<DDD&Redwork框架学习指南>

<for project dddemo>

编写：姜维 日期： 2010年 5 月 10 日

1. 本机运行环境搭建

|  |  |  |
| --- | --- | --- |
| 名称 | 地址 | 版本 |
| TortoiseSVN | http://tortoisesvn.tigris.org/ | 1.6.8 |
| Maven2 | http://maven.apache.org/ | 2.2.1 |
| M2eclipse | http://m2eclipse.sonatype.org/ | 0.10.0 |
| mysql | http://www.mysql.com/ | 5.0 |

1. 软件环境搭建
   1. Maven2安装

1.1.1下载到本地，直接解压即可（尽量避免父级文件夹存在中文，空格等）如：c:\maven

1.1.2.maven配置

* 环境变量：新增M2\_HOME=c:\maven，加入%M2\_HOME%\bin到path变量
* settings.xml文件配置

打开%用户目录%\.m2\settings.xml进行相关配置：

配置本地仓库路径（路径避免空格和中文）

<localRepository>D:/Java/mavenrepository/ </localRepository>

南航内网配置代理

<proxy>

<id>csair</id>

<active>true</active>

<protocol>http</protocol>

<username>footmark</username>

<password>fm.123</password>

<host>10.101.1.6</host>

<port>80</port>

<!—不使用代理的地址列表-->

<nonProxyHosts>10.192.9.15</nonProxyHosts>

</proxy>

</proxies>

* 1. M2eclipse插件安装（推荐eclipse在线安装）
  2. Mysql安装

注：为了不修改项目工程本身配置，请设置mysql帐号：root,密码:1234.

2.项目检出

1.1工程svn地址：http://code.google.com/p/dddemo

1.2 TortoiseSVN 检出步骤

* 新建目录。如D:\projects\dddemo（以后都在该目录下进行讲解）
* 切换到D:\projects\dddemo（目录，单击右键”SVN CheckOut”,出现检出界面输入svn地址，确定。

注：南航内网代理设置

用记事本打开：%APPDATA%\Subversion\servers（如C:\Documents and Settings\Administrator\Application Data\Subversion）

增加以下内容：

http-proxy-exceptions = 10.\*

http-proxy-host =10.101.1.6

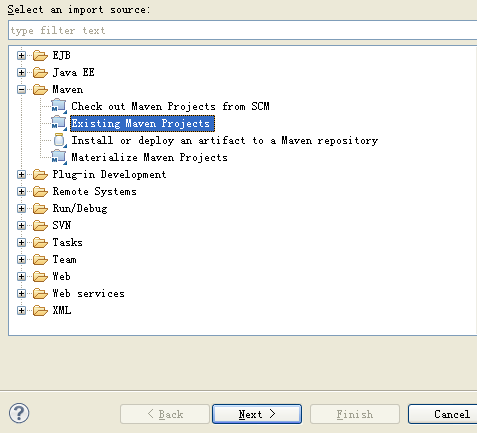
http-proxy-port =80

http-proxy-username =footmark

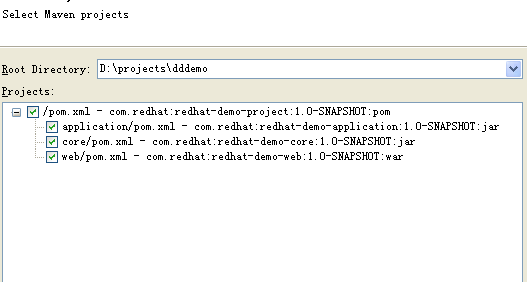
http-proxy-password =fm.123

3.maven构建管理

* 1. 编译：Cmd下进入到D:\projects\dddemo目录。运行命令：mvn clean install
  2. 运用m2eclipse插件导入eclipse（前提：前一步编译通过）
* 步骤1：打开eclipse，file->import->选择”existing Maven Projects”



