



# Tapestry5.3 手册

编写：飞风

Tapestry5.3 手册.....	1
tapestry 开发环境所需软件安装 .....	2
tapestry 项目创建与运行 .....	6
eclipse 导入 tapestry 创建的项目与运行 .....	7
第一讲：hello tapestry .....	11
第二讲：tapestry 增删改查---添加数据 .....	14
第三讲：tapestry 增删改查---查询数据 .....	25
第四讲：tapestry 增删改查---修改数据 .....	29
第五讲：tapestry 增删改查---删除数据 .....	31
第六讲：tapestry 输出方式 .....	33
第七讲：tapestry 可预览的模板页 .....	35
第八讲：tapestry 组件 .....	35
第九讲：tapestry pagelink 组件 .....	36
第十讲：tapestry 通过地址栏传值@PageActivationContext .....	37
第十一讲：tapestry session 存储.....	38
第十二讲：tapestry actionlink 与 eventlink 组件 .....	41
第十三讲：tapestry form 组件 .....	42
第十四讲：tapestry form 验证 .....	45
第十五讲：tapestry form 自定义验证 .....	48
第十六讲：tapestry 数据循环显示 loop 与 grid 组件.....	57
第十七讲：tapestry 不同类型的变量在页面之间值的传递 .....	59
第十八讲：tapestry 表单核心组件 .....	63
第十九讲：tapestry 多个提交按钮(submit).....	68
第二十讲：tapestry 多表单(forms) .....	71
第二十一讲：tapestry 的 LinkSubmit 等同于 submit 来使用 .....	74
第二十二讲：tapestry 的 Select 组件 .....	76
第二十三讲：tapestry 的 Delegate 和 Blocks 组件 .....	78
第二十三讲：tapestry 的实体(entity)范本 .....	79
第二十四讲：tapestry 的 grid 组件.....	81
第二十五讲：tapestry 的 loop 组件 .....	86
第二十六讲：tapestry 的树状(tree)组件 .....	104
第二十七讲：tapestry 与 javaScript.....	109
第二十八讲：tapestry 与可重复使用的 javaScript .....	110
第二十九讲：tapestry 更好的可重复使用 javaScript.....	113



第三十讲: tapestry 强大的 javaScript 混合(mixins) .....	116
第三十一讲: tapestry 的 javaScript 混合(mixins)统计点击数.....	118
第三十二讲: tapestry javaScript 动画 .....	124
第三十三讲: tapestry Ajax eventlink 无刷新页面.....	125
第三十四讲: tapestry Ajax actionlink 无刷新页面 .....	127
第三十五讲: tapestry Ajax zone 组件无黄色闪烁的背景 .....	128
第三十六讲: tapestry Ajax 表单(form).....	130
第三十七讲: tapestry Ajax 表单(form)多 zone 组件 .....	133
第三十八讲: tapestry Ajax 关联下拉选框(select)组件.....	136
第三十九讲: tapestry 页面加载等待过程(ProgressiveDisplay)组件.....	140
第四十讲: tapestry ProgressiveDisplay 组件详细使用 .....	142
第四十一讲: tapestry Ajax 周期性更新 .....	144
第四十二讲: tapestry Ajax 混合(mixins)周期性更新 .....	147
第四十三讲: tapestry Ajax 表单验证 .....	150
第四十四讲: tapestry upload 组件 .....	157
第四十五讲: tapestry upload 组件使用 input type="file"实现上传.....	160
第四十六讲: tapestry upload 组件常用的配置详解 .....	162
第四十七讲: tapestry 与 swfupload 上传.....	166
第四十八讲: tapestry 与 淘宝 kissy editor 编辑器带图片上传.....	172
第四十九讲: tapestry+spring 整合 .....	183
第五十讲: tapestry 自定义 404 错误页面 .....	196
第五十一讲: tapestry 自定义异常页面 .....	198

## tapestry 开发环境所需软件安装

tapestry 开发环境所需软件安装 (安装中出现问题可加入技术交流 QQ 群:41138107或33106572询问):

1,JDK-----[下载网址](#)

JDK 安装完成之后, 需配置下环境变量。



新建全局变量 JAVA\_HOME: jdk 安装目录



设置全局变量 Path: %JAVA\_HOME%\bin



2,eclipse3.7.2---[Eclipse Classic 3.7.2](http://www.eclipse.org/downloads/),---官方下载地址: <http://www.eclipse.org/downloads/>  
只要 jdk 正确安装, eclipse 直接解压就可以使用了。

3,maven 客户端---官方下载地址: <http://maven.apache.org/>

注意: 下载的时候选择.zip 的文件, 另外的.tar.gz 是源文件无需下载。解压放到电脑上, 我的电脑路径: D:\java\apache-maven-3.0.4

配置 maven 环境变量:

新建 M2\_HOME: maven 安装路径;



增加 path:%M2\_HOME%\bin;



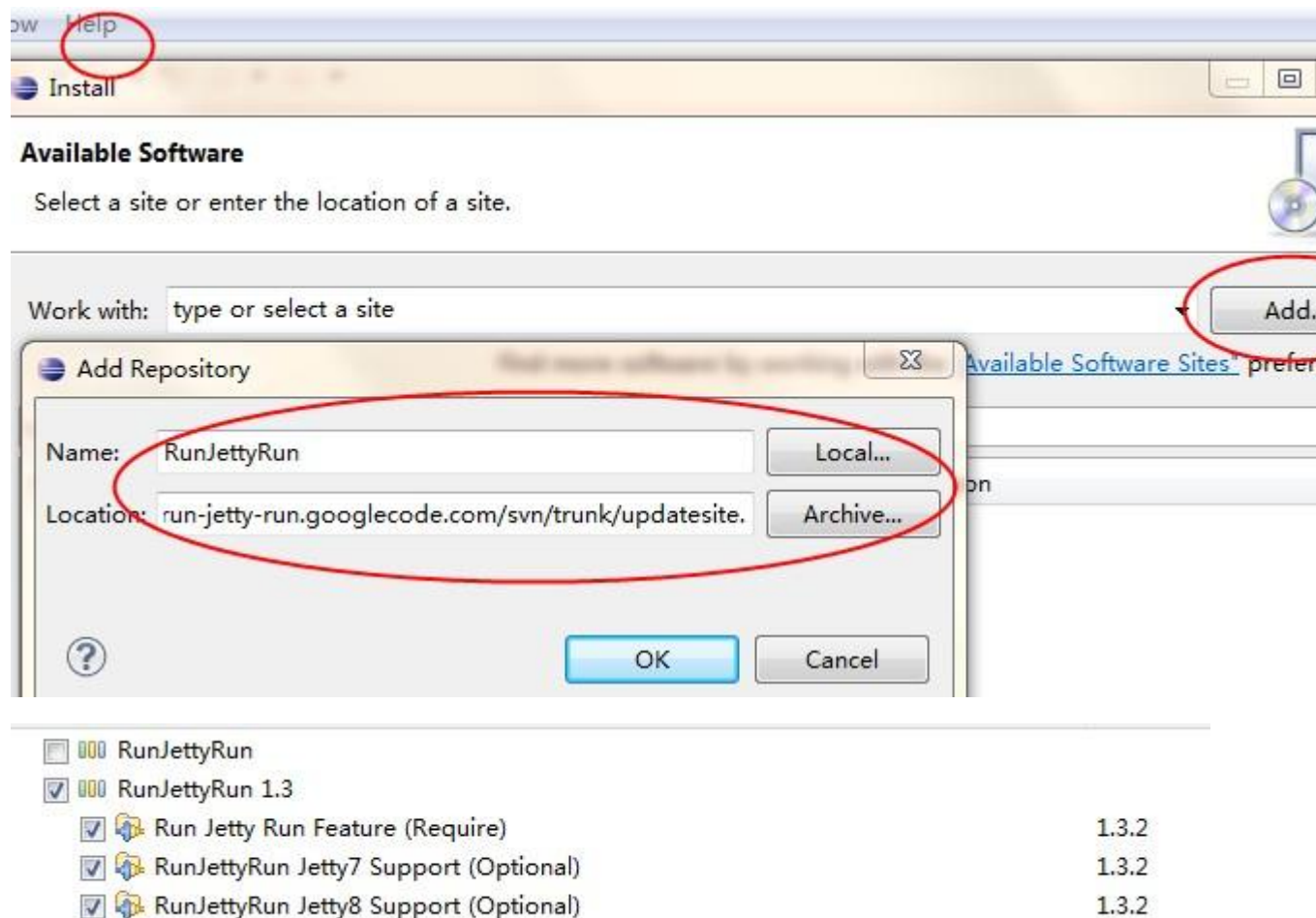
测试: 点击电脑-开始-附件-命令提示符(如果是 W7 请点右键以管理员身份运行), 输入 mvn -v  
结果如下, 说明安装成功。



```
C:\Windows\system32>mvn -v
Apache Maven 3.0.4 (r1232337; 2012-01-17 16:44:56+0800)
Maven home: D:\java\apache-maven-3.0.4
Java version: 1.7.0_05, vendor: Oracle Corporation
Java home: C:\Program Files\Java\jdk1.7.0_05\jre
Default locale: zh_CN, platform encoding: GBK
OS name: "windows 7", version: "6.1", arch: "x86", family: "windows"
C:\Windows\system32>
```

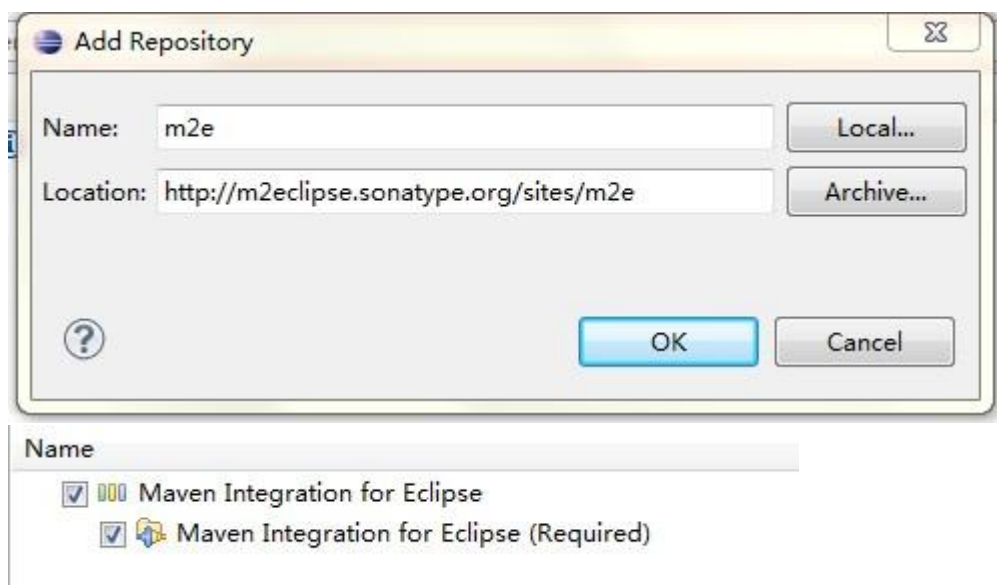
4,runJettyRun、maven---eclipse 的插件：安装方法可查看本站 [eclipse 常用插件地址](#)与 [eclipse 开发环境\(支持 tomcat、maven、jetty、JBoss\)](#)

RunJettyRun Eclipse 插件，选择 help--Install new software---add 然后输入 RunJettyRun，地址：<http://run-jetty-run.googlecode.com/svn/trunk/updatesite>



安装 maven eclipse 插件，地址：<http://m2eclipse.sonatype.org/sites/m2e>





## tapestry 项目创建与运行

1、在 eclipse 工程目录下创建项目：

第一次运行 eclipse 的时候会弹出选择工程项目存放地点，如我的存放路径 D:\workspace。我们就把 tapestry 创建的项目放到这个目录方便统一管理。

打开系统的命令提示符界面，以 windows 为例，路径：开始--附件--命令提示符。注意如果您的系统是 vista 或 W7，必须右键选择以管理员身份运行。

打开界面，输入 d:按回车键进入 d 盘。输入 cd D:\workspace 按回车进入工作空间目录。



输入：mvn archetype:generate -DarchetypeCatalog=http://tapestry.apache.org 回车，选择1回车，选择6回车，输入 groupId 值 com.tapestry 回车，输入 artifactId 值 tapestryStart 回车，输入版本号为1.0回车，输入包的名字为 com.tapestry.app 回车,输入 Y 回车创建成功！



```
[INFO] No archetype defined. Using maven-archetype-quickstart (org.apache.maven.plugins:maven-archetype-quickstart:1.0)
Choose archetype:
1: http://tapestry.apache.org -> quickstart (Tapestry 5 Quickstart Project)
2: http://tapestry.apache.org -> tapestry-archetype (Tapestry 4.1.6 Archetype)
Choose a number: : 1
Choose version:
1: 5.0.19
2: 5.1.0.5
3: 5.2.6
4: 5.3
5: 5.3.1
6: 5.3.2
Choose a number: 6: 6
Define value for property 'groupId': : com.tapestry
Define value for property 'artifactId': : tapestryStart
Define value for property 'version': : 1.0-SNAPSHOT: : 1.0
Define value for property 'package': : com.tapestry: : com.tapestry.app
Confirm properties configuration:
groupId: com.tapestry
artifactId: tapestryStart
version: 1.0
package: com.tapestry.app
Y: :
```

打开 D:\workspace 会看到已经创建了一个文件夹名为 `tapestryStart` 的 `tapestry` 工程，在提示命令符中输入 `cd tapestryStart` 回车进入该项目，再输入 `mvn eclipse:eclipse -DdownloadSources=true` 回车下载 eclipse 所需的 jar 文件，到此所有创建完成。

```
[INFO] BUILD SUCCESS
[INFO] Total time: 10.984s
[INFO] Finished at: Fri Feb 10 09:56:49 CST 2012
[INFO] Final Memory: 6M/15M
D:\workspace\tapestryStart>
```

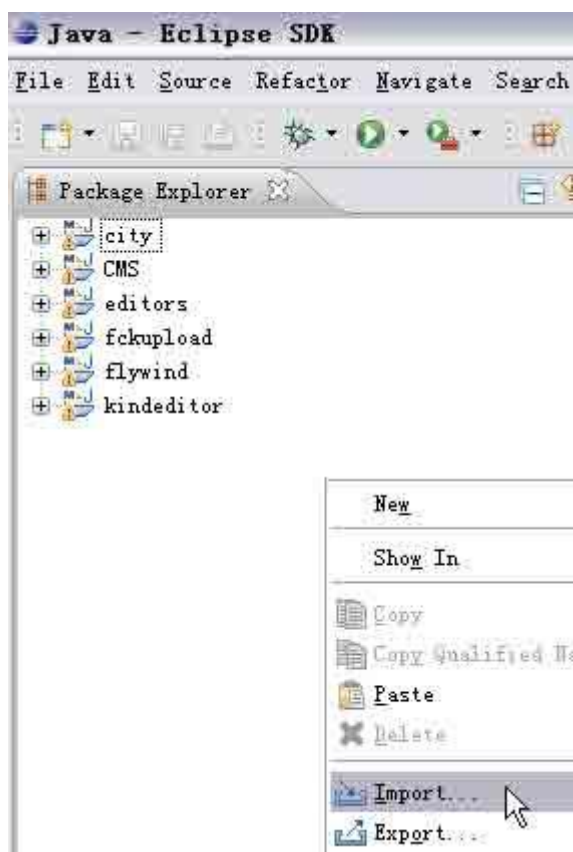
输入 `mvn jetty:run` 回车运行该项目后，打开浏览器输入 <http://localhost:8080/tapestryStart/>



