

spring在MVC层解决JPA的缓迟加载问题 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/182/2021_2022_spring_E5_9C_A8M_c104_182496.htm 作为EJB3.0的一部分，JPA是一个好东西。其简单的配置方式及强大的默认配置支持，使其可以轻松自由的存在于轻量与重量之间，如果现在您的JavaEE项目，不管是选择轻量级构架还是重量级构架，如果持久层不选择使用JPA，而是用一些ORM框架（如Hibernate、TopLink）的专用API，那么在将来的某一天一定会为这个选择而说出至尊宝那句“假如上天再给我一个机会...”的至理名言。下面是一个简单的Entity，是对一个CMS系统中，关于树状信息目录实体类的定义，包括了一些详细的映射的配置信息

```
@Entitypublic class NewsDir ...{ @Id @GeneratedValue(strategy = GenerationType.TABLE) private Long id;// 主键
@Column(unique = true, nullable = false, length = 16) private String sn;// 目录编号 private String title. // 目录名称
@OneToMany(mappedBy = "parent", cascade = javax.persistence.CascadeType.REMOVE) private List children = new java.util.ArrayList();// 下级目录 @ManyToOne private NewsDir parent;// 父级目录...}
```

当然，跟任何其它优秀的技术一样，JPA也不是完美的，在使用的过程中难免都会出这样那样的问题，这就需要我们程序员具有格物致知的本领，在应用中灵活应付这些问题。这里例举一个缓迟加载的问题，以上面的新闻目录Entity为例。对于parent与children这个一对多的双向关联，为了提高系统效率，children默认使用的是缓迟加载的方式。在一些轻量级的构架中，由于脱离了J2EE容器

及事务支持，经常会出现Entity脱离了Persistence Context，变成了detach或EntityManager关闭，导致一些我们预想中的一些功能无法正常运行。最常见的就是在使用MVC框架的时候，在表示层无法加载需要缓迟加载的数据。比如，在一个基于EasyJWeb的mvc应用中，action中的方法如下：public Page doList(WebForm form, Module module) ...{ NewsDirQueryObject ndqo = new NewsDirQueryObject(). form.toPo(ndqo).

ndqo.setDel(true). IPageList pageList = service.queryDirsByConditions(ndqo).

CommUtilForTeaec.saveIPageList2WebForm(pageList, form).

form.setResult("dirPath", this.getDirPath(form)). return module.findPage("list"). }在模板文件中有如下内容

：#foreach(\$info in \${dir.children})目录名称：\${info.title}#end关于业务逻辑层Bean的配置： expression="execution(* com.easyjf.cms.service.*.*(..))" /> pointcut-ref="CmsManage" /> transaction-manager="transactionManager"> read-only="true" /> read-only="true" />

class="com.easyjf.cms.service.impl.CmsManageServiceImpl"> 在这里，当mvc层执行到\$info.getChildren()方法的时候，将会用到缓迟加载，由于Spring的事务是配置在service层的，因此在执行service.queryDirsByConditions方法完成后就关闭了事务。因此运行程序就会出现类似下面的错误信息:2007-03-28

00:39:35,750 ERROR [org.hibernate.LazyInitializationException] - failed to lazily initialize a collection of role:

com.easyjf.cms.domain.NewsDir.children, no session or session was closedorg.hibernate.LazyInitializationException: failed to lazily

initialize a collection of role:

com.easyjf.cms.domain.NewsDir.children, no session or session was closed at

org.hibernate.collection.AbstractPersistentCollection.throwLazyInitializationException(AbstractPersistentCollection.java:358) at
org.hibernate.collection.AbstractPersistentCollection.throwLazyInitializationExceptionIfNotConnected(AbstractPersistentCollection.java:350) at

org.hibernate.collection.AbstractPersistentCollection.readSize(AbstractPersistentCollection.java:97) 使用其它的mvc如struts

、webwork乃至spring mvc都会有这样的问题，问题的核心是在事务启动及结束上，由于我们都习惯于在service层而非mvc配置及使用事务，导致了这样的问题。解决的办法其实很简单，就是把事务的启动放到mvc层，让mvc层的controller来开启事务，而让业务层的方法加入的事务中。比如，

在EasyJWeb中，可以通过如下的配置来实现现在action中开启事务：在Spring配置文件中配置EasyJWeb的核心处理器，并把process方法添加到事务中，配置文件如下：

```
expression="execution(*  
com.easyjf.web.RequestProcessor.process(..))" />  
pointcut-ref="easyjwebProcessor" />
```

transaction-manager="transactionManager"> 只需要这样简单的配置，你会惊奇的发现，所有缓迟加载及其它由Persistence Context无效而引起的问题均解决了。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com