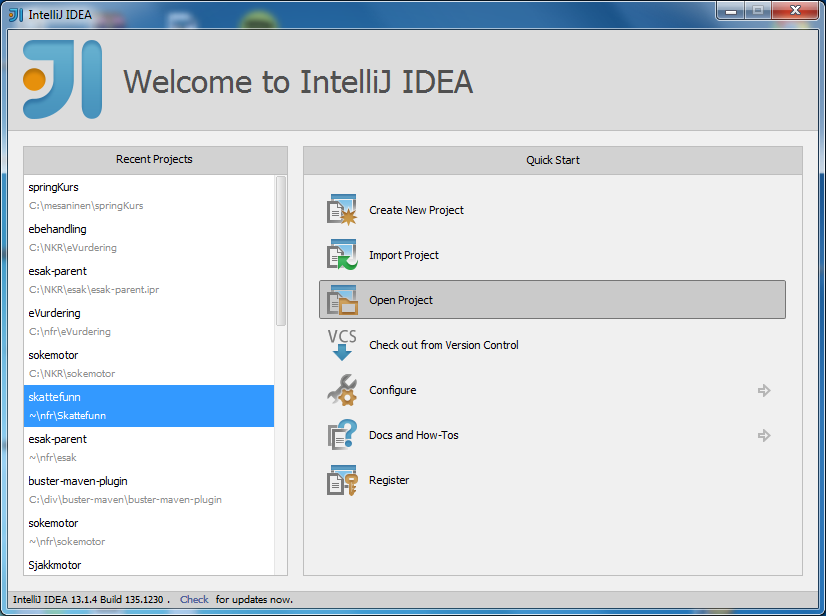
1. Gjenta det samme for modulen «Aspekter»
2. For modulen «Personregister», legg til mapper som vist på bildet under



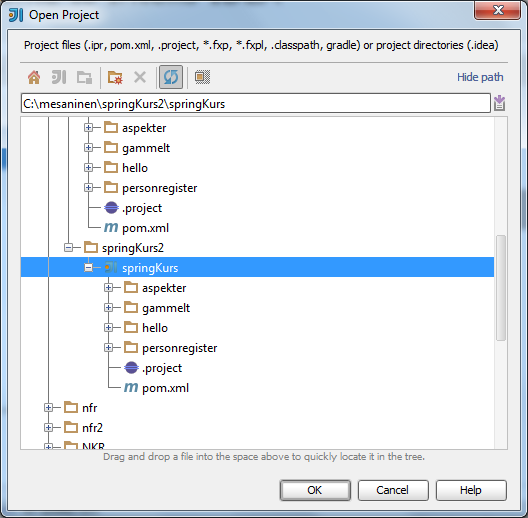
Si fra hvis dere opplever trøbbel med encoding!

