

- Criteria is good for: SimpleController + Multiple fields

Criteria = many criterions

- Restrictions is CriterionFactory to create Criterion

Criteria criteria = session.createCriteria(Product.class);

crit.add(Restrictions.[factoryMethod\(\)](#));

- criteria could add : Criterion, order, projections, logicalExpression, sqlRestriction, example, disjoint

- Crit.add(Restrictions.isNull("proName"));

Equals to:

FROM Product WHERE proName is null;

- Pagination: Url rewriting to pass pageNum

crit.setFirtsResult(1) //equals to set offset

Limit numperpage

offset (pagenum-1)\*numperpage

e.g.

Limit 10

Offset 14\*10

15页, 每页10条记录

- Formbackingobj: check previous first then leave empty if there is no previous value, Usually used to update sth at detail
- Bootstrap is client side ,so could not be used in spring form
- REST obj: Representational state transfer

Consuming a service from other web app

- 4 ways to query in hibernate

Query q = session.createQuery(hql);

Criteria c = session.createCriteria();

SQLQuery s = session.createSQL(Native)Query(sql);

ProcedureCall p = session.createStoredProcedure(sp);

- Threadlocal used for session obj to synchronize, but not sessionFactory.  
SessionFactory is thread safe already.  
1 db instance 1 sessionFactory.

- To avoid injection. We use “?”. But it is difficult to change for it is index based. So we give “?” a name, that is, q.setString(“un”, “xinrong”);
- uniqueResult //one or exception or null  
distinct // one of records in resultset

•

@NamedQueries({

    @NamedQuery(name = “user.findAll”,query =“from User u”),

...

)

Defined inside polo, and used in controller

to get: session.getNamedQuery()

•

hqlKeywords not case sensitive

but Pojo and attribute is case sensitive

## Final

- au·then·ti·ca·tion/ AUTHORIZATION (Interceptors, httpfilters, spring security..)
  - Cross cutting task: common in each controller, so we handle req before reach controller
  - No sql query,
  - Sanitization
  - Complete functions
  - Technologies Make Sense
  - ~~DAO new method (previously Threadlocal to make sure hsession Threadsafe get save update delete obj)~~
- 

- Could use hibernate console plugin to check hql before run it

- .setComment("search hql: "+ hql);
- 

HQL —> POJO

If in PersistentState,

all properties, collections, and entities will be available

If session is closed (you navigate to View page),

all properties will be available

collections—>will not be available (LazyInitializationException)

entities—> will not be available (LazyInitializationException)

So we use CACHING Technology

- EHCache (OSCache, JBossCache, SwarmCache)
- Guava
- MemCached
- JCS (Apache Java Caching System)
- Redis
- use Java.io.File (you have to delete files after some time)

-

Spring XML

```
<bean id="udao" class="UserDAO"/>
```

In Controller

There are 3 ways to get UserDAO

1. Use New (Not recommended as this will violate DI, IoC)

```
UserDAO udao = new UserDAO();
```

2. Use getBean() method, so that Spring Container will inject this instance into the controller

```
UserDAO udao = ctx.getBean("udao");
```

3. Use AutoWired

```
@AutoWired
```

```
UserDAO udao;
```

- The easiest way to create .xls in browser:

Html tbl

Set mimetype = excel

(Excel and word but not pdf know how to render html tbl)

```
- if (reportType.equals()) {
    response.setContentType
    response.setHeader
}
```

- How to deploy app

Web project settings set context root

Hosting or AWSaccount to deploy app

Export .war and unzip it: Eclipse folder - javaee folder

Ask Tomcat to point to another folder not web apps:<Context path="/img"

docBase="C:/IMG"/>inside <Host>... in server.xml

```

```

That is no using actual address.