可以看到已经跑起来了。

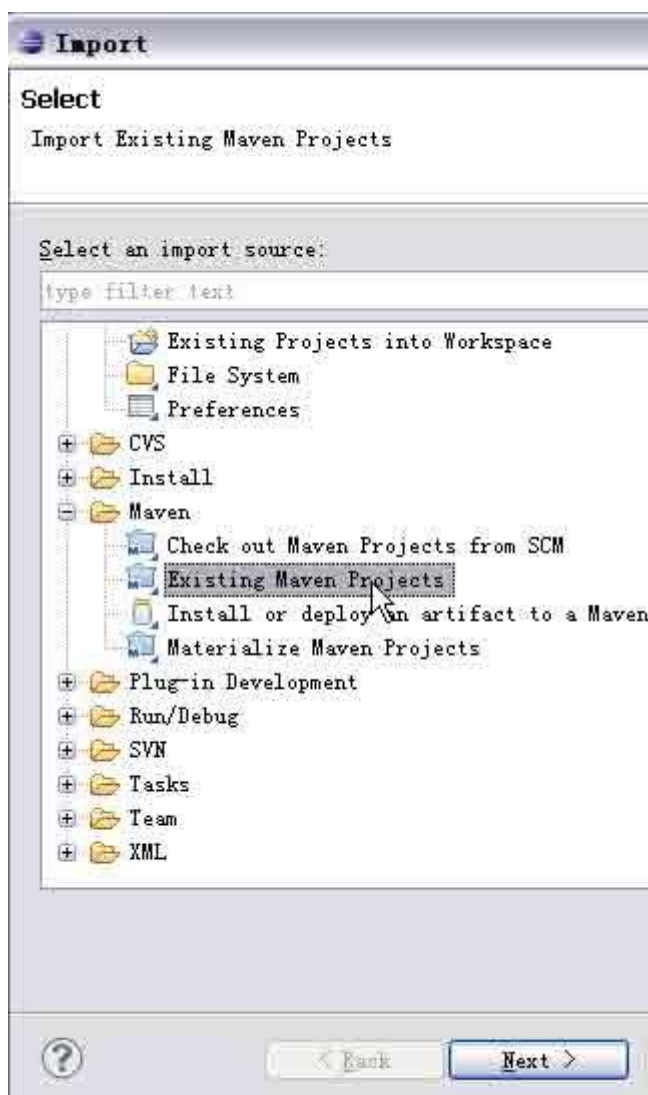
## eclipse 导入 tapestry 创建的项目与运行

运行 eclipse3.7 在左边的项目管理中右键选择 import.



点击 maven 展开菜单选择 Existing Maven Projects 现有的项目，



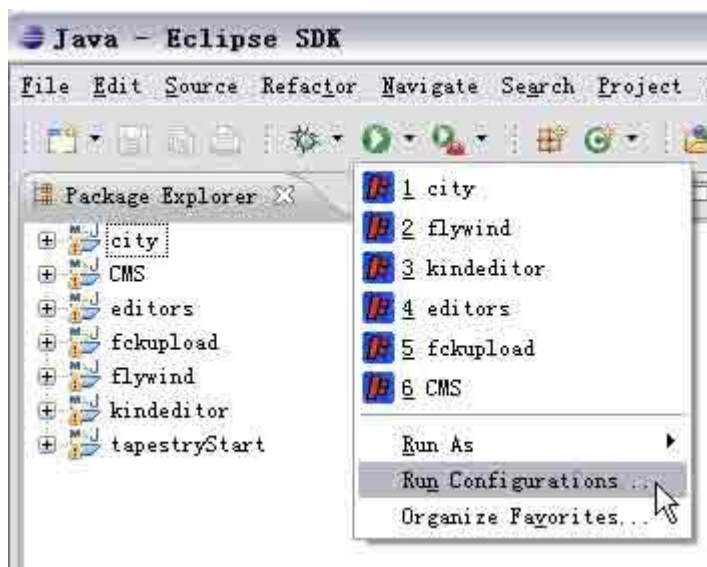


点击 next 在 Root Directory 选择我们创建的 D:\workspace\tapestryStart 的项目,点击 finish 完成导入。

至此我们已经成功导入了创建的项目，下面运行起来。



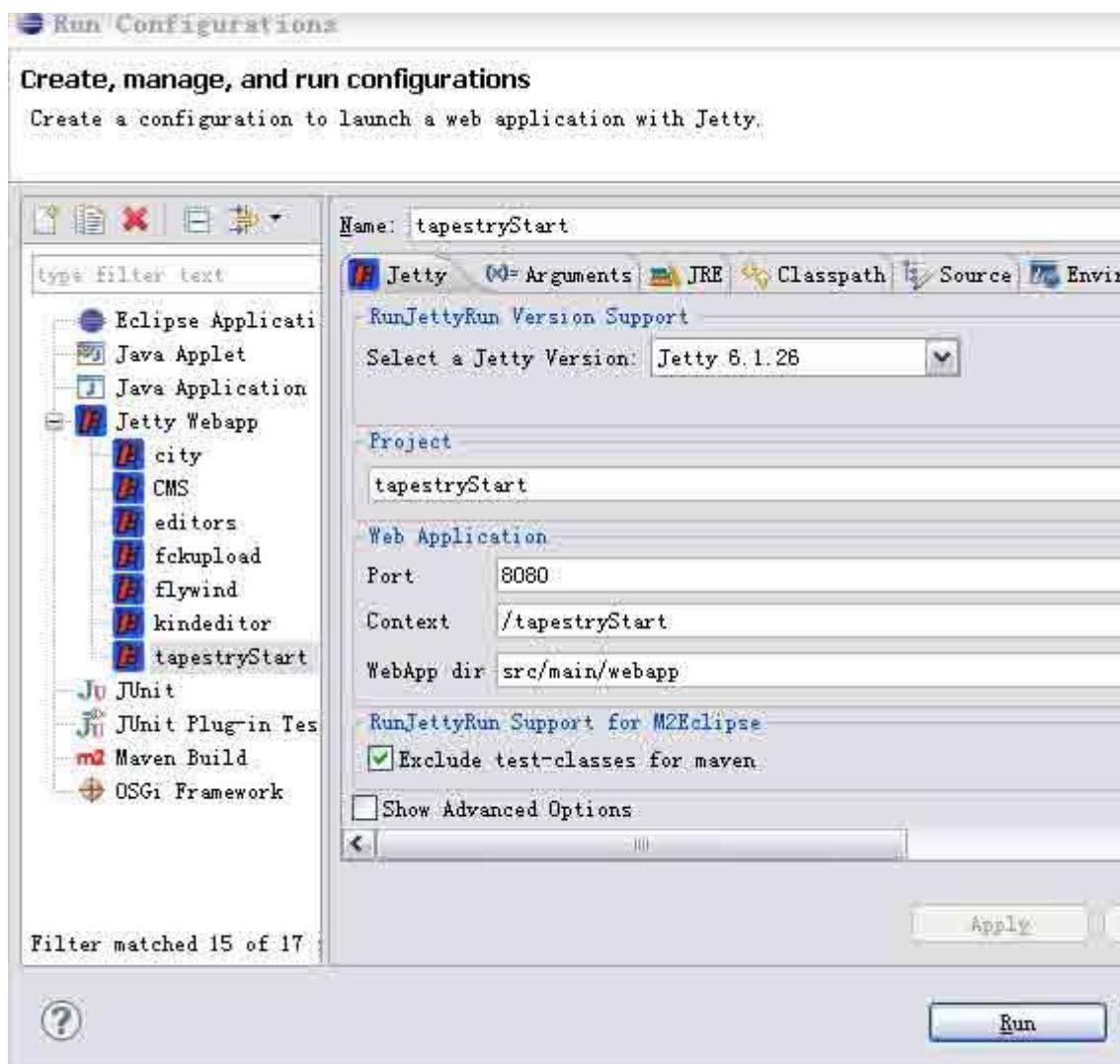
点击左边的 tapestryStart 文件夹，再 eclipse 菜单的 run 按钮，就是那个类似播放的按钮，选择 Run configurations,



右键点击 Jetty Webapp 选择 New 创建 jetty 服务，最后点击 run 开启 jetty 服务，在浏览器窗口中输入 <http://localhost:8080/tapestryStart/> 可以看到项目跑起来了，注意刚才那个命令窗口必须关掉，因为不能有2个8080的端口，避免冲突，当然你也可以



设置下端口就不出冲突了。



## 第一讲：hello tapestry

tapestry 具体的介绍，大家可以去官网 <http://tapestry.apache.org/> 了解，apache 的强大大家也有目共睹。

tapestry 使用了组件开发模式，很大的提高了工作效率，速度也很快，特别的5.3之后版本，官方文档也很齐全可惜是全英文，英文不好的很头疼，版本升级超级容易等等好处实在是很多啊！有兴趣的可去官网了解，编写这个教程是为了给初学着，或者没有学过程序开发又想从事程序开发的童鞋有很大的参考价值。

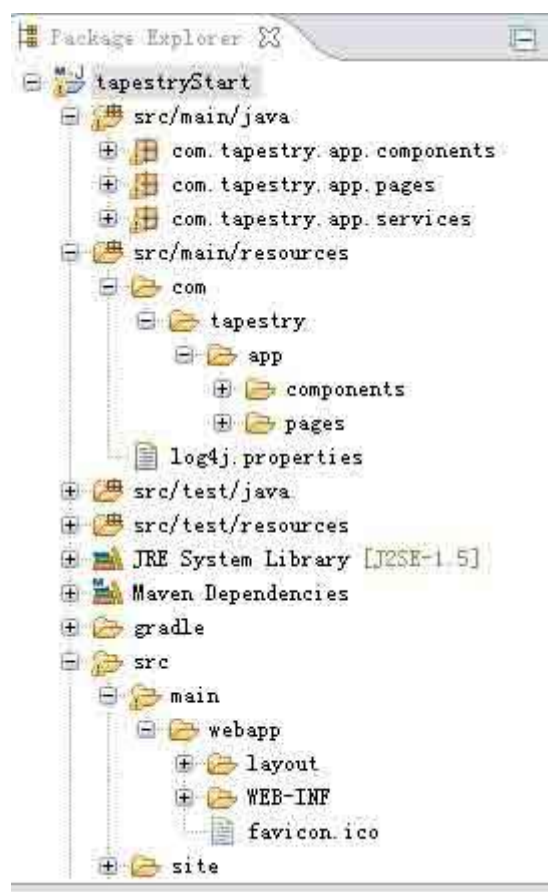
个人简单介绍下，我并非专业计算机出身而是广美毕业，毕业后一直从事 IT 行业，之前在小公司写过几年的 PHP，有同类经验的童鞋都知道，在小公司你会多东西越多，老板越



赏识加上个人对程序开发有很大兴趣，对 **java** 也是第一次写，所以有写错的地方请大家指正、也很乐意与各位前辈交流学习、希望 **tapestry** 在国内能早日像他的兄弟框架 **struts** 那么流行。本编系列教程初定为 **tapestryStart**，为了能正确运行案例，需要的可以[下载 tapestryStart 源码](#)仅供参考。安装方法请看[eclipse 导入 tapestry 创建的项目与运行](#)。

言归正传，**tapestry** 分开发模式与产品模式(生产模式)两类，我们开发的时候使用开发模式，打包部署的时候使用产品模式。却换非常简单，只要简单设置就行了。本系列教程开发模式：**tapestryX+tapestry-hibernate+mySql**。在开发模式下一个页面由一个 **.java** 跟一个配对的 **.tml** 文件组成。

来看下 **tapestry** 项目，官方推荐使用 **maven** 管理项目，我们之前创建的项目也是 **maven** 项目，下面我们来看下结构。如图



**components** 存放组件的文件夹，**page** 存放页面的文件夹，**services** 存放服务的文件夹，默认就这3个，我们在开发中可自己增加相应文件夹，常用的有 **entities** 存放实体，**until** 存放常用的 **java** 转换静态函数，**data** 存放枚举，**mixins** 混合，**base** 可被继承的基础类等等。

hello **tapestryStart**。

在 **com.tapestry.app.pages** 下创建一个 **TapestryStart.java** 文件，在 **webapp** 下创建 **TapestryStart.tml** 文件。下面看下他们的源代码。



TapestryStart.java 在这个文件中创建了一个私有变量 `hello`, `tapestry` 读写是使用 `get`+变量名字(首字母大写)与 `set`+变量名字(首字母大写)函数。也就是说 `hello` 读使用 `getHello` 函数、写使用 `setHello` 函数, 以此类推。这里其实不用写 `setHello`, 写上是为了方便大家理解。读写一起的时候可以使用简写 `@Property private String hello`;这样的话读写都已经有了, 以后会讲到。

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

public class TapestryStart {

    private String hello;

    public String getHello() {
        return "hello Tapestry Start";
    }

    public void setHello(String hello) {
        this.hello = hello;
    }
}
```

TapestryStart.tm

```
<html t:type="layout" title="About tapestryStart"
      xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
      xmlns:p="tapestry:parameter">

    <p style="color:#ff6600">${hello}</p>

</html>
```



## 第二讲：tapestry 增删改查---添加数据

增、删、改、查。这是开发程序最基本的、必须掌握的事。如果看不懂可以先跳过此文，先看看后面的再回来仔细阅读，应该能更好理解。

实现使用了 **tapestry ioc** 的接口开发方式，也就是一个接口 **DAO**，一个接口方法 **Impl**，使用的时候调用接口的方法。要做增删改查还需要实体 **entity** 以及数据库，这里使用 **mySql**。实体与数据库的连接使用 **tapestry-hibernate**、**c3p0**、**mysql-connector-java**。

打开项目根目录下的 **pom.xml**，增加需要的 **tapestry-hibernate**、**c3p0**、**mysql-connector-java**、**tapestry-upload** 连接，代码如下：

```
<dependency>
<groupId>org.apache.tapestry</groupId>
<artifactId>tapestry-hibernate</artifactId>
<version>${tapestry-release-version}</version>
</dependency>
<dependency>
    <groupId>org.apache.tapestry</groupId>
    <artifactId>tapestry-upload</artifactId>
    <version>${tapestry-release-version}</version>
</dependency>
<dependency>
<groupId>c3p0</groupId>
<artifactId>c3p0</artifactId>
<version>0.9.1.2</version>
</dependency>
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>5.1.17</version>
</dependency>
```

修改服务，打开 **com.tapestry.app.services** 下的 **AppModule.java** 文件进行配置，源码如下：

```
package com.tapestry.app.services;

import java.io.IOException;

import org.apache.tapestry5.*;
import org.apache.tapestry5.hibernate.HibernateCoreModule;
```