* 步骤2：选择root directory为：D:\projects\dddemo，确定。出现下图说明可以成功导入eclipse了。

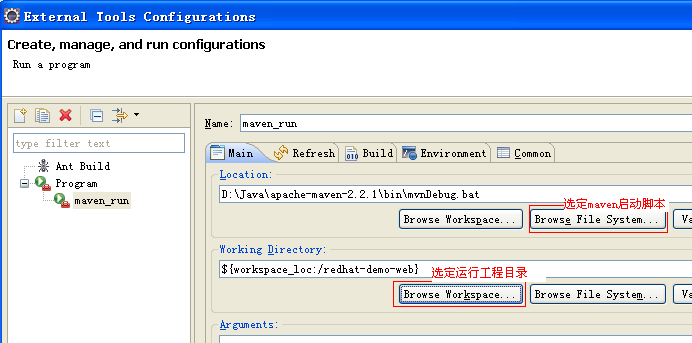


* 1. 运行

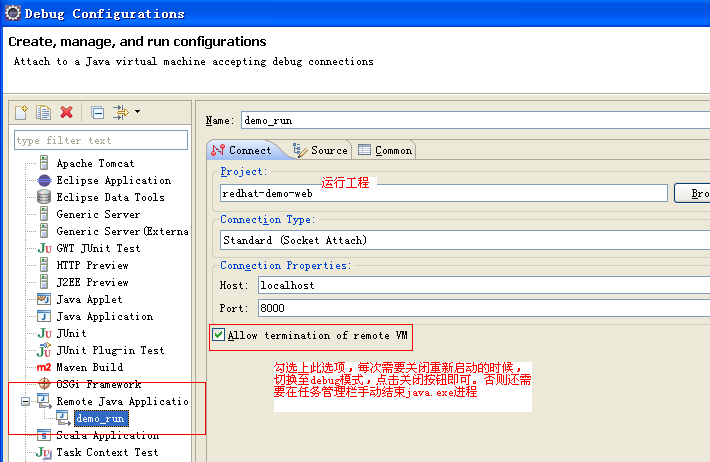
1）.命令行运行：mvn jetty:run

2）.eclipse调试环境运行

* 步骤1：在eclipse中新建一个External Tools Configurtions启动配置文件，然后启动，如下图：



* 步骤2：在eclipse中新建一个Debug Configurtions启动配置文件，然后启动，如下图：



二.运行架构图

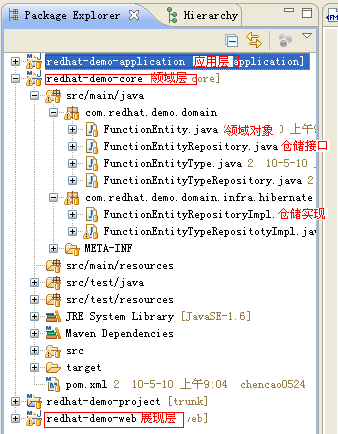
三.开发指南

1.领域驱动设计(DDD)分层界定

1.1领域层(domain)：包含业务对象和仓储接口及其仓储实现

1.2 应用层(application)：用与非业务逻辑的逻辑组装，桥接展现层和领域层

1.3 展现层(web)：用于与用户交互，此处主要用struts2+freemaker+jquery实现



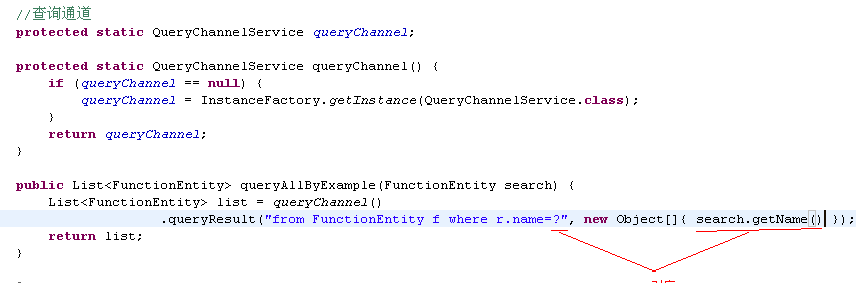
2.领域层(domain)开发指南

* 步骤1：编写领域对象，包括对象属性，领域方法（该对象对应的自身业务方法），ORM配置，仓储对象。需要继承com.dayatang.domain.BaseEntity基类
* 步骤2：编写仓储接口,位于领域对象同一包名下。其中只定义单纯数据库CRUD方法而不包含业务逻辑。需要继承com.dayatang.domain.BaseEntityRepository基类，这样就默认实现了常用的数据库CRUD方法。
* 步骤3：编写仓储实现类：在此用hibernate做数据库持久化工具，因此我们同样需要继承com.dayatang.spring.repository.BaseEntityRepositoryHibernateSpring基类
* 步骤4：配置hibernate mapping