用 Context 来覆盖 Host 的 appBase 的设定而不是直接修改 appBase。

- ModelAndView Pass multiple objs:

request.setAttribute

pass one map

- velocity views - alternative of jsp

For designers, there is no java code

Velocity

\$Expressions = references

#Statements = directives

## comment

Quiet reference notation – put nothing if not exist

HOWEVER,

\$result = ...

if ... not null, will use previous value

So we set "" each time

To use Velocity:

Import jar

```
<bean id="viewResolver" class="...  
    <property name="suffix" value=".vm"/> //mav return .vm  
</>
```

- `<a href="/view-welcome.htm"></>`

`http://host:port/view-welcome.htm` //root folder

BUT

In mapping `/view-welcome`

`http://host:port/app/view-welcome.htm`

- File Upload:  
any file type  
2 requirements: `methodPost`; `multipart`;  
`addDependencyMultipartResolver`

`<form:form command="user" enctype="">` //spring form default post

`<input type="file" name="resume"/>`

————to save in db

```
- Class Applicant {
    @Column(type=blob)
    CommonsMultipartFile resume; //dataBinder will bind attr to file
}

- Class Applicant {
    //do not save it - transient
    @Transient
    CommonsMultipartFile resume; //in the memory -> file system
    @Column
    String filename;
}
```

•

Ways to use AJAX:

-plain js

-jQuery notation

`$.post...`

-DirectWebRemoting - easy AJAX

-AngularJS

•

Alternative Views - velocity, pdf, excel, jsp...

- Di - **annotation**. `@Autowired` //create an instance and inject there  
**-bean wiring**. `<bean id="userdao" class="">`  
`(UserDao)getContext.getBean`

one Dao per pojo, extends superDao  
 //static - sharing with everyone  
 public class userDao extends DAO{  
     public void save(user){  
         getSession.save(user);  
     }  
 }

DAO

USERDAO EXTENDS DAO

USERCONTROLLER {Di USERDAO}

- ThreadLocal: 是一个关于创建线程局部变量的类。通常情况下，我们创建的变量是可以被任何一个线程访问并修改的。而使用 **ThreadLocal** 创建的变量只能被当前线程访问，其他线程则无法访问和修改。

- JPA ANNOTATION

```
@Entity
@Table(name="userTable")
Public class User
{
    @id
    @GeneratedValue(strategy=GenerationType.Auto )
    private int userid;

    @column
    private String name;
}
```

- **Bi-directional** needs 1 owning side only

```
Class Phone{
    @manytomany (mappedBy ="phones")
```



```

        set customer c;
    }
    Class Customer{
        @manytomany
        set phone p;
    }

```

Nonowning side must use **mappedBy**, or else infinite pointing.  
 We define which is nonowning

- In cfg  
 add <mapping class=""/>. //pojo

Dml

ddl

Ddl - definition schema - **tell hib to create table for us**

Dcl

- 

Cfg file -

```

    <property name="hibernate.hbm2ddl.auto">create</property>
    <property name="show_sql">true</property> //see query or use
queryAnalyzer

```

- Annotation: @ManyToOne(mappedBy = 'person'... (bi-directional)  
XML: Inverse = true
- 

- 3 ways to make a controller:
    - implements Interface Controller
    - Implementation (e.g. abstractController)
    - annotations
- 

- ModelAndView is requestScope
- 

- (LazyInitializationException) because 2 reasons: not fetch yet| if session close, keep primitive but no collections and pojos in lazy loading [hsession's scope is local method, outside the method, detached status  
//transient never saved  
//only dml need transc  
//(shared) pool be global , the others no  
2 way to fix it:
  - set lazy to false -> eager loading(Keep all in memory) in hibernate.cfg.file define
  - reattach it - put in extra requestScope
  - 2nd level cache (opt), before disconnect, put it 2nd level file sys, its providers : EH,OS,swarn,JBoss (entity, coll)  
//EhcacheHibernate part of Hibernate Api(no need to import)  
//first level cache is session cache  
//speed up dynamic web app – put in cache while persistent and retrieve after detached

- Inheritance of POJO

Inheritance:

e.g.

```
class Person
```

```
{
```

```
    name;
```

```
}
```

```
class Professor extends Person
```

```
{
```

```
    dept;  
}
```

4 techniques to fool db to store inheritance:

- - 1 Table per concrete class with implicit *polymorphism* (*diff forms depending on how you define them, you need Inheritance| ref to superclass*) **repeated+noRefOfFather** Use when **modification of superclass in the future is unlikely**
- - Table per concrete class (UNION)
- - Table per class hierarchy (ONE TABLE FOR FATHER)(x for denormalization + some fields nullable ) (2+1 col, 1 is discriminator **nullableShouldAllowed + Denormalized**)
- - Table per subclass (FK POINT TO FATHER)(**BEST**)  
<joint-subclass>