```
import org.apache.tapestry5.hibernate.HibernateModule;
import org.apache.tapestry5.hibernate.HibernateSymbols;
import org.apache.tapestry5.hibernate.HibernateTransactionAdvisor;
import org.apache.tapestry5.ioc.MappedConfiguration;
import org.apache.tapestry5.ioc.MethodAdviceReceiver;
import org.apache.tapestry5.ioc.OrderedConfiguration;
import org.apache.tapestry5.ioc.ServiceBinder;
import org.apache.tapestry5.ioc.annotations.Local;
import org.apache.tapestry5.ioc.annotations.Match;
import org.apache.tapestry5.ioc.annotations.SubModule;
import org.apache.tapestry5.services.Request;
import org.apache.tapestry5.services.RequestFilter;
import org.apache.tapestry5.services.RequestHandler;
import org.apache.tapestry5.services.Response;
import org.apache.tapestry5.upload.services.UploadModule;
import org.apache.tapestry5.upload.services.UploadSymbols;
import org.slf4j.Logger;

@SubModule({HibernateModule.class,
HibernateCoreModule.class,UploadModule.class})
public class AppModule
{
    public static void bind(ServiceBinder binder)
    {
        // binder.bind(MyServiceInterface.class, MyServiceImpl.class);
    }

    public static void contributeFactoryDefaults(
        MappedConfiguration<String, Object> configuration)
    {
        configuration.override(SymbolConstants.APPLICATION_VERSION,
"1.0");
    }

    public static void contributeApplicationDefaults(
        MappedConfiguration<String, Object> configuration)
    {
        configuration.add(SymbolConstants.SUPPORTED_LOCALES, "en");
        configuration.add(SymbolConstants.PRODUCTION_MODE, "false");
        configuration.add(UploadSymbols.FILESIZE_MAX, "50000000");
        configuration.add(SymbolConstants.APPLICATION_VERSION,
"1.0-SNAPSHOT");
        configuration
```



```
.add(HibernateSymbols.ENTITY_SESSION_STATE_PERSISTENCE_STRATEGY_ENABLED
,
"true");
    }

    @Match("*.DAO")
    public static void adviseTransactions(HibernateTransactionAdvisor advisor,
    MethodAdviceReceiver receiver) {
    advisor.addTransactionCommitAdvice(receiver);
    }

    public RequestFilter buildTimingFilter(final Logger log)
    {
        return new RequestFilter()
        {
            public boolean service(Request request, Response response,
    RequestHandler handler)
                throws IOException
            {
                long startTime = System.currentTimeMillis();

                try
                {
                    return handler.service(request, response);
                } finally
                {
                    long elapsed = System.currentTimeMillis() - startTime;

                    log.info(String.format("Request    time:    %d    ms",
    elapsed));
                }
            }
        };
    }

    public void
    contributeRequestHandler(OrderedConfiguration<RequestFilter>
    configuration,

                                @Local
                                RequestFilter filter)
    {
        configuration.add("Timing", filter);
    }
}
```



```
}  
}
```

在 `src/main/resources` 下增加 `hibernate.cfg.xml` 文件, 代码如下:

```
<?xml version='1.0' encoding='utf-8'?>  
<!--  
    Copyright 2007 The Apache Software Foundation  
  
    Licensed under the Apache License, Version 2.0 (the "License");  
    you may not use this file except in compliance with the License.  
    You may obtain a copy of the License at  
  
        http://www.apache.org/licenses/LICENSE-2.0  
  
    Unless required by applicable law or agreed to in writing, software  
    distributed under the License is distributed on an "AS IS" BASIS,  
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
    See the License for the specific language governing permissions and  
    limitations under the License.  
-->  
  
<!DOCTYPE hibernate-configuration PUBLIC  
    "-//Hibernate/Hibernate Configuration DTD//EN"  
    "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">  
  
<hibernate-configuration>  
  
    <session-factory>  
        <property  
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</propert  
y>  
        <property  
name="hibernate.connection.url">jdbc:mysql:///example?useUnicode=true&a  
mp;characterEncoding=UTF-8</property>  
        <property name="hibernate.connection.username">root</property>  
        <property name="hibernate.connection.password">root</property>  
        <property  
name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>  
  
        <property name="show_sql">true</property>  
        <property name="format_sql">true</property>
```



```
<property name="hbm2ddl.auto">update</property>

<property name="hibernate.c3p0.min_size">5</property>
<property name="hibernate.c3p0.max_size">20</property>
<property name="hibernate.c3p0.timeout">300</property>
<property name="hibernate.c3p0.max_statements">50</property>
<property name="hibernate.c3p0.idle_test_period">3000</property>

<property name="hibernate.generate_statistics">true</property>
</session-factory>

</hibernate-configuration>
```

在数据库里面创建名为 **example** 的数据库。

创建 **com.tapestry.app.entities** 包，在包内创建 **User.java** 的实体。代码如下：

```
package com.tapestry.app.entities;

import java.util.Date;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class User {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="user_id")
    private Long id;

    private String name;

    private int age;

    private Date time;

    public Long getId() {
```



```
return id;
}

public void setId(Long id) {
    this.id = id;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public int getAge() {
    return age;
}

public void setAge(int age) {
    this.age = age;
}

public Date getTime() {
    return time;
}

public void setTime(Date time) {
    this.time = time;
}
}
```

创建 `com.tapestry.app.pages.crud` 包，包内创建 `UserCreate.java`。在 `webapp` 下创建 `crud` 文件夹，文件夹内创建 `UserCreate.tml`。他们的代码如下：

`UserCreate.java`

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
```



```
* 时间: 2012-02-29
*/
package com.tapestry.app.pages.crud;

import java.util.Date;

import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.ioc.annotations.Inject;

import com.tapestry.app.entities.User;
import com.tapestry.app.services.StartDAO;

public class UserCreate {

    @Property
    private User user;

    //导入服务接口
    @Inject
    private StartDAO dao;

    //初始化 user 实体
    void onPrepare(){
        user = new User();
    }

    //提交表单的时候执行存储,返回当前页面
    Object onSuccess(){
        //如果时间为空值输入系统当前时间
        if(user.getTime() == null){
            user.setTime(new Date());
        }
        dao.create(user);
        return this;
    }
}
```

#### UserCreate.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
  xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
  xmlns:p="tapestry:parameter">
```





```
<t:form>
  <t:errors/>
  <p> 用 户 名 : <t:textfield t:id="name" value="user.name"
t:validate="required"/></p>
  <p> 年 龄 : <t:textfield t:id="age" value="user.age"
t:validate="required"/></p>
  <p><input type="submit" value="创建"/></p>
</t:form>
</html>
```

创建服务，在 `com.tapestry.app.services` 包里创建操作数据的接口与方法：`StartDAO.java` 与 `StartDAOImpl.java` 这是我写的增删改查代码，已经封装过很方便以后使用。源码如下：

`StartDAO.java`

```
package com.tapestry.app.services;

import java.io.Serializable;
import java.util.List;
import java.util.Map;

import org.apache.tapestry5.hibernate.annotations.CommitAfter;

public interface StartDAO {

    <T, PK extends Serializable> T findById(Class<T> type, PK id);

    @CommitAfter
    <T> T create(T t);

    @CommitAfter
    <T> T update(T t);

    @CommitAfter
    <T, PK extends Serializable> void deleteById(Class<T> type, PK id);

    <T> List<T> findWithNameQuery(String queryName, int num);
    <T> List<T> findWithQuery(String queryName);

}
```



#### StartDAOImpl.java

```
package com.tapestry.app.services;

import java.io.Serializable;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;

import org.hibernate.Query;
import org.hibernate.Session;

public class StartDAOImpl implements StartDAO {

    private Session session;

    public StartDAOImpl(Session session){
        this.session = session;
    }

    @SuppressWarnings("unchecked")
    public <T, PK extends Serializable> T findById(Class<T> type, PK id){
        return (T) session.get(type, id);
    }

    public <T> T create(T t){
        session.persist(t);
        session.flush();
        session.refresh(t);
        //session.save(t);
        return t;
    }

    public <T> T update(T t){
```



```
session.merge(t);
return t;
}

public <T, PK extends Serializable> void deleteByID(Class<T> type, PK id){
@SuppressWarnings("unchecked")
T t = (T) session.get(type, id);
session.delete(t);
}

@SuppressWarnings("unchecked")
public <T> List<T> findWithNameQuery(String queryName, int num){
return session.createQuery(queryName).setMaxResults(num).list();
}

@SuppressWarnings("unchecked")
public <T> List<T> findWithQuery(String queryName){
return session.createQuery(queryName).list();
}
}
```

最后修改下 **AppModule.java** 把服务绑定起来，代码如下：

```
package com.tapestry.app.services;

import java.io.IOException;

import org.apache.tapestry5.*;
import org.apache.tapestry5.hibernate.HibernateCoreModule;
import org.apache.tapestry5.hibernate.HibernateModule;
import org.apache.tapestry5.hibernate.HibernateSymbols;
import org.apache.tapestry5.hibernate.HibernateTransactionAdvisor;
import org.apache.tapestry5.ioc.MappedConfiguration;
import org.apache.tapestry5.ioc.MethodAdviceReceiver;
import org.apache.tapestry5.ioc.OrderedConfiguration;
import org.apache.tapestry5.ioc.ServiceBinder;
import org.apache.tapestry5.ioc.annotations.Local;
import org.apache.tapestry5.ioc.annotations.Match;
import org.apache.tapestry5.ioc.annotations.SubModule;
import org.apache.tapestry5.services.Request;
import org.apache.tapestry5.services.RequestFilter;
import org.apache.tapestry5.services.RequestHandler;
import org.apache.tapestry5.services.Response;
```



```
import org.apache.tapestry5.upload.services.UploadModule;
import org.apache.tapestry5.upload.services.UploadSymbols;
import org.slf4j.Logger;

@SubModule({HibernateModule.class,
HibernateCoreModule.class,UploadModule.class})
public class AppModule
{
    public static void bind(ServiceBinder binder)
    {
        binder.bind(StartDAO.class, StartDAOImpl.class);
    }

    public static void contributeFactoryDefaults(
        MappedConfiguration<String, Object> configuration)
    {
        configuration.override(SymbolConstants.APPLICATION_VERSION,
"1.0");
    }

    public static void contributeApplicationDefaults(
        MappedConfiguration<String, Object> configuration)
    {
        configuration.add(SymbolConstants.SUPPORTED_LOCALES, "en");
        configuration.add(SymbolConstants.PRODUCTION_MODE, "false");
        configuration.add(UploadSymbols.FILESIZE_MAX, "50000000");
        configuration.add(SymbolConstants.APPLICATION_VERSION,
"1.0-SNAPSHOT");
        configuration
.add(HibernateSymbols.ENTITY_SESSION_STATE_PERSISTENCE_STRATEGY_ENABLED
,
"true");
    }

    @Match("*/DAO")
    public static void adviseTransactions(HibernateTransactionAdvisor advisor,
MethodAdviceReceiver receiver) {
        advisor.addTransactionCommitAdvice(receiver);
    }

    public RequestFilter buildTimingFilter(final Logger log)
    {
        return new RequestFilter()
```



```
{
    public boolean service(Request request, Response response,
RequestHandler handler)
        throws IOException
    {
        long startTime = System.currentTimeMillis();

        try
        {
            return handler.service(request, response);
        } finally
        {
            long elapsed = System.currentTimeMillis() - startTime;

            log.info(String.format("Request    time:    %d    ms",
elapsed));
        }
    }
};
}
```

```
public                                     void
contributeRequestHandler(OrderedConfiguration<RequestFilter>
configuration,

                                @Local
                                RequestFilter filter)
{
    configuration.add("Timing", filter);
}
}
```

运行 <http://localhost:8080/tapestryStart/crud/usercreate> 可以添加数据了



## 第三讲：tapestry 增删改查---查询数据

思路：使用 loop 组件循环显示每条增加的数据，添加用户使用了 pagelink。

在 com.tapestry.app.pages.crud 包里创建 UserList.java, 在 webapp/crud 里创建 UserList.tml



代码如下:

UserList.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages.crud;

import java.util.List;

import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.ioc.annotations.Inject;

import com.tapestry.app.entities.User;
import com.tapestry.app.services.StartDAO;

public class UserList {

    //打开 user 读写
    @Property
    private User user;

    //打开 user 阵列的读写
    @Property
    private List<User> users;

    //导入操作数据库的服务
    @Inject
    private StartDAO dao;

    //页面加载时设置渲染
    void setupRender(){
        //查询 User 数据表
        StringBuffer sql = new StringBuffer();
        sql.append("from User");
        users = dao.findWithQuery(sql.toString());
    }
}
```





```
}
```

#### UserList.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
  xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
  <style>
    .table{border-collapse: collapse; }
    .table td,table th{border:1px solid #999; padding:5px;}
  </style>
  <t:pagelink page="crud/UserCreate">添加用户</t:pagelink><br/><br/>
  <table width="100%" border="0" cellspacing="0" cellpadding="0"
class="table">
    <tr>
      <th scope="col">id</th>
      <th scope="col">用户名</th>
      <th scope="col">年龄</th>
      <th scope="col">时间</th>
      <th scope="col">操作</th>
    </tr>
    <tr t:type="loop" t:source="users" t:value="user">
      <td>${user.id}</td>
      <td>${user.name}</td>
      <td>${user.time}</td>
      <td>${user.id}</td>
      <td></td>
    </tr>
  </table>
</html>
```

修改下 UserCreate.java 与 UserCreate.tml 文件，实现添加完成之后，直接跳转到 UserList 页面

#### UserCreate.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
```



```
*/  
package com.tapestry.app.pages.crud;  
  
import java.util.Date;  
  
import org.apache.tapestry5.annotations.Property;  
import org.apache.tapestry5.ioc.annotations.Inject;  
  
import com.tapestry.app.entities.User;  
import com.tapestry.app.services.StartDAO;  
  
public class UserCreate {  
  
    @Property  
    private User user;  
  
    //导入服务接口  
    @Inject  
    private StartDAO dao;  
  
    //初始化 user 实体  
    void onPrepare(){  
        user = new User();  
    }  
  
    //提交表单的时候执行存储,返回当前页面  
    Object onSuccess(){  
        //如果时间为空值输入系统当前时间  
        if(user.getTime() == null){  
            user.setTime(new Date());  
        }  
        dao.create(user);  
        return UserList.class;  
    }  
}
```

UserCreate.tml

```
<html      t:type="layout"      title="tapestryStart      Index"  
t:sidebarTitle="Framework Version"  
  xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"  
  xmlns:p="tapestry:parameter">  
  <t:form>  
    <t:errors/>
```



```
<p> 用 户 名 : <t:textfield t:id="name" value="user.name"
t:validate="required"/></p>
<p> 年 龄 : <t:textfield t:id="age" value="user.age"
t:validate="required"/></p>
<p><input type="submit" value="创建"/><t:pagelink page="crud/UserList">
返回查看页面</t:pagelink></p>
</t:form>
</html>
```

查看 <http://localhost:8080/tapestryStart/crud/userlist> 已经实现了数据的查询。

## 第四讲：tapestry 增删改查---修改数据

思路：在 UserList 页面传递 user 的 id 到修改也 UesrUpdate 页面，再查询出所有对应的数据。

修改 UserList.html 页面，代码如下：

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<style>
.table{border-collapse: collapse; }
.table td,table th{border:1px solid #999; padding:5px;}
</style>
<t:pagelink page="crud/UserCreate">添加用户</t:pagelink><br/><br/>
<table width="100%" border="0" cellpadding="0" cellspacing="0"
class="table">
<tr>
<th scope="col">id</th>
<th scope="col">用户名</th>
<th scope="col">年龄</th>
<th scope="col">时间</th>
<th scope="col">操作</th>
</tr>
<tr t:type="loop" t:source="users" t:value="user">
<td>${user.id}</td>
<td>${user.name}</td>
<td>${user.age}</td>
<td>${user.time}</td>
<td><t:pagelink page="crud/UserUpdate" t:context="${user.id}"> 修 改
```



```
</t:pagelink></td>
</tr>
</table>
</html>
```

在 com.tapestry.app.pages.crud 包里添加 UserUpdate.java，在 webapp/crud 下增加 UserUpdate.tml 代码如下：

UserUpdate.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages.crud;

import org.apache.tapestry5.annotations.PageActivationContext;
import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.ioc.annotations.Inject;

import com.tapestry.app.entities.User;
import com.tapestry.app.services.StartDAO;

public class UserUpdate {

    //接收页面传来的 id 值
    @PageActivationContext
    private Long id;

    //设置 user 可读写
    @Property
    private User user;

    //导入服务
    @Inject
    private StartDAO dao;

    //页面加载时运行
    void onPrepare(){
        //user 为空数据时根据页面传递过来的 id 查询数据
        if(user == null){
```



```
user = dao.findById(User.class, id);
}
}
//提交表单保存 user 数据
Object onSuccess(){
dao.update(user);
return UserList.class;
}
}
```

UserUpdate.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
  <t:form>
    <t:errors/>
    <p> 用 户 名 :    <t:textfield    t:id="name"    value="user.name"
t:validate="required"/></p>
    <p> 年 龄 :    <t:textfield    t:id="age"    value="user.age"
t:validate="required"/></p>
    <p><input type="submit" value="保存"/><t:pagelink page="crud/UserList">
返回查看页面</t:pagelink></p>
  </t:form>
</html>
```

查看 <http://localhost:8080/tapestryStart/crud/userlist> 修改数据也做好了。

## 第五讲：tapestry 增删改查---删除数据

思路：根据 user 的 id 去删除数据，直接从 UserList 页面实现删除。

修改 UserList.java，与 UserList.tml 文件。代码如下：

UserList.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
```



```
package com.tapestry.app.pages.crud;

import java.util.List;

import org.apache.tapestry5.annotations.PageActivationContext;
import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.ioc.annotations.Inject;

import com.tapestry.app.entities.User;
import com.tapestry.app.services.StartDAO;

public class UserList {

    //打开 user 读写
    @Property
    private User user;

    //打开 user 阵列的读写
    @Property
    private List<User> users;

    //导入操作数据库的服务
    @Inject
    private StartDAO dao;

    //当前页面接收 user 的 id 值
    @PageActivationContext
    private Long id;

    //页面加载时设置渲染
    void setupRender(){
        //查询 User 数据表
        StringBuffer sql = new StringBuffer();
        sql.append("from User");
        users = dao.findWithQuery(sql.toString());
    }

    //单击 eventlink 执行删除操作
    Object onDelete(Long id){
        dao.deleteByID(User.class, id);
        return this;
    }
}
```





```
}
```

UserList.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
  xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
  <style>
    .table{border-collapse: collapse; }
    .table td,table th{border:1px solid #999; padding:5px;}
  </style>
  <t:pagelink page="crud/UserCreate">添加用户</t:pagelink><br/><br/>
  <table  width="100%"  border="0"  cellspacing="0"  cellpadding="0"
class="table">
    <tr>
      <th scope="col">id</th>
      <th scope="col">用户名</th>
      <th scope="col">年龄</th>
      <th scope="col">时间</th>
      <th scope="col">操作</th>
    </tr>
    <tr t:type="loop" t:source="users" t:value="user">
      <td>${user.id}</td>
      <td>${user.name}</td>
      <td>${user.age}</td>
      <td>${user.time}</td>
      <td><t:pagelink page="crud/UserUpdate" t:context="${user.id}"> 修 改
</t:pagelink><t:eventlink t:event="delete" t:context="${user.id}"> 删 除
</t:eventlink></td>
    </tr>
  </table>
</html>
```

查看 <http://localhost:8080/tapestryStart/crud/userlist> 删除已经做好了

## 第六讲：tapestry 输出方式

tapestry 输出常用的有：通过 `get` 方法输出、通过 `output` 输出、通过 `setupRender()` 输出、通过 `outputraw` 输出。源码如下：



## OutPut.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages.output;

import java.text.Format;
import java.text.SimpleDateFormat;
import java.util.Date;

import org.apache.tapestry5.annotations.Property;

public class OutPut {

    private Date outOne;

    //通过 get 方法在页面上读出当前时间
    public Date getOutOne(){
        return new Date();
    }

    //通过 output 转换时间输出格式
    public Format getDateFormat(){
        return new SimpleDateFormat("yyyy-MM-dd HH:ss:mm");
    }

    //设置变量 outTwo 为可读,@Property 属性有2个就是分别对应 get 与 set 方法的 read
    与 write,
    //使用@Property 好处是可以不用写 get 与 set 方法。
    //注意: 不是任何地方都有效, 当 get 或 set 方法被调用的时候, 这样写就无效了, 因为
    找不到 get 或 set 方法。
    @Property(read=true)
    private String outTwo;

    //通过 setupRender 方法给 outTwo 赋值输出
    void setupRender(){
        outTwo = "通过 setupRender 方法给 outTwo 赋值输出";
    }
}
```



```
private String outThree;

//通过 get 方法读出 outThree 值,这里可以看到 outputraw 是会解析 html 节点的<br/>
public String getOutThree(){
    return "hello<br/>tapestry";
}

}
```

OutPut.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
  xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
  xmlns:p="tapestry:parameter">
  ${outOne}<br/><br/>
<t:output value="outOne" format="dateFormat"/><br/><br/>
  ${outTwo}<br/><br/>
  ${outThree}<br/><br/>
<t:outputraw value="${outThree}"/><br/><br/>
</html>
```

## 第七讲：tapestry 可预览的模板页

tapestry 模板页是\*.tml 文件，为了给美工美化静态页面带来更大方便，tapestry 的所有组件都可以 t:type 写成类似 htm 可预览的代码！。例如：

```
<t:pagelink t:page="Index">index</pagelink>
```

可写成

```
<a href="#" t:type="pagelink" t:page="Index">index</a>
```

其他的以此类推。

## 第八讲：tapestry 组件

tapestry 组件有不少，具体的使用方法可去官网了解，常用的组件会在以后的教程中讲到。这篇文章主要以 pagelink 为例，简单讲解下组件通常使用的方法，以下可看出同是跳转到 Index 页面，但写法就就不是一样！可选择其中自己的喜欢的方式写，所有组件都适用这样



的写法。代码如下：

AtComponent.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.Component;
import org.apache.tapestry5.corelib.components.PageLink;

public class AtComponent {

    @Component(id="myPage", parameters={"page=Index"})
    private PageLink index;
}
```

AtComponent.tml

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<a t:type="pagelink" t:page="Index" href="#;">index</a><br/>
<t:pagelink page="Index">index</t:pagelink><br/>
<a t:type="pagelink" t:id="myPage" href="#;">index</a><br/>
</html>
```

## 第九讲：tapestry pagelink 组件

tapestry 页面间的跳转使用了 pagelink 组件，等同于 html 的<a>节点。使用方法非常简单，具体使用上一讲已经讲过。



## 第十讲：tapestry 通过地址栏传值 @PageActivationContext

tapestry 通过地址栏传值常用的有：通过组件的 `context` 传值、通过 `activate()`, `passivate` 传值、通过 `@PageActivationContext` 传值。  
`@PageActivationContext` 等同于 `activate()`, `passivate`。代码如下：

ActivationContext.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.PageActivationContext;
import org.apache.tapestry5.annotations.Property;

public class ActivationContext {

    @Property
    private String strOne;

    //传递 strOne 到浏览器地址栏
    String onPassivate(){
        return strOne;
    }

    //从地址栏中获得 strOne 值，注意：当传递的值是中文是会被转码，所以看不到中文。
    //当做检索的时候想通过地址栏传递查询变量时，可使用 ajax 不刷新查看中文。
    void onActivate(String strOne){
        this.strOne = strOne;
    }

    // @PageActivationContext 是 onPassivate()与 onActivate()2个函数的封装。
    // 使用 @PageActivationContext 就不用在写 onPassivate()与 onActivate()了
}
```



```
//跟@Property 类似。  
@Property  
@PageActivationContext  
private String strTwo;  
}
```

ActivationContext.tml

```
<html t:type="layout" title="tapestryStart Index"  
t:sidebarTitle="Framework Version"  
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"  
xmlns:p="tapestry:parameter">  
<a t:type="pagelink" t:page="ActivationContext" t:context="literal:5">把5  
传递到 ActivationContext 页面</a><br/><br/><br/>  
<t:form t:id="form1">  
<input t:type="textfield" t:id="strOne" value="strOne"/>  
<input type="submit" value="传递 strOne 值"/>  
</t:form>  
<t:form t:id="form2">  
<input t:type="textfield" t:id="strTwo" value="strTwo"/>  
<input type="submit" value="传递 strTwo 值"/>  
</t:form>  
</html>
```

## 第十一讲：tapestry session 存储

tapestry session 通常有两种方式：persist 与 sso，persist 方法虽然简单，但是过多的使用会增加服务器的负担，同时其 URL 也无法被做为书签保存。sso 被保存在 Session 中，可以被同一用户的所有页面共享，但不会被其他用户共享，通过@SessionState 标记，做用户登录使用。源码如下：

MyPersist.java

```
/**  
 * 项目名称: TapestryStart  
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql  
 * 网址: http://www.flywind.org  
 * 版本: 1.0  
 * 编写: 飞风  
 * 时间: 2012-02-29  
 */  
package com.tapestry.app.pages;
```



```
import org.apache.tapestry5.annotations.Persist;
import org.apache.tapestry5.annotations.Property;

public class MyPersist {

    //直接把 hello 存储到 session
    @Persist
    @Property
    private String hello;

}
```

#### MyPersist.tml

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
    您输入的值: ${hello}<br/>
    <form t:type="form">
        <input t:type="textfield" t:id="hello" value="hello"/>
        <input type="submit" value="提交"/>
    </form>
</html>
```

#### Signin.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.annotations.SessionState;

public class Signin {

    @Property
    private String userName;
```



```
@Property
private String password;

@SessionState
private String user;

Object onSuccess(){
    if("fly".equals(userName) && "123".equals(password)){
        user = userName;
        return Welcome.class;
    }
    return null;
}
}
```

#### Signin.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
  xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
  xmlns:p="tapestry:parameter">
  <t:form>
    用户名: <input t:type="textfield" t:id="userName" value="userName"/><br/>
    密码: <input t:type="textfield" t:id="password" value="password"/><br/>
    <input type="submit" value="登录"/>
  </t:form>
</html>
```

#### Welcome.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.annotations.SessionState;

public class Welcome {
```





```
@Property
@SessionState
private String userName;
```

```
Object onLogout(){
    userName = null;
    return Signin.class;
}
}
```

Welcome.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
  xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
  xmlns:p="tapestry:parameter">
  ${userName}, 欢迎您! <t:eventlink t:event="logOut">退出</t:eventlink>
</html>
```

## 第十二讲：tapestry actionlink 与 eventlink 组件

actionlink 与 eventlink 组件非常相似，都是执行一个事件，在代码编写上有一点点差异。源代码如下：

MyEvent.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.PageActivationContext;
import org.apache.tapestry5.annotations.Property;
```



```
public class MyEvent {

    @Property
    @PageActivationContext
    private int count;

    //执行 eventlink 递增加+1,newCount 就是默认值1
    void onAdd(int newCount){
        count += newCount;
    }

    //执行 actionlink 递增加+2,meCount 就是默认值2
    void onActionFromAddTwo(int meCount){
        count += meCount;
    }

    //设置为0
    void onClear(){
        count = 0;
    }
}
```

MyEvent.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
  xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
  xmlns:p="tapestry:parameter">
<div style="color:#ff6600">${count}</div>
<t:eventlink t:event="add" t:context="literal:1">单击递增1，初始值为
1</t:eventlink><br/>
<t:actionlink t:id="addTwo" t:context="literal:2">单击递增2，初始值为
2</t:actionlink><br/>
<a href="#" t:type="eventlink" t:event="clear">清除</a>
</html>
```

## 第十三讲：tapestry form 组件

form 组件，源码如下：

Forms1.java

```
/**
```



```
* 项目名称: TapestryStart
* 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
* 网址: http://www.flywind.org
* 版本: 1.0
* 编写: 飞风
* 时间: 2012-02-29
*/
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.Component;
import org.apache.tapestry5.annotations.InjectPage;
import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.corelib.components.Form;
import org.apache.tapestry5.corelib.components.TextField;

public class Forms1 {

    @Property
    private String firstName;

    @Property
    private String lastName;

    @Component(id="myForm")
    private Form form;

    @Component(id="firstName")
    private TextField firstNameField;

    @Component(id="lastName")
    private TextField lastNameField;

    @InjectPage
    private Forms2 forms2;

    void onValidateFromMyForm(){
        if(firstName == null || firstName.trim().equals("")){
            form.recordError(firstNameField, "姓不能为空值!");
        }
        if(lastName == null || lastName.trim().equals("")){
            form.recordError(lastNameField, "名字不能为空值!");
        }
    }
}
```



```
Object onSuccess(){
forms2.setFirstName(firstName);
forms2.setLastName(lastName);
return forms2;
}
}
```

#### Forms1.tml

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<t:form t:id="myForm">
<t:errors/>
姓: <input t:type="textfield" t:id="firstName" value="firstName"/><br/>
名: <input t:type="textfield" t:id="lastName" value="lastName"/><br/>
<input type="submit" value="提交"/>
</t:form>
</html>
```

#### Forms2.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.Persist;

public class Forms2 {

    @Persist
    private String firstName;

    @Persist
    private String lastName;

    public String getLastName() {
```



```
return lastName;
}

public void setLastName(String lastName) {
    this.lastName = lastName;
}

public String getFirstName() {
    return firstName;
}

public void setFirstName(String firstName) {
    this.firstName = firstName;
}
}
```

Forms2.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
  xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
  xmlns:p="tapestry:parameter">
  用户名: ${firstName}${lastName}
  <t:pagelink page="Forms1">返回</t:pagelink>
</html>
```

## 第十四讲：tapestry form 验证

tapestry 提供了表单验证，大大简化了我们手工编写验证代码的麻烦。常用验证有：required(必须的)、email(邮箱验证)、maxLength(最大字符数)、minLength(最小字符数)、max(最大值)、min(最小值)、regex(正则表达式)。奇怪的是5.3.1验证有 bug，5.3.2已经修复了....源码如下：

Validators1.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
```



```
*/  
package com.tapestry.app.pages;  
  
import org.apache.tapestry5.annotations.InjectPage;  
import org.apache.tapestry5.annotations.Property;  
import org.apache.tapestry5.beaneditor.Validate;  
  
public class Validators1 {  
  
    @Property  
    @Validate("required")  
    private String firstName;  
  
    @Property  
    @Validate("required")  
    private String lastName;  
  
    @InjectPage  
    private Validators2 validators2;  
  
    Object onSuccess(){  
        validators2.set(firstName, lastName);  
        return validators2;  
    }  
}
```

#### Validators1.tml

