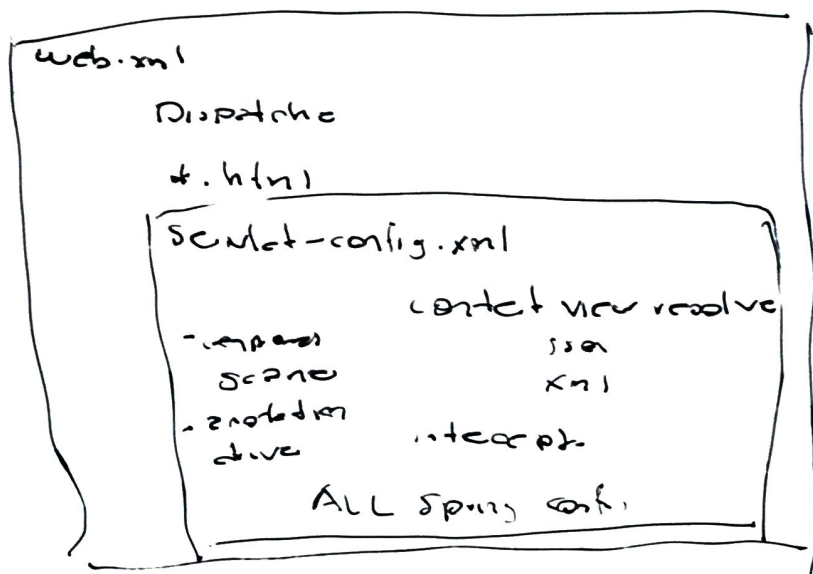


Spring with JPA and hibernate

Por que usa spring y hibernate

- Es mas facil probarlo
- Es mas facil usar transacciones o son transparentes
- La configuracion del data source



Configure the application server in IntelliJ  
(tomcat)

Run

Edit configuration / click upper left  
tomcat / local

+ More especial. tip

cuida aradimos un dependencia  
ve transitive dependencias por  
ve si no debe escribir algun  
version por otra anterior

Persister.xml ~ Empty

data source

allowed operations

caching

etc

Por que usar spring entones?

Podemos sobrescribir por ambiente

persister.xml debe de estar en resources

META-INF

---

Una vez que tenemos lo anterior vamos  
al web.xml y creamos

context-param

listener

Esto hace que cuando inicie nuestra aplicacion  
el listener ve a llama al contexto y  
este llamara al archivo xml que esta  
definido en el contexto.

---

¿por que Context es donde ponemos lo que  
hay en el persister.xml pero fíjate  
que se así para que lo maneje  
spring

---

Transaction manager es hecho por spring

Por que usa JPA con hibernate o viceversa

Por que si usalo con JPA podemos cambiar de implementacion, es como una interfaz JPA y clases concreta hibernate, toplink

---

Que es un entity

② Entity > Un pozo con estas anotaciones  
② Id

---

Que es el dialecto

Es el tipo de sql que pertenece del vendedor que estamos usando

---

Cuales son los posibles valores de h2ddl auto

create  
create-drop  
update

Validate → Para probar que todo este alineado entre objetos y clases, se usa para verificar por ejemplo en produccion  
none

---

② Generated Value (Sequence, auto, identity)

② Table es name

② Id → primary key

Create-drop solo hacer tablas que tenga  
objetos estado a tablas, es decir solo  
hacer tablas referenciadas por objetos

---

Optionals

② Table

② Column

---

② Persistence Context → Atado a un persistence  
unit, es por eso que  
no se llama entity  
manager, porque solo  
representa a un  
persistence unit.

Invoca el entity  
manager en nuestro  
codigo, recordemos

que un entity manager a traves de un persistence  
unit

---

② Persistence Unit

Private Entity Manager va en el repository  
layer

---

JpaContext, em, todos  
persistenceContext → solo cuando controlamos  
servicio-config = solo cuando controlamos

En capa service tenemos ② transaction en  
el metodo

con jpa tenemos que hacer flush() para  
que persiste

Spring 3 necesita salir 2

Spring 4 necesita salir 8

② OneToMany → un objeto a una lista de objetos

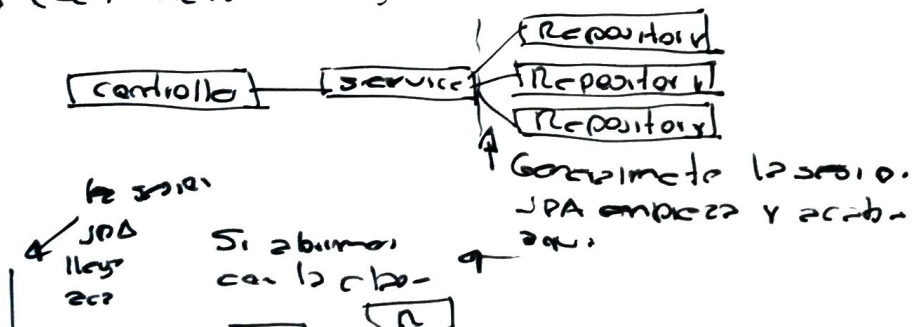
② OneToMany (mappedBy ("goal", cascade...  
(list < Goal > ...  
mappedBy significa que le pertenece al objeto  
"goal" → esta debe existir de lo otro lado  
para hacerla bidireccional

Fetch : 122v → cuando llamamos a la propiedad  
eager → cuando usamos el objeto

Cuando tenemos 122v loading, marcaría un  
error si intentamos cargar el recurso  
por que como es 122v necesita un  
señal para cargar, por ejemplo clic de  
un boton, para evitar eso tenemos que  
extender un poco mas la sesion de spa,  
esto se hace por medio de

"Open Entity Manager in View Filter"

lo cual hace lo siguiente



## Projection JP41

5

- Necesitamos un constructor con los campos que requiramos y los joins necesarios, por ejemplo.

```
Select new com.pluralsight.GoalReport (
    g.minutes,
    e.minutes,
    e.activity) from Goals, Exercise e
where g.id = e.id;
```

Es como crear vista, en el caso para reports

## Named Queries

Se pone en el dominio que le corresponde  
Carga primera cuando se este levantando  
el servidor.

Usamos named parameter  
son tipados

## UPDATE

mejs no acepta flush

mejs requery objeto

posible de requery nada

\* Si tenemos un controller, le podemos agregar  
a HttpSession sin romper la firma, por lo  
que lo permite spring