3.应用层(application)开发指南

* 注意点1：应用层可以按每一个业务对象对应一个应用层服务，如dddemo。也可以多个业务对象共用一个应用层服务。
* 注意点2：应用层只能包含非业务逻辑的代码，用于数据包装等
* 注意点3：应用层不应该引用具体具体仓储对象进行数据库操作，如果需要进行数据访问需要用到查询通道（本指南介绍DDD框架引入一个数据访问的概念）。

查询通道应用代码片段：



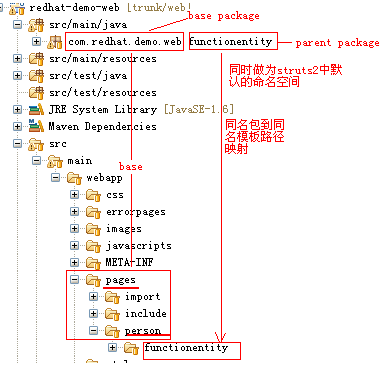
4.展现层(web)开发指南

4.1.redwork框架硬性结构规约

Redwork框架一个重要特性就是实现了零配置。其主要原理就是通过包名以及类名进行页面渲染，完成跳转。所以为了让零配置特性能够发挥作用，我们在开发过程就必须遵循一些默认的规约进行包结构、模板页面放置路径结构、类名和页面模板文件名称定义。

4.1.1 首先我在struts.xml中定义了基础package和模板页面的基础路径。如下图：

也就是说在基础package和模板页面的基础路径以下的内容都是相互对应的。如下图：



4.2.action开发

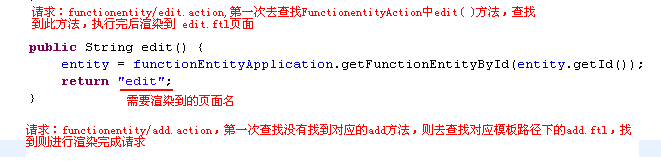
* 步骤1： 新建action继承 RedworkActionSupport 并实现RedworkAction接口
* 规约1：如果请求url是：parent package/xxxx.action,就会默认对应到parent packageAction 中xxxx的方法。

如：请求url：functionentity/add.action，就会对应到FunctionentityAction这个类中的add方法。

* 规约2：如果请求url是：parent package/ xxxx1-xxxxx2.action,就会默认对应到xxxx1Action 中xxxxx2的方法。

如：请求url：functionentity/ functionEntityType-addFunctionEntityType.action，就会对应到functionEntityTypeAction这个类中的addFunctionEntityType方法。

* 规约3：请求action：xxx.action首先会去查找按规约1，2对应action类中的同名的xxx方法。如果存在则执行该方法，完成后跳转至return map所要渲染的页面。否则直接渲染到对应页面模板文件夹中的xxx.ftl页面。



4.3页面开发：熟悉freemaker语法

4.4 组件引用

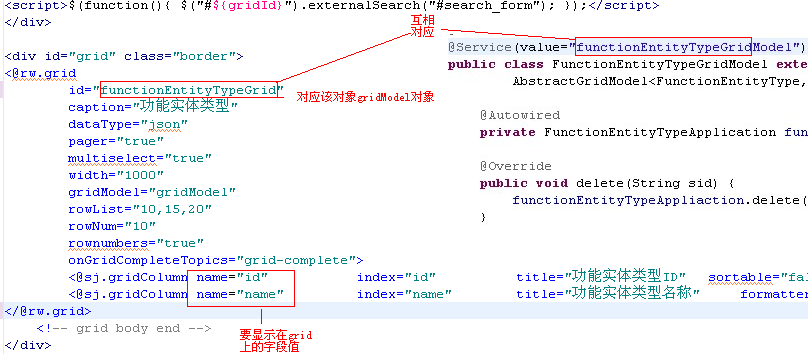
4.4.1 Grid组件

* 步骤1：在action所在同一包结构下编写gridModel对象，继承AbstractGridModel基类，重写需要重新的方法。代码片段：



注意点：编写完成后一定要注册成spring管理下的服务，通过spring XML配置或者标注。

* 步骤2：页面引用



4.4.2 tree组件

* 步骤1：
* 步骤2：
* 步骤3：