```
<html t:type="layout" title="tapestryStart Index"  
t:sidebarTitle="Framework Version"  
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"  
xmlns:p="tapestry:parameter">  
    <t:form>  
        <t:errors/>  
        姓: <input t:type="textField" t:id="firstName" value="firstName"/><br/>  
        名字: <input t:type="textField" t:id="lastName" value="lastName"/><br/>  
        <input type="submit" value="提交"/>  
    </t:form>  
</html>
```

#### Validators2.java

```
/**  
 * 项目名称: TapestryStart  
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
```



```
* 网址: http://www.flywind.org
* 版本: 1.0
* 编写: 飞风
* 时间: 2012-02-29
*/
package com.tapestry.app.pages;

public class Validators2 {

    private String firstName;

    private String lastName;

    public void set(String firstName, String lastName){
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public String getName(){
        return firstName + " " + lastName;
    }

    String[] onPassivate(){
        return new String[]{firstName, lastName};
    }

    void onActivate(String firstName, String lastName){
        this.firstName = firstName;
        this.lastName = lastName;
    }

}
```

#### Validators2.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
  xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
  xmlns:p="tapestry:parameter">
    姓名: ${name}<br/>
    <t:pagelink page="Validators1">返回</t:pagelink>
</html>
```



## 第十五讲：tapestry form 自定义验证

自定义表单比较复杂，编写了几个组件 CustomForm 组件、CustomError 组件、CustomValidationDecorator 自定义验证该文件使用了@Override 如果报错请选择项目的 jdk 版本为1.6。代码如下：

CustomForm.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.components;

import org.apache.tapestry5.Field;
import org.apache.tapestry5.FormValidationControl;
import org.apache.tapestry5.MarkupWriter;
import org.apache.tapestry5.ValidationDecorator;
import org.apache.tapestry5.annotations.Component;
import org.apache.tapestry5.annotations.SupportsInformalParameters;
import org.apache.tapestry5.corelib.components.Form;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.services.Environment;

import com.tapestry.app.commons.CustomValidationDecorator;

@SupportsInformalParameters
public class CustomForm implements FormValidationControl {

    @Inject
    private Environment environment;

    @Component(id = "form", publishParameters = "context, tracker,
    clientValidation, zone, secure, validationId, validate, class",
    inheritInformalParameters = true)
    private Form form;
```





```
void beginRender(MarkupWriter writer) {
    environment.push(ValidationDecorator.class,
CustomValidationDecorator(environment, writer));
}

void afterRender(MarkupWriter writer) {
    environment.pop(ValidationDecorator.class);
}

@Override
public void clearErrors() {
    form.clearErrors();
}

@Override
public boolean getHasErrors() {
    return form.getHasErrors();
}

@Override
public boolean isValid() {
    return form.isValid();
}

@Override
public void recordError(String errorMessage) {
    form.recordError(errorMessage);
}

@Override
public void recordError(Field field, String errorMessage) {
    form.recordError(field, errorMessage);
}
}
```

#### CustomForm.tml

```
<html xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd">
<t:content>
<t:form t:id="form" autoFocus="false">
<t:body/>
</t:form>
</t:content>
```



</html>

#### CustomError.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.components;

import org.apache.tapestry5.BindingConstants;
import org.apache.tapestry5.Field;
import org.apache.tapestry5.MarkupWriter;
import org.apache.tapestry5.annotations.Environmental;
import org.apache.tapestry5.annotations.Import;
import org.apache.tapestry5.annotations.Parameter;
import org.apache.tapestry5.dom.Element;
import org.apache.tapestry5.services.Heartbeat;

@Import(library = "CustomError.js")
public class CustomError {

    @Parameter(name = "for", required = true, allowNull = false, defaultPrefix
    = BindingConstants.COMPONENT)
    private Field field;

    @Environmental
    private Heartbeat heartbeat;

    private Element labelElement;

    boolean beginRender(MarkupWriter writer) {
        final Field field = this.field;

        labelElement = writer.element("span", "class", "msg");

        Runnable command = new Runnable() {
            public void run() {
                String fieldId = field.getClientId();
                labelElement.forceAttributes("id", fieldId + "-msg");
            }
        };
    }
}
```



```
}  
};  
  
heartbeat.defer(command);  
  
return false;  
}  
  
void afterRender(MarkupWriter writer) {  
writer.end();  
}  
}
```

#### CustomValidationDecorator.java

```
/**  
 * 项目名称: TapestryStart  
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql  
 * 网址: http://www.flywind.org  
 * 版本: 1.0  
 * 编写: 飞风  
 * 时间: 2012-02-29  
 */  
package com.tapestry.app.common;  
  
import org.apache.tapestry5.BaseValidationDecorator;  
import org.apache.tapestry5.Field;  
import org.apache.tapestry5.MarkupWriter;  
import org.apache.tapestry5.ValidationTracker;  
import org.apache.tapestry5.corelib.components.RadioGroup;  
import org.apache.tapestry5.dom.Element;  
import org.apache.tapestry5.services.Environment;  
  
public class CustomValidationDecorator extends BaseValidationDecorator {  
  
private final Environment environment;  
private final MarkupWriter markupWriter;  
  
public CustomValidationDecorator(Environment environment, MarkupWriter  
markupWriter) {  
this.environment = environment;  
this.markupWriter = markupWriter;  
}  
}
```



```
@Override
public void insideLabel(Field field, Element element) {
    final String radioGroupClassName = RadioGroup.class.getName();

    if (field == null) {
        return;
    }

    if (!field.getClass().getName().equals(radioGroupClassName)) {
        if (field.isRequired()) {
            element.addClassName("required-label");
            element.getContainer().addClassName("required-label-c");
        }
    }

    if (inError(field)) {
        element.addClassName("error-label");
        element.getContainer().addClassName("error-label-c");
    }
}

@Override
public void insideField(Field field) {

    if (field.isRequired()) {
        getElement().addClassName("required-field");
        getElement().getContainer().addClassName("required-field-c");
    }

    if (inError(field)) {
        getElement().addClassName("error-field");
        getElement().getContainer().addClassName("error-field-c");
    }
}

private boolean inError(Field field) {
    ValidationTracker tracker = getTracker();
    return tracker.inError(field);
}

private ValidationTracker getTracker() {
    return environment.peekRequired(ValidationTracker.class);
}
```



```
private Element getElement() {  
    return markupWriter.getElement();  
}  
}
```

#### CustomError.js

```
Tapestry.FieldEventManager.addMethods( {  
  
    initialize : function(field) {  
        this.field = $(field);  
  
        var id = this.field.id;  
        this.fieldContainer = this.field.parentNode;  
        Element.extend(this.fieldContainer);  
  
        var labelSelector = "label[for='" + id + "']";  
        this.label = this.field.up("form").down(labelSelector);  
  
        if (this.label) {  
            this.labelContainer = this.label.parentNode;  
            Element.extend(this.labelContainer);  
        }  
  
        var msgId = id + '-msg';  
        this.msg = $(msgId);  
  
        if (!this.msg) {  
            this.msg = new Element('strong', {  
                'id' : msgId,  
                'class' : 'msg'  
            });  
            this.fieldContainer.insert( {  
                bottom : this.msg  
            });  
        }  
  
        this.msgContainer = this.msg.parentNode;  
        Element.extend(this.msgContainer);  
  
        this.translator = Prototype.K;  
  
        var fem = $(this.field.form).getFormEventManager();
```



```
if (fem.validateOnBlur) {

document.observe(Tapestry.FOCUS_CHANGE_EVENT, function(event) {
/*
* If changing focus *within the same form* then perform
* validation. Note that Tapestry.currentFocusField does not
* change until after the FOCUS_CHANGE_EVENT notification.
*/
if (Tapestry.currentFocusField == this.field
&& this.field.form == event.memo.form)
this.validateInput();

}.bindAsEventListener(this));
}

if (fem.validateOnSubmit) {
$(this.field.form).observe(Tapestry.FORM_VALIDATE_FIELDS_EVENT,
this.validateInput.bindAsEventListener(this));
}
},

showValidationMessage : function(message) {
$(this.field).getStorage().validationError = true;
$(this.field.form).getStorage().validationError = true;

this.msg.update(message);

if (this.label) {
this.label.addClassName("error-label");
this.labelContainer.addClassName("error-label-c");
}
this.field.addClassName("error-field");
this.fieldContainer.addClassName("error-field-c");
this.msg.addClassName("error-msg");
this.msgContainer.addClassName("error-msg-c");
},
```



```
removeDecorations : function() {

this.msg.update(null);

if (this.label) {
this.label.removeClassName("error-label");
this.labelContainer.removeClassName("error-label-c");
}
this.field.removeClassName("error-field");
this.fieldContainer.removeClassName("error-field-c");
this.msg.removeClassName("error-msg");
this.msgContainer.removeClassName("error-msg-c");
}
} );
```

以上就是自定义验证所有需要的文件。以下看下使用自定义验证的代码：

#### CValidators1.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.InjectPage;
import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.beaneditor.Validate;

public class CValidators1 {

@Property
@Validate("required")
private String firstName;

@Property
@Validate("required")
private String lastName;
```



```
@InjectPage
private CValidators2 validators2;

Object onSuccess(){
    validators2.set(firstName, lastName);
    return validators2;
}
}
```

#### CValidators1.tml

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
    <style>
        .error-msg{color:red}
    </style>
    <form t:type="CustomForm">
        <t:errors/>
        姓 : <input t:type="textField" t:id="firstName"
value="firstName"/><t:CustomError for="firstName"/><br/>
        名字 : <input t:type="textField" t:id="lastName"
value="lastName"/><t:CustomError for="lastName"/><br/>
        <input type="submit" value="提交"/>
    </form>
</html>
```

#### CValidators2.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

public class CValidators2 {

    private String firstName;
```





```
private String lastName;

public void set(String firstName, String lastName){
    this.firstName = firstName;
    this.lastName = lastName;
}

public String getName(){
    return firstName + " " + lastName;
}

String[] onPassivate(){
    return new String[]{firstName, lastName};
}

void onActivate(String firstName, String lastName){
    this.firstName = firstName;
    this.lastName = lastName;
}

}
```

CValidators2.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
  xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
  xmlns:p="tapestry:parameter">
    姓名: ${name}<br/>
    <t:pagelink page="CValidators1">返回</t:pagelink>
</html>
```

## 第十六讲：tapestry 数据循环显示 loop 与 grid 组件

tapestry 提供了2种常用的数据循环显示：loop 组件与 grid 组件，两者没多大区别，loop 组件没有提供分页，使用比较灵活，grid 组件提供了分页功能，相反自己定义方面没那么灵活，但提供了字段排序的功能。此例子写的比较复杂，因为之前不知道怎么使用 loop 组件显示



string[]数组，因此把例子写下来便于以后查看，其实很简单的首先要把 string[]转变成 List<String>再使用 loop 组件显示。源码如下：

DataList.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import java.util.ArrayList;
import java.util.List;

import org.apache.tapestry5.annotations.Property;

public class DataList {

    @Property
    private String nameOne;

    @Property
    private String nameTwo;

    @Property
    private String[] nameOnes;

    @Property
    private String[] nameTwos;

    @Property
    private List<String> oneLists;

    @Property
    private List<String> twoLists;

    void setupRender(){
        String oneString = "张三/李四/王五/";
        nameOnes = oneString.split("/");
        oneLists = new ArrayList<String>();
    }
}
```



```
for(String str:nameOnes){
oneLists.add(str);
}

String twoString = "刘备/张飞/关于/";
nameTwos = twoString.split("/");
twoLists = new ArrayList<String>();
for(String str2:nameTwos){
twoLists.add(str2);
}
}
}
```

DataList.html

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
  xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
  xmlns:p="tapestry:parameter">
  name<br/>
<t:loop source="oneLists" value="nameOne">
${nameOne}<br/>
</t:loop>
<t:grid source="twoLists" add="name" row="nameTwo">
<p:NameCell>
${nameTwo}<br/>
</p:NameCell>
</t:grid>
</html>
```

## 第十七讲: **tapestry** 不同类型的变量在页面之间值的传递

页面之间传递值，以前已经讲过，这里不一样的是变量的类型不一样了，具体看源码吧：

Translators1.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
```



```
* 版本: 1.0
* 编写: 飞风
* 时间: 2012-02-29
*/
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.InjectPage;
import org.apache.tapestry5.annotations.Property;

public class Translators1 {
    @Property
    private Integer integerField;

    @Property
    private Long longField;

    @Property
    private Double doubleField;

    @Property
    private String stringField;

    // Other pages

    @InjectPage
    private Translators2 page2;

    // The code

    Object onSuccess() {
        page2.set(integerField, longField, doubleField, stringField);
        return page2;
    }
}
```

#### Translators1.tml

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
    <form t:type="form" t:id="inputs">
    <t:errors/>
    <table>
```



```
<tr>
<td>Integer 类型.</td>
<td><t:label for="integerField"/></td>
<td><input t:type="TextField" t:id="integerField"
t:value="integerField"/></td>
</tr>
<tr>
<td>Long 类型.</td>
<td><t:label for="longField"/></td>
<td><input t:type="TextField" t:id="longField" t:value="longField"/></td>
</tr>
<tr>
<td>Double 类型.</td>
<td><t:label for="doubleField"/></td>
<td><input t:type="TextField" t:id="doubleField"/></td>
</tr>
<tr>
<td>String 类型.</td>
<td><t:label for="stringField"/></td>
<td><input t:type="TextField" t:id="stringField"
t:value="stringField"/></td>
</tr>
<tr>
<td></td>
<td></td>
<td><input type="submit" value="传递"/></td>
</tr>
</table>
</form>
</html>
```

#### Translators2.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.Property;
```



```
public class Translators2 {
    @Property(write = false)
    private Integer integerField;

    @Property(write = false)
    private Long longField;

    @Property(write = false)
    private Double doubleField;

    @Property(write = false)
    private String stringField;

    public void set(Integer integerField, Long longField, Double doubleField,
String stringField) {
        this.integerField = integerField;
        this.longField = longField;
        this.doubleField = doubleField;
        this.stringField = stringField;
    }

    Object[] onPassivate() {
        return new Object[] { integerField, longField, doubleField, stringField };
    }

    void onActivate(Integer integerField, Long longField, Double doubleField,
String stringField) {
        this.integerField = integerField;
        this.longField = longField;
        this.doubleField = doubleField;
        this.stringField = stringField;
    }
}
```

#### Translators2.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
  xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
  xmlns:p="tapestry:parameter">
<table>
<tr>
<td>integer 类型:</td><td>${integerField}</td>
```



```
</tr>
<tr>
<td>long 类型:</td><td>${longField}</td>
</tr>
<tr>
<td>double 类型:</td><td>${doubleField}</td>
</tr>
<tr>
<td>string 类型:</td><td>${stringField}</td>
</tr>
</table>
</html>
```

## 第十八讲：tapestry 表单核心组件

源码如下：

InputCore.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import java.util.ArrayList;
import java.util.Date;
import java.util.List;

import org.apache.tapestry5.annotations.Persist;
import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.internal.services.StringValueEncoder;

public class InputCore {
    @Property
    @SuppressWarnings("unused")
    private StringValueEncoder stringEncoder = new StringValueEncoder();
```



```
/* Checkbox */

@property
@Persist
@SuppressWarnings("unused")
private boolean checkboxValue;

/* Checklist */

@property
@Persist
@SuppressWarnings("unused")
private List<String> checklistSelectedValues;

@property
@SuppressWarnings("unused")
private final String[] STATIONERY = { "Pens", "Pencils", "Paper" };

/* DateField */

@property
@Persist
private Date dateValue;

// We could return a DateFormat, but instead we'll return a String which
// DateField will coerce into a DateFormat.
public String getDateFieldFormat() {
    return "dd/MM/yyyy";
}

/* Palette */

@property
@Persist
private List<String> paletteSelectedValues;

@property
@SuppressWarnings("unused")
private final String[] PETS = { "Dog", "Cat", "Parrot", "Mouse" };

/* Password */

@property
```





```
@Persist
@SuppressWarnings("unused")
private String passwordValue;

/* RadioGroup and Radio */

@property
@Persist
@SuppressWarnings("unused")
private String radioSelectedValue;

/* Select */

@property
@Persist
@SuppressWarnings("unused")
private String selectedValue;

/* TextArea */

@property
@Persist
@SuppressWarnings("unused")
private String textAreaValue;

/* TextField */

@property
@Persist
@SuppressWarnings("unused")
private String textValue;

/* Life-cycle stuff. Fields that are marked @Persist MUST be initialized
here rather than where they are declared. */

void setupRender() {
    if (dateValue == null) {
        dateValue = new Date();
    }
    if (paletteSelectedValues == null) {
        paletteSelectedValues = new ArrayList<String>();
    }
}
```



```
}
```

#### InputCore.tml

```
<html          t:type="layout"          title="tapestryStart      Index"
t:sidebarTitle="Framework Version"
  xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
  xmlns:p="tapestry:parameter">
<style type="text/css">
body { font-family: Arial, Helvetica, sans-serif; font-size: 13px; }
form { text-align: center; }
table { font-family: Arial, Helvetica, sans-serif;  font-size: 13px;
font-weight: normal; margin: auto; }
th { background-color: #bee; text-align: center; padding: 10px 10px; }
td { background-color: #eeb; padding: 10px 10px; }
td.name { text-align: right; vertical-align: top; }
div.t-palette { width: 460px; }
.datePicker td .topLabel { color:black; }
</style>
<form t:type="form" t:id="inputs">
<table>
<tr>
<th>Component Name</th>
<th>Use them and press Submit</th>
<th>Submitted Value</th>
</tr>
<tr>
<td class="name">Checkbox:</td>
<td><input t:type="checkbox" value="checkboxValue"/></td>
<td>${checkboxValue}</td>
</tr>
<tr>
<td class="name">Checklist:</td>
<td><input          t:type="Checklist"          t:model="stationery"
t:selected="checklistSelectedValues" t:encoder="stringEncoder"/></td>
<td>${checklistSelectedValues}</td>
</tr>
<tr>
<td class="name">DateField:</td>
<td><input          t:type="DateField"          value="dateValue"
t:format="prop:dateFieldFormat"/> (${dateFieldFormat})</td>
<td>${dateValue}</td>
</tr>
<tr>
```



```
<td class="name">Palette:</td>
<td><input                                t:type="Palette"                t:model="pets"
t:selected="paletteSelectedValues" t:encoder="stringEncoder"/></td>
<td>${paletteSelectedValues}</td>
</tr>
<tr>
<td class="name">PasswordField:</td>
<td><input t:type="PasswordField" value="passwordValue"/></td>
<td>${passwordValue}</td>
</tr>
<tr>
<td class="name">RadioGroup,<br/> Radio,<br/> and Label:</td>
<td>
<t:RadioGroup value="radioSelectedValue" encoder="stringEncoder">
<t:Radio t:id="radioH" value="literal:H" label="High"/><t:label
for="radioH"/><br/>
<t:Radio t:id="radioM" value="literal:M" label="Medium"/><t:label
for="radioM"/><br/>
<t:Radio t:id="radioL" value="literal:L" label="Low"/><t:label
for="radioL"/>
</t:RadioGroup>
</td>
<td>${radioSelectedValue}</td>
</tr>
<tr>
<td class="name">Select:</td>
<td><select t:type="Select" t:model="literal:Ms,Mrs,Mr,Dr,Prof"
t:value="selectedValue"/></td>
<td>${selectedValue}</td>
</tr>
<tr>
<td class="name">TextArea:</td>
<td><input t:type="TextArea" value="textAreaValue"/></td>
<td><t:textoutput value="textAreaValue"/></td>
</tr>
<tr>
<td class="name">TextField:</td>
<td><input t:type="TextField" value="textValue"/></td>
<td>${textValue}</td>
</tr>
</table><br/>

<input type="submit" value="Submit"/>
```



```
</form>
</html>
```

## 第十九讲：tapestry 多个提交按钮(submit)

多个提交按钮。源码如下：

MultipleSubmits1.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.InjectPage;
import org.apache.tapestry5.annotations.Property;

import com.tapestry.app.pages.MultipleSubmits2.SearchType;

public class MultipleSubmits1 {
    @Property
    private String name;

    @InjectPage
    private MultipleSubmits2 page2;

    private SearchType searchType;

    void onActivate() {
        searchType = SearchType.CUSTOMERS;
    }

    void onSelectedFromSuppliers() {
        searchType = SearchType.SUPPLIERS;
    }
}
```



```
Object onSuccess() {  
    page2.set(searchType, name);  
    return page2;  
}  
}
```

#### MultipleSubmits1.tml

```
<html          t:type="layout"          title="tapestryStart          Index"  
t:sidebarTitle="Framework Version"  
  xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"  
  xmlns:p="tapestry:parameter">  
  
  <form t:type="form" t:id="search" style="border: medium solid gray">  
    <t:errors/>  
    <table>  
      <tr>  
        <td>名字:</td>  
        <td><input      t:type="TextField"      t:id="name"      t:validate="required,  
maxlength=10" size="10"/></td>  
      </tr>  
      <tr>  
        <td></td>  
        <td>  
          <input type="submit" value="Search Customers"/>  
          <input t:type="Submit" t:id="suppliers" value="Search Suppliers"/>  
        </td>  
      </tr>  
    </table>  
    </form>  
  </html>
```

#### MultipleSubmits2

```
/**  
 * 项目名称: TapestryStart  
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql  
 * 网址: http://www.flywind.org  
 * 版本: 1.0  
 * 编写: 飞风  
 * 时间: 2012-02-29  
 */  
package com.tapestry.app.pages;
```



```
public class MultipleSubmits2 {
private SearchType searchType;

private String name;

public enum SearchType {
CUSTOMERS, SUPPLIERS;
}

public void set(SearchType searchType, String lastName) {
this.searchType = searchType;
this.name = lastName;
}

Object[] onPassivate() {
return new Object[] { searchType, name };
}

void onActivate(SearchType searchType, String name) {
this.searchType = searchType;
this.name = name;
}

public String getYourSearch() {
if (searchType == SearchType.CUSTOMERS) {
return "你选择 Customers 名字 \"" + name + "\".";
}
else {
return "你选择 Suppliers 名字 \"" + name + "\".";
}
}
}
```

MultipleSubmits2.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
  xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
${yourSearch}
</html>
```



## 第二十讲：tapestry 多表单(forms)

源码如下：

MultipleForms1.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.InjectPage;
import org.apache.tapestry5.annotations.Property;

import com.tapestry.app.pages.MultipleForms2.SearchType;

public class MultipleForms1 {
    @Property
    private String customerName;

    @Property
    private String supplierName;

    @InjectPage
    private MultipleForms2 page2;

    void onPrepareFromSearchCustomers() {
        //可初始化 Customers 表单元素
    }

    void onPrepareFromSearchSuppliers() {
        //可初始化 Suppliers 表单元素
    }

    Object onSuccessFromSearchCustomers() {
        page2.set(SearchType.CUSTOMERS, customerName);
        return page2;
    }
}
```



```
Object onSuccessFromSearchSuppliers() {  
    page2.set(SearchType.SUPPLIERS, supplierName);  
    return page2;  
}  
}
```

#### MultipleForms1.tml

```
<html          t:type="layout"          title="tapestryStart          Index"  
t:sidebarTitle="Framework Version"  
  xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"  
  xmlns:p="tapestry:parameter">  
  
  <form t:type="form" t:id="searchcustomers" style="border: medium solid  
gray">  
    <t:errors/>  
    Customer Name:  
    <input  t:type="TextField"  t:id="customerName"  t:validate="required,  
maxlength=10" size="10"/>  
    <input type="submit" value="Search Customers"/>  
  </form><br/>  
  
  <form t:type="form" t:id="searchsuppliers" style="border: medium solid  
gray">  
    <t:errors/>  
    Supplier Name:  
    <input  t:type="TextField"  t:id="supplierName"  t:validate="required,  
maxlength=10" size="10"/>  
    <input type="submit" value="Search Suppliers"/>  
  </form><br/>  
</html>
```

#### MultipleForms2.java

```
/**  
 * 项目名称: TapestryStart  
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql  
 * 网址: http://www.flywind.org  
 * 版本: 1.0  
 * 编写: 飞风  
 * 时间: 2012-02-29  
 */  
package com.tapestry.app.pages;
```





```
public class MultipleForms2 {
    private SearchType searchType;

    private String name;

    public enum SearchType {
        CUSTOMERS, SUPPLIERS;
    }

    public void set(SearchType searchType, String lastName) {
        this.searchType = searchType;
        this.name = lastName;
    }

    Object[] onPassivate() {
        return new Object[] { searchType, name };
    }

    void onActivate(SearchType searchType, String lastName) {
        this.searchType = searchType;
        this.name = lastName;
    }

    public String getYourSearch() {
        if (searchType == SearchType.CUSTOMERS) {
            return "你选择 Customers 名字为: \"" + name + "\".";
        }
        else {
            return "你选择 Suppliers 名字为: \"" + name + "\".";
        }
    }
}
```

#### MultipleForms2.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
    xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
    xmlns:p="tapestry:parameter">
    ${yourSearch}<br/><br/>
    <a t:type="pagelink" t:page="MultipleForms1" href="#">返回</a>
</html>
```



## 第二十一讲：tapestry 的 LinkSubmit 等同于 submit 来使用

tapestry 的 LinkSubmit 其实就等同于一个 submit 按钮，它的存在可使前端开发很容易的修改<a>样式，非常容易做出好的按钮样式。源码如下：

LinkSubmits1.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.Component;
import org.apache.tapestry5.annotations.Persist;
import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.corelib.components.Form;
import org.apache.tapestry5.corelib.components.TextField;

public class LinkSubmits1 {
    @Persist
    @Property
    private String firstName;

    @Persist
    @Property
    private String lastName;

    @Persist
    @Property
    private String selectName;

    @Component(id = "names")
    private Form form;

    @Component(id = "firstName")
```



```
private TextField firstNameField;

@Component(id = "lastName")
private TextField lastNameField;

void onValidateFromNames() {
    if (firstName == null || firstName.trim().equals("")) {
        form.recordError(firstNameField, "First Name is required.");
    }
    if (lastName == null || lastName.trim().equals("")) {
        form.recordError(lastNameField, "Last Name is required.");
    }
}

Object onSuccess() {
    selectName = firstName + lastName;
    return this;
}
}
```

#### LinkSubmits1.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
  xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
  xmlns:p="tapestry:parameter">
  ${selectName}
  <form t:type="form" t:id="names" style="border: medium solid gray">
  <t:errors/>
  <table>
  <tr>
  <td>姓:</td>
  <td><input          t:type="TextField"          t:id="firstName"
t:validate="required"/></td>
  </tr>
  <tr>
  <td>名:</td>
  <td><input          t:type="TextField"          t:id="lastName"
t:validate="required"/></td>
  </tr>
  <tr>
  <td></td>
  <td><t:linksubmit t:id="newlinksubmit">提交显示姓名</t:linksubmit></td>
  </tr>
```



```
</table>
  </form>
</html>
```

## 第二十二讲：tapestry 的 Select 组件

Select 组件，具体效果直接运行实例吧，源码如下：

VariedSelect.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.Property;

public class VariedSelect {
    @Property
    private String color0;

    @Property
    private String color1;

    @Property
    private String color2;

    @Property
    private String color3;

    String[] onPassivate() {
        return new String[] { color0, color1, color2, color3 };
    }

    void onActivate(String color0, String color1, String color2, String color3)
    {
        this.color0 = color0;
    }
}
```



```
this.color1 = color1;
this.color2 = color2;
this.color3 = color3;
}
}
```

#### VariedSelect.tml

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<form t:type="form">
This example demonstrates labelling the blank option.<br/><br/>

<t:label for="color0"/>:
<select t:type="select" t:id="color0" t:model="literal:Red,Yellow,Blue"
t:blankLabel="Choose..."></select>
<input type="submit" value="Submit"/>
You chose: ${color0}
</form><br/>

<form t:type="form">
This example demonstrates labelling the blank option using the page's
properties file.<br/><br/>

<t:label for="color1"/>:
<select t:type="select" t:id="color1"
t:model="literal:Red,Yellow,Blue"></select>
<input type="submit" value="Submit"/>
You chose: ${color1}
</form><br/>

<form t:type="form">
This example demonstrates making it mandatory to make a choice.<br/><br/>

<t:label for="color2"/>:
<select t:type="select" t:id="color2" t:model="literal:Red,Yellow,Blue"
t:validate="required"></select>
<input type="submit" value="Submit"/>
You chose: ${color2}
</form><br/>

<form t:type="form">
```



This example demonstrates making it mandatory to make a choice - but it also has a blank option.<br/><br/>

```
<t:label for="color3"/>:
<select t:type="select" t:id="color3" t:model="literal:Red,Yellow,Blue"
t:validate="required" t:blankOption="ALWAYS"></select>
<input type="submit" value="Submit"/>
You chose: ${color3}
</form><br/>
</html>
```

## 第二十三讲: tapestry 的 Delegate 和 Blocks 组件

源码如下:

DelegateAndBlocks.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

public class DelegateAndBlocks {

}
```

DelegateAndBlocks.tml

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<t:delegate to="block:blockA"/>
<t:delegate to="block:blockB"/>
<t:delegate to="block:blockC"/>
```



```
<t:delegate to="block:blockB"/>
<t:delegate to="block:blockA"/> <br/>
<t:block id="blockA">
<div style="margin: 2px; padding: 10px; background-color: #f8f8f8;">
This is block A.
</div>
</t:block>
<t:block id="blockB">
<div style="margin: 2px; padding: 10px; background-color: #f0f0f0;">
This is block B.
</div>
</t:block>
<t:block id="blockC">
<div style="margin: 2px; padding: 10px; background-color: #e8e8e8;">
This is block C.
</div>
</t:block>
</html>
```

## 第二十三讲：tapestry 的实体(entity)范本

源码如下：

Person.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.entities;

import java.io.Serializable;
import java.util.Date;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
```



```
import javax.persistence.Id;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import javax.persistence.Version;

@Entity
@SuppressWarnings("serial")
public class Person implements Serializable{

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(nullable = false)
    private Long id;

    @Version
    @Column(nullable = false)
    private Integer version;

    @Column(length = 10, nullable = false)
    private String firstName;

    @Column(length = 20, nullable = false)
    private String lastName;

    @Column(length = 30, nullable = false)
    private String region;

    @Temporal(TemporalType.DATE)
    private Date startDate;

    public Long getId() {
        return id;
    }

    public Integer getVersion() {
        return version;
    }

    public void setVersion(Integer version) {
        this.version = version;
    }
}
```





```
public String getFirstName() {
    return firstName;
}

public void setFirstName(String firstName) {
    this.firstName = firstName;
}

public String getLastName() {
    return lastName;
}

public void setLastName(String lastName) {
    this.lastName = lastName;
}

public String getRegion() {
    return region;
}

public void setRegion(String region) {
    this.region = region;
}

public Date getStartDate() {
    return startDate;
}

public void setStartDate(Date startDate) {
    this.startDate = startDate;
}
}
```

## 第二十四讲：tapestry 的 grid 组件

tapestry 的 grid 组件，具体实现方法看源码吧，之前的教程也有讲到，这次配合数据库操作使用，实体 `Person.java` 在第二十三讲中已经贴出来了，这里就不贴了，其他源码如下：

`PersonCreate.java`

```
/**
```



```
* 项目名称: TapestryStart
* 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
* 网址: http://www.flywind.org
* 版本: 1.0
* 编写: 飞风
* 时间: 2012-02-29
*/
```

```
package com.tapestry.app.pages.crud;

import java.util.Date;

import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.ioc.annotations.Inject;

import com.tapestry.app.entities.Person;
import com.tapestry.app.services.StartDAO;

public class PersonCreate {

    @Property
    private Person person;

    //导入服务接口
    @Inject
    private StartDAO dao;

    //初始化 user 实体
    void onPrepare(){
        person = new Person();
    }

    //提交表单的时候执行存储,返回当前页面
    Object onSuccess(){
        //如果时间为空值输入系统当前时间
        if(person.getStartDate() == null){
            person.setStartDate(new Date());
        }
        dao.create(person);
        return PersonList.class;
    }
}
```



#### PersonCreate.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
  xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
  xmlns:p="tapestry:parameter">
  <t:form t:id="personForm">
    <t:errors/>
    <p>版本(version): <t:textfield t:id="version" value="person.version"
t:validate="required"/></p>
    <p>姓(firstName): <t:textfield t:id="firstName" value="person.firstName"
t:validate="required"/></p>
    <p>名(lastName): <t:textfield t:id="lastName" value="person.lastName"
t:validate="required"/></p>
    <p>地区(region): <t:select t:id="region" value="person.region"
t:model="literal:深圳,北京" t:validate="required"/></p>
    <p><input type="submit" value=" 创 建 " /><!-- <t:pagelink
page="crud/UserList">返回查看页面</t:pagelink> --></p>
  </t:form>
</html>
```

#### PersonList.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages.crud;

import java.util.List;

import org.apache.tapestry5.annotations.PageActivationContext;
import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.ioc.annotations.Inject;

import com.tapestry.app.entities.Person;
import com.tapestry.app.services.StartDAO;

public class PersonList {

    //打开 user 读写
```



```
@Property
private Person person;

//打开 user 阵列的读写
@property
private List<Person> persons;

//导入操作数据库的服务
@Inject
private StartDAO dao;

//当前页面接收 user 的 id 值
@PageActivationContext
private Long id;

//页面加载时设置渲染
void setupRender(){
//查询 User 数据表
StringBuffer sql = new StringBuffer();
sql.append("from Person");
persons = dao.findWithQuery(sql.toString());
}

//单击 eventlink 执行删除操作
Object onDelete(Long id){
dao.deleteByID(Person.class, id);
return this;
}

}
```

#### PersonList.tml

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<style>
.table{border-collapse: collapse; }
.table td,table th{border:1px solid #999; padding:5px;}
</style>
<t:pagelink page="crud/PersonCreate">添加用户</t:pagelink><br/><br/>
<t:grid t:include="id,version,firstName,lastName,region,startDate"
t:source="persons" t:row="person" t:pagerPosition="top" t:rowsPerPage="1"
```



```
t:add="action" class="table">
  <p:actionCell>
    <t:pagelink page="crud/PersonUpdate" t:context="${person.id}"> 修 改
  </t:pagelink><t:eventlink t:event="delete" t:context="${person.id}">删除
</t:eventlink>
  </p:actionCell>
</t:grid>

</html>
```

#### PersonUpdate.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages.crud;

import org.apache.tapestry5.annotations.PageActivationContext;
import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.ioc.annotations.Inject;

import com.tapestry.app.entities.Person;
import com.tapestry.app.services.StartDAO;

public class PersonUpdate {

    //接收页面传来的 id 值
    @PageActivationContext
    private Long id;

    //设置 user 可读写
    @Property
    private Person person;

    //导入服务
    @Inject
    private StartDAO dao;

    //页面加载时运行
    void onPrepare(){
```



```
//user 为空数据时根据页面传递过来的 id 查询数据
if(person == null){
person = dao.findById(Person.class, id);
}
}
//提交表单保存 user 数据
Object onSuccess(){
dao.update(person);
return PersonList.class;
}
}
```

PersonUpdate.tml

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<t:form>
<t:errors/>
<p>版本(version): <t:textfield t:id="version" value="person.version"
t:validate="required"/></p>
<p>姓(firstName): <t:textfield t:id="firstName" value="person.firstName"
t:validate="required"/></p>
<p>名(lastName): <t:textfield t:id="lastName" value="person.lastName"
t:validate="required"/></p>
<p>地区(region): <t:select t:id="region" value="person.region"
t:model="literal:深圳,北京" t:validate="required"/></p>
<p><input type="submit" value="保存"/><t:pagelink
page="crud/PersonList">返回查看页面</t:pagelink></p>
</t:form>
</html>
```

## 第二十五讲：tapestry 的 loop 组件

tapestry 的 loop 组件，官方没有出分页组件，这里使用了 PagedLoop 分页组件，这个组件是 chenillekit 组件里的一个小组件，因不需要其它组件，这里就把 PagedLoop 组件单独提出来使用。实现从数据库中抓取每页所需的记录，而不是所有记录。下面先看下 PagedLoop 源码：

PagedLoop.java



```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.components.pagedLoop;

import org.apache.tapestry5.Block;
import org.apache.tapestry5.ClientElement;
import org.apache.tapestry5.ComponentResources;
import org.apache.tapestry5.ValueEncoder;
import org.apache.tapestry5.annotations.BeginRender;
import org.apache.tapestry5.annotations.Component;
import org.apache.tapestry5.annotations.Environmental;
import org.apache.tapestry5.annotations.Parameter;
import org.apache.tapestry5.annotations.Persist;
import org.apache.tapestry5.corelib.components.Delegate;
import org.apache.tapestry5.corelib.components.Looped;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.services.ComponentDefaultProvider;
import org.apache.tapestry5.services.javascript.JavaScriptSupport;
import org.apache.tapestry5.util.StringToEnumCoercion;

import com.tapestry.app.data.pagedloop.PagedSource;
import com.tapestry.app.data.pagedloop.PagerPosition;

public class PagedLoop implements ClientElement
{
    @Environmental
    private JavaScriptSupport javascriptSupport;

    @Parameter(value = "prop:componentResources.id", defaultPrefix =
"literal")
    private String clientId;

    /**
     * The element to render. If not null, then the loop will render the
     indicated element around its body (on each pass through the loop).

```



```
    * The default is derived from the component template.
    */
    @Parameter(value = "prop:componentResources.elementName",
defaultPrefix = "literal")
    private String element;

    /**
     * Defines the collection of values for the loop to iterate over.
     */
    @SuppressWarnings("unused")
    @Parameter(required = true)
    private Iterable<?> source;

    private PagedSource<?> pagedSource;

    /**
     * Defines where the pager (used to navigate within the "pages" of
results)
     * should be displayed: "top", "bottom", "both" or "none".
     */
    @Parameter(value = "bottom", defaultPrefix = "literal")
    private String pagerPosition;

    private PagerPosition internalPagerPosition;

    /**
     * The number of rows of data displayed on each page. If there are more
rows than will fit, the Grid will divide
     * up the rows into "pages" and (normally) provide a pager to allow
the user to navigate within the overall result set.
     */
    @Parameter("25")
    private int rowsPerPage;

    @Persist
    private int currentPage;

    /**
     * The current value, set before the component renders its body.
     */
    @SuppressWarnings("unused")
    @Parameter
    private Object value;
```





```
/**
 * If true and the Loop is enclosed by a Form, then the normal state
saving logic is turned off.
 * Defaults to false, enabling state saving logic within Forms.
 */
@SuppressWarnings("unused")
@Parameter(name = "volatile")
private boolean isVolatile;

/**
 * The index into the source items.
 */
@SuppressWarnings("unused")
@Parameter
private int index;

/**
 * Value encoder for the value, usually determined automatically from the
type of the property bound to the value
 * parameter.
 */
@Parameter(required = true)
private ValueEncoder encoder;

@SuppressWarnings("unused")
@Component(parameters = {"source=pagedSource",
    "element=prop:element", "value=inherit:value",
    "volatile=inherit:volatile", "encoder=inherit:encoder",
    "index=inherit:index"})
private Loop loop;

@Component(parameters = {"source=pagedSource",
    "rowsPerPage=rowsPerPage",
    "currentPage=currentPage"})
private Pager internalPager;

@SuppressWarnings("unused")
@Component(parameters = "to=pagerTop")
private Delegate pagerTop;

@SuppressWarnings("unused")
@Component(parameters = "to=pagerBottom")
```



```
private Delegate pagerBottom;

/**
 * A Block to render instead of the table (and pager, etc.) when the
source
 * is empty. The default is simply the text "There is no data to display".
 * This parameter is used to customize that message, possibly including
 * components to allow the user to create new objects.
 */
@Parameter(value = "block:empty")
private Block empty;

@Inject
private ComponentResources resources;

@Inject
private ComponentDefaultProvider defaultProvider;

private String assignedClientId;

public String getElement()
{
    return element;
}

public Object getPagerTop()
{
    return internalPagerPosition.isMatchTop() ? internalPager : null;
}

public Object getPagerBottom()
{
    return internalPagerPosition.isMatchBottom() ? internalPager :
null;
}

public PagedSource<?> getPagedSource()
{
    return pagedSource;
}

public int getRowsPerPage()
{

```



```
        return rowsPerPage;
    }

    public void setRowsPerPage(int rowsPerPage)
    {
        this.rowsPerPage = rowsPerPage;
    }

    public int getCurrentPage()
    {
        return currentPage;
    }

    public void setCurrentPage(int currentPage)
    {
        this.currentPage = currentPage;
    }

    ValueEncoder defaultEncoder()
    {
        return defaultProvider.defaultValueEncoder("value", resources);
    }

    @SuppressWarnings("unchecked")
    Object setupRender()
    {
        if (currentPage == 0)
            currentPage = 1;

        assignedClientId = javascriptSupport.allocateClientId(clientId);
        internalPagerPosition = new StringToEnumCoercion<PagerPosition>(
            PagerPosition.class).coerce(pagerPosition);

        pagedSource = new PagedSource(source);

        int availableRows = pagedSource.getTotalRowCount();

        // If there's no rows, display the empty block placeholder.
        if (availableRows == 0)
        {
            return empty;
        }
    }
}
```



```
int startIndex = (currentPage - 1) * rowsPerPage;
int endIndex = Math.min(startIndex + rowsPerPage - 1,
                        availableRows - 1);

pagedSource.prepare(startIndex, endIndex);

return null;
}

@BeginRender
Object begin()
{
    // Skip rendering of component (template, body, etc.) when there's
    // nothing to display.
    // The empty placeholder will already have rendered.
    return (pagedSource.getTotalRowCount() != 0);
}

void onAction(int newPage)
{
    // TODO: Validate newPage in range
    currentPage = newPage;
}

/**
 * Returns a unique id for the element. This value will be unique for
any given rendering of a
 * page. This value is intended for use as the id attribute of the
client-side element, and will
 * be used with any DHTML/Ajax related JavaScript.
 */
public String getClientId()
{
    return assignedClientId;
}
}
```

#### PagedLoop.tml

```
<!--
~ Apache License
~ Version 2.0, January 2004
~ http://www.apache.org/licenses/
~
```



```
~ Copyright 2008 by chenillekit.org
~
~ Licensed under the Apache License, Version 2.0 (the "License");
~ you may not use this file except in compliance with the License.
~ You may obtain a copy of the License at
~
~ http://www.apache.org/licenses/LICENSE-2.0
-->
```

```
<t:container
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd">

    <div t:id="pagerTop"/>

    <t:loop t:id="loop"><t:body/></t:loop>
<div class="clear"></div>
    <div t:id="pagerBottom"/>

    <t:block>
        <div t:id="internalPager"/>
    </t:block>

    <t:block id="empty">${message:empty.list}</t:block>

</t:container>
```

#### Pager.Java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.components.pagedLoop;

import org.apache.tapestry5.BindingConstants;
import org.apache.tapestry5.ComponentResources;
import org.apache.tapestry5.EventConstants;
import org.apache.tapestry5.Link;
import org.apache.tapestry5.MarkupWriter;
import org.apache.tapestry5.annotations.Import;
```



```
import org.apache.tapestry5.annotations.Parameter;
import org.apache.tapestry5.ioc.Messages;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.runtime.Component;

import com.tapestry.app.data.pagedloop.PagedSource;

@Import(stylesheet = {"Pager.css"})
public class Pager
{
    /**
     * The source of the data displayed by the PagedList (this is used to
     * determine
     * {@link PagedSource#getTotalRowCount()} how many rows are available},
     * which in turn determines the page count).
     */
    @Parameter
    private PagedSource<?> source;

    /**
     * The number of rows displayed per page.
     */
    @Parameter
    private int rowsPerPage;

    /**
     * The current page number (indexed from 1).
     */
    @Parameter
    private int currentPage;

    /**
     * If this pager is in a custom position it must provide the id of the
     * PagedLoop it is associated with.
     */
    @Parameter(defaultPrefix = BindingConstants.LITERAL, name = "for")
    private String forId;

    /**
     * Number of pages before and after the current page in the range. The pager
     * always displays links for 2 * range + 1 pages, unless that's more than
```



```
* the total number of available pages.
*/
@Parameter("5")
private int range;

private int lastIndex;

private int maxPages;

@Inject
private ComponentResources resources;

@Inject
private Messages messages;

private Component pagedLoopComponent;

private PagedLoop pagedLoop;

void setupRender()
{
    if (forId != null)
    {
        source = getPagedLoop().getPagedSource();
        rowsPerPage = getPagedLoop().getRowsPerPage();
        currentPage = getPagedLoop().getCurrentPage();
    }
}

void beginRender(MarkupWriter writer)
{
    int availableRows = source.getTotalRowCount();

    maxPages = ((availableRows - 1) / rowsPerPage) + 1;

    if (maxPages < 2)
        return;

    writer.element(resources.getContainer().getComponentResources().getElementName(), "class", "ck_paged_loop_pager");
    //writer.element("div", "class", "ck_paged_loop_pager");
```



```
lastIndex = 0;

for (int i = 1; i <= 2; i++)
writePageLink(writer, i);

int low = currentPage - range;
int high = currentPage + range;

if (low < 1)
{
low = 1;
high = 2 * range + 1;
}
else
{
if (high > maxPages)
{
high = maxPages;
low = high - 2 * range;
}
}

for (int i = low; i <= high; i++)
writePageLink(writer, i);

for (int i = maxPages - 1; i <= maxPages; i++)
writePageLink(writer, i);

writer.end();
//writer.end();
}

void onAction(int newPage)
{
// TODO: Validate newPage in range
currentPage = newPage;
if (forId != null)
{
getPagedLoopComponent().getComponentResources().triggerEvent(
EventConstants.ACTION, new Integer[]{newPage},
null);
}
}
```





```
private Component getPagedLoopComponent()
{
    if (forId != null && pagedLoopComponent == null)
        pagedLoopComponent = resources.getPage().getComponentResources().getEmbeddedComponent(forId);
    ;

    return pagedLoopComponent;
}

private PagedLoop getPagedLoop()
{
    if (forId != null && pagedLoop == null)
        pagedLoop = (PagedLoop) getPagedLoopComponent();

    return pagedLoop;
}

private void writePageLink(MarkupWriter writer, int pageIndex)
{
    if (pageIndex < 1 || pageIndex > maxPages)
        return;

    if (pageIndex <= lastIndex)
        return;

    if (pageIndex != lastIndex + 1)
        writer.write(" ... "); // &#8230; is ellipsis

    lastIndex = pageIndex;

    if (pageIndex == currentPage)
    {
        writer.element("span", "class", "ck_paged_loop_current");
        writer.write(Integer.toString(pageIndex));
        writer.end();
        return;
    }

    Link link = resources.createEventLink(EventConstants.ACTION, pageIndex);

    writer.element("a", "href", link, "title", messages.format("goto-page",
```



```
pageIndex));  
  
writer.write(Integer.toString(pageIndex));  
writer.end();  
}  
}
```

#### Pager.css

```
/*  
 * Apache License  
 * Version 2.0, January 2004  
 * http://www.apache.org/licenses/  
 *  
 * Copyright 2008 by chenillekit.org  
 *  
 * Licensed under the Apache License, Version 2.0 (the "License");  
 * you may not use this file except in compliance with the License.  
 * You may obtain a copy of the License at  
 *  
 * http://www.apache.org/licenses/LICENSE-2.0  
 */  
DIV.ck_paged_loop_pager{padding:10px 0;}  
DIV.ck_paged_loop_pager A {  
    border: 1px solid silver;  
    color: black;  
    font-size: medium;  
    margin-right: 5px;  
    padding: 2px 5px;  
    text-decoration: none;  
}  
DIV.ck_paged_loop_pager A:hover {  
    border: 1px solid #000;  
    color: black;  
    font-size: medium;  
    margin-right: 5px;  
    padding: 2px 5px;  
    text-decoration: none;  
}  
SPAN.ck_paged_loop_current {  
    border: 1px solid silver;  
    font-size: medium;  
    margin-right: 5px;  
    padding: 2px 5px;
```



```
text-decoration: none;
background-color: #809FFF;
color: white;
}
```

#### PagedSource.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.data.pagedloop;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

public class PagedSource<T> implements Iterable<T>
{
    private List<T> _source = new ArrayList<T>();

    private List<T> _pageSource = new ArrayList<T>();

    private Integer _iterableSize;

    public PagedSource(Iterable<T> source)
    {
        for (T aSource : source)
            _source.add(aSource);
    }

    /**
     * @return
     *
     * @see java.lang.Iterable#iterator()
     */
    public Iterator<T> iterator()
    {
        return _pageSource.iterator();
    }
}
```



```
}

public int getTotalRowCount()
{
    if (_iterableSize != null)
        return _iterableSize;

    _iterableSize = 0;

    Iterator<?> it = _source.iterator();
    while (it.hasNext())
    {
        it.next();
        _iterableSize++;
    }

    return _iterableSize;
}

public void prepare(int startIndex, int endIndex)
{
    for (int i = startIndex; i <= endIndex; i++)
    {
        _pageSource.add(_source.get(i));
    }
}
}
```

#### PagerPosition.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.data.pagedloop;

public enum PagerPosition
{
    /**
```



```
* Position the pager above the paged content.
*/
TOP(true, false),

/**
 * Position the pager below the paged content (this is the default).
 */
BOTTOM(false, true),

/**
 * Show the pager above and below the paged content.
 */
BOTH(true, true),

/**
 * Don't show a pager (the application will need to supply its own
 * navigation mechanism).
 */
NONE(false, false);

private final boolean _matchTop;

private final boolean _matchBottom;

private PagerPosition(boolean matchTop, boolean matchBottom)
{
    _matchTop = matchTop;
    _matchBottom = matchBottom;
}

public boolean isMatchBottom()
{
    return _matchBottom;
}

public boolean isMatchTop()
{
    return _matchTop;
}
}
```

下面是使用 PagedLoop 组件的代码:

PersonList2.java



```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages.crud;

import java.util.List;

import org.apache.tapestry5.annotations.Component;
import org.apache.tapestry5.annotations.PageActivationContext;
import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.ioc.annotations.Inject;

import com.tapestry.app.components.pagedLoop.PagedLoop;
import com.tapestry.app.entities.Person;
import com.tapestry.app.services.StartDAO;

public class PersonList2 {

    //打开 user 读写
    @Property
    private Person person;

    //打开 user 阵列的读写
    @Property
    private List<Person> persons;

    //导入操作数据库的服务
    @Inject
    private StartDAO dao;

    //当前页面接收 user 的 id 值
    @PageActivationContext
    private Long id;

    //开启 PagedLoop
    @Component(parameters={"source=persons", "rowsPerPage=1", "pagerPosition=bottom"}, "value=person",
    private PagedLoop personLoop;
```



```
//页面加载时设置渲染
void setupRender(){
//查询 User 数据表
StringBuffer sql = new StringBuffer();
sql.append("from Person");
persons = dao.findWithQuery(sql.toString());
}

//单击 eventlink 执行删除操作
Object onDelete(Long id){
dao.deleteByID(Person.class, id);
return this;
}

}
```

#### PersonList2.tml

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<style>
.table{border-collapse: collapse; }
.table td,table th{border:1px solid #999; padding:5px;}
.meul{list-style:none;}
.meul li{ width:80px; float:left; height:20px;}
.clear{clear:both; height:0; display:block; overflow:hidden;}
</style>
<t:pagelink page="crud/PersonCreate">添加用户</t:pagelink><br/><br/>
<ul class="meul">
<li>id</li>
<li>version</li>
<li>firstName</li>
<li>lastName</li>
<li>region</li>
<li>startDate</li>
<li>操作</li>
<div class="clear"></div>
<div t:id="personLoop">
<li>${person.id}</li>
<li>${person.version}</li>
<li>${person.firstName}</li>
```



```
<li>${person.lastName}</li>
<li>${person.region}</li>
<li>${person.startDate}</li>
<li><t:pagelink page="crud/PersonUpdate" t:context="${person.id}">修
改</t:pagelink><t:eventlink t:event="delete" t:context="${person.id}">删
除</t:eventlink></li>
</div>
</ul>
</html>
```

## 第二十六讲：tapestry 的树状(tree)组件

tapestry 提供了树状组件 tree，源码如下：

TreeBrowse.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.ValueEncoder;
import org.apache.tapestry5.annotations.InjectComponent;
import org.apache.tapestry5.corelib.components.Tree;
import org.apache.tapestry5.tree.DefaultTreeModel;
import org.apache.tapestry5.tree.TreeModel;

import com.tapestry.app.entities.Stuff;
import com.tapestry.app.services.StuffTreeModelAdapter;

public class TreeBrowse {
    // Screen fields

    private TreeModel<Stuff> stuffModel;

    // Generally useful bits and pieces
```





```
@InjectComponent
private Tree tree;

// The code

void onClearExpansions() {
    tree.clearExpansions();
}

// Getters and setters

public TreeModel<Stuff> getStuffModel() {

    if (stuffModel == null) {
        ValueEncoder<Stuff> stuffEncoder = new ValueEncoder<Stuff>() {

            @Override
            public String toClient(Stuff stuff) {
                return stuff.uuid;
            }

            @Override
            public Stuff toValue(String uuid) {
                return Stuff.ROOT.searchSubTree(uuid);
            }

        };
        stuffModel = new DefaultTreeModel<Stuff>(stuffEncoder, new
        StuffTreeModelAdapter(), Stuff.ROOT.children);
    }
    return stuffModel;
}
}
```

#### TreeBrowse.tml

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<div style="margin: 20px;">
<a t:type="eventlink" t:event="clearExpansions">Clear
Expansions</a><br/><br/>
<t:tree t:id="tree" model="stuffModel"/><br/>
```



```
<a t:type="pagelink" t:page="TreeBrowse" href="#">Refresh</a>
</div>
</html>
```

#### Stuff.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.entities;

import java.util.List;
import java.util.UUID;

import org.apache.tapestry5.ioc.internal.util.CollectionFactory;

public class Stuff {
    public static final Stuff ROOT = new Stuff("<root>");

    static {
        ROOT.addChild(new Stuff("Pets").addChildNamed("Oscar", "Gromit", "Max",
            "Roger", "Cooper"));

        ROOT.addChild(new Stuff("Games").addChild(
            new Stuff("Board Games").addChildNamed("Settlers of Catan", "Agricola",
            "Ra", "Risk", "Dvonn"))
            .addChild(new Stuff("Card Games").addChildNamed("Magic the Gathering",
            "Dominion", "Mu"))));

        Stuff numbers = new Stuff("Numbers");
        for (int i = 0; i < 100; i++) {
            numbers.addChild(new Stuff(Integer.toString(i)));
        }
        ROOT.addChild(numbers);
    }

    public final String uuid = UUID.randomUUID().toString();
    public final String name;
    public List<Stuff> children;
```



```
public Stuff(String name) {
    this.name = name;
}

public Stuff addChild(Stuff child) {
    if (children == null) {
        children = CollectionFactory.newList();
    }

    children.add(child);

    return this;
}

public Stuff addChildrenNamed(String... names) {
    for (String name : names) {
        addChild(new Stuff(name));
    }

    return this;
}

public Stuff searchSubTree(String uuid) {

    if (this.uuid.equals(uuid)) {
        return this;
    }

    for (Stuff child : children) {
        Stuff match = child.searchSubTree(uuid);

        if (match != null) {
            return match;
        }
    }

    return null;
}
```



#### StuffTreeModelAdapter.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.services;

import java.util.List;

import org.apache.tapestry5.tree.TreeModelAdapter;

import com.tapestry.app.entities.Stuff;

public class StuffTreeModelAdapter implements TreeModelAdapter<Stuff> {

    @Override
    public boolean isLeaf(Stuff stuff) {
        return !hasChildren(stuff);
    }

    @Override
    public boolean hasChildren(Stuff stuff) {
        return stuff.children != null && !stuff.children.isEmpty();
    }

    @Override
    public List<Stuff> getChildren(Stuff stuff) {
        return stuff.children;
    }

    @Override
    public String getLabel(Stuff stuff) {
        return stuff.name;
    }

}
```



## 第二十七讲：tapestry 与 javaScript

tapestry 的 js 库基于 prototype，写法非常简单。这个实例 JS 实现了当输入单字符时显示绿色、双字符时显示红色。代码如下：

JavaScript.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.Import;
import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.services.javascript.JavaScriptSupport;

@Import(library="context:assets/js/JavaScript.js")
public class JavaScript {

    @Property
    @SuppressWarnings("unused")
    private String firstName;

    @Inject
    private JavaScriptSupport javaScriptSupport;

    public void setupRender() {

        javaScriptSupport.addScript("new ColorSwitcher();");
    }
}
```

JavaScript.tml

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
```



```
<form t:type="form">
First name: <input t:type="TextField" t:id="firstName"/><br/>
</form>
</html>
```

JavaScript.js

```
ColorSwitcher = Class.create( {

  initialize : function() {
    this.element = $('firstName');
    Event.observe(this.element, 'keyup',
    this.doSwitchColor.bindAsEventListener(this));
    this.element.setStyle({color: 'red'});
  },

  doSwitchColor : function(e) {
    if (this.element.getStyle('color') == 'red') {
      this.element.setStyle({color: 'green'});
    }
    else {
      this.element.setStyle({color: 'red'});
    }
  }

} )
```

## 第二十八讲：tapestry 与可重复使用的 javaScript

tapestry 与可重复使用的 javaScript，实现方法是给函数传变量，代码如下：

ReusableJavaScript.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
```



```
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.Import;
import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.services.javascript.JavaScriptSupport;

@Import(library="context:assets/js/textbox_hint.js")
public class ReusableJavaScript {
    @Property
    @SuppressWarnings("unused")
    private String firstName;

    @Property
    @SuppressWarnings("unused")
    private String lastName;

    @Inject
    private JavaScriptSupport javaScriptSupport;

    public void setupRender() {

        javaScriptSupport.addScript(String.format("new TextboxHint('%s', '%s', '%s');", "firstName",
            "Enter First Name", "#ff6600"));
        javaScriptSupport.addScript(String.format("new TextboxHint('%s', '%s', '%s');", "lastName", "Enter Last Name",
            "blue"));
    }
}
```

#### ReusableJavaScript.tml

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
    <form t:type="form" t:autofocus="false">
        <input t:type="TextField" t:id="firstName"/>
        <input t:type="TextField" t:id="lastName"/><br/>
    </form>
</html>
```

textbox\_hint.js



```
TextboxHint = Class.create( {

  initialize : function(textboxId, hintText, hintColor) {
    this.textbox = $(textboxId);
    this.hintText = hintText;
    this.hintColor = hintColor;
    this.normalColor = this.textbox.getStyle('color');

    Event.observe(this.textbox, 'focus',
      this.doClearHint.bindAsEventListener(this));
    Event.observe(this.textbox, 'blur',
      this.doCheckHint.bindAsEventListener(this));
    Event.observe(this.textbox, 'change',
      this.doCheckHint.bindAsEventListener(this));
    Event.observe(this.textbox.form, 'submit',
      this.doClearHint.bindAsEventListener(this));

    this.doCheckHint();
  },

  doClearHint : function(e) {
    if (this.textbox.value == this.hintText) {
      this.textbox.value = "";
    }
    this.textbox.setStyle({color: this.normalColor});
  },

  doCheckHint : function(e) {

    // If field is empty, put the hintText in it and set its color to hintColor

    if (this.textbox.value.length == 0) {
      this.textbox.value = this.hintText;
      this.textbox.setStyle({color: this.hintColor});
    }

    // Else if field contains hintText, set its color to hintColor

    else if (this.textbox.value == this.hintText) {
      this.textbox.setStyle({color: this.hintColor});
    }

    // Else, set the field's color to its normal color
```





```
else {  
this.textbox.setStyle({color: this.normalColor});  
}  
  
}  
  
} )
```

## 第二十九讲：tapestry 更好的可重复使用 JavaScript

源码如下：

RobustJavaScript.java

```
/**  
 * 项目名称: TapestryStart  
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql  
 * 网址: http://www.flywind.org  
 * 版本: 1.0  
 * 编写: 飞风  
 * 时间: 2012-02-29  
 */  
package com.tapestry.app.pages;  
  
import org.apache.tapestry5.annotations.Import;  
import org.apache.tapestry5.annotations.Property;  
import org.apache.tapestry5.ioc.annotations.Inject;  
import org.apache.tapestry5.json.JSONObject;  
import org.apache.tapestry5.services.javascript.JavaScriptSupport