### IDEA

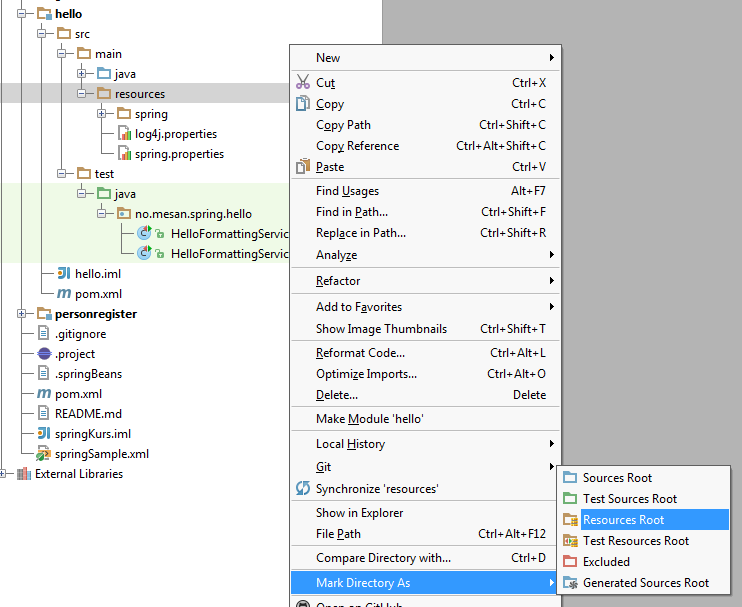
1. Velg Open project



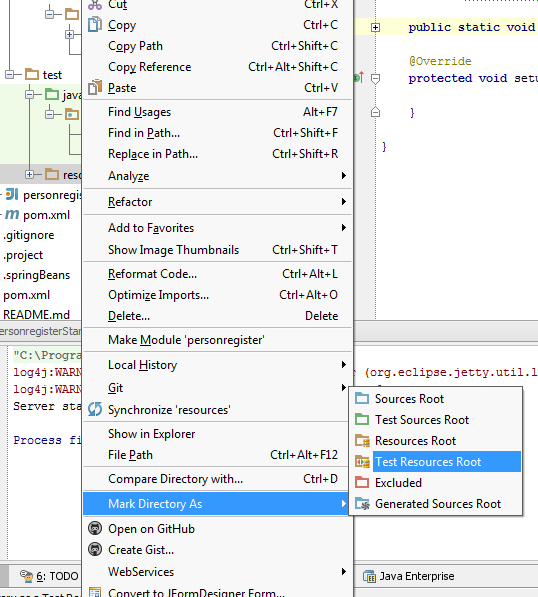
1. Velg mappen springKurs (eller pom.xml innenfor springKurs)



1. (Om nødvendig) Høyreklikk på src\main\resources i modul «Hello» og velg mark as -> resources root



1. (Om nødvendig) Gjenta for modulene «Aspekter» og «Personregister»
2. (Om nødvendig) I modul «Personregister», høyreklikk på mappen src\test\resources og velg Mark Directory As -> Test Resources Root



## Oppgaver

### Oppgave 1 – Dependency Injection (xml):

Få en og en test i HelloFormattingServiceXmlTest til å kjøre grønt. Testene kjører ikke grønt samtidig.

### Oppgave 2 – Dependency Injection (annotasjon):

Få en og en test i HelloFormattingServiceAnnotasjonTest til å kjøre grønt. Testene kjører ikke grønt samtidig.

### Oppgave 3 – Dependency Injection (lazy og primary)

Få testen EnServiceTest til å kjøre grønt ved å kun benytte @Lazy og @Primary. Ingen andre kodeendringer tillatt! (Oppgaven er også et eksempel på config uten xml, ta gjerne en titt )

### Oppgave 4 – Full stack

Personregisteret har blitt rammet av et hissig virus som har spist opp all spring-configen. Få personregisteret til å kjøre igjen via klassen PersonregisterStarter.

Klassene i personregisteret skal kobles sammen slik:

* Bønne for PersonRepositoryDummy
* Bønne for PersonServiceImpl
* Inject av PersonRepositoryDummy inn i PersonServiceImpl
* (Inject av PersonServiceImpl inn i kontrollere, allerede gjort)

Kjør PersonregisterStarter for å sjekke om det fungerer (ingen stacktrace lover godt.)

Besøk Personregister på <http://localhost:8080/personregister>

1. Konfigurer med kun xml. Ingen annotasjoner tillatt!
2. Omkonfigurer med bruk av @Inject på setter/konstruktør. Ingen <property> eller <constructor-arg> tillatt!
3. Omkonfigurer med bruk av @Inject på felter. Ingen settere/konstruktører tillatt!
4. Omkonfigurer med bruk av @Component el. Ingen <bean>-elementer tillatt!
5. Bruk klassen Konfigurasjon og lag bønner med @Bean. Ingen component-scan! (krever endring i web.xml, se TODOs)
6. Bruk klassen Konfigurasjon og @ComponentScan. Ingen @Bean-definisjoner.
7. Vi vil ikke bruke dummy-repository lenger. Bytt ut med PersonRepositoryJPA uten å endre dummy-repo (skal ha @Component/@Repository).Hint: @Primary.
8. Nå var det dårlig med data.. Legg til en @PostConstruct-metode i PersonServiceImpl som oppretter og lagrer noen personer
   1. Tips: For å få lov til å lagre må man være innenfor en transaksjon!
   2. Tips: Spring må få beskjed om å lete etter @Transactional-annotasjoner
9. Legg til @Transactional på alle metoder i PersonRepositoryJPA
   1. Husk readOnly=true på de som skal ha det!

### Oppgave 5 - Aspektorientering:

Få testene i ApestrekerTest til å kjøre grønt.

### Oppgave 6 – Spring MVC

Hvis det blir tomt for oppgaver, fortsett arbeid med Personregister og gjør:

1. Utvid personregister med mulighet for å opprette og oppdatere personer.
2. Legg til utsending av endringsmeldinger vha Spring JmsTemplate ved opprettelse eller oppdatering av personer