如何GORM对你来说还不够灵活强大，那么也可以直接使用Hibernate来做对象映射。

映射可以使用XML文件或者JPA Annotation。这对于面对历史遗留的数据库表结构的项目特别有用。最重要的是，直接使用Hibernate做OR Mapping，GORM提供给domain class的动态finder和动态持久化特性仍然保留！

# 使用Hibernate XML Mapping文件

只需要在grails-app/conf/hibernate目录下创建hibernate.cfg.xml文件即可给domain对象做OR Mapping。可以使用Grails命令：grails create-hibernate-cfg-xml创建hibernate.cfg.xml文件，文件内容如下：

|  |
| --- |
| <?xml version='1.0' encoding='UTF-8'?>  <!DOCTYPE hibernate-configuration PUBLIC  "-//Hibernate/Hibernate Configuration DTD 3.0//EN"  "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">  <hibernate-configuration>  <session-factory>  <!-- Example mapping file inclusion -->  <mapping resource="org.example.Book.hbm.xml"/>  …  </session-factory>  </hibernate-configuration> |

各个映射文件例如org.example.Book.hbm.xml，也都位于grails-app/conf/hibernate目录下。

Grails也支持复用已有Java项目的OR Mapping文件。只需要将java对象放在src/java目录下，将映射文件放在grails-app/conf/hibernate目录下即可。

# 使用Hibernate Annotation

使用Annotation，只需要在src/java目录下创建domain class即可使用[EJB3.0](http://annotations.hibernate.org/)标准annotation。例如：

|  |
| --- |
| package com.books;  import javax.persistence.Entity;  import javax.persistence.GeneratedValue;  import javax.persistence.Id;  @Entity  public class Book {  private Long id;  private String title;  private String description;  private Date date;  @Id  @GeneratedValue  public Long getId() {  return id;  }  public void setId(Long id) {  this.id = id;  }  public String getTitle() {  return title;  }  public void setTitle(String title) {  this.title = title;  }  public String getDescription() {  return description;  }  public void setDescription(String description) {  this.description = description;  }  } |

然后在hibernate.cfg.xml文件中注册domain class就行了：

|  |
| --- |
| <!DOCTYPE hibernate-configuration SYSTEM  "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">  <hibernate-configuration>  <session-factory>  <mapping package="com.books" />  <mapping class="com.books.Book" />  </session-factory>  </hibernate-configuration> |

Grails从src/java目录下载入domain class之后，就会给这些domain class加入动态方法以使他们支持GORM的动态查询、持久化等特性。

# 加入约束

对于java类型的domain class，Grails同样也支持给它们假如GORM约束。例如有domain class：org.example.Book，则约束文件也同样在src/java目录下且和domainclass在一个包下面：src/java/org/example/BookConstraints.groovy。

约束的groovy代码没有任何特别之处：

|  |
| --- |
| constraints = {  title blank: false  author blank: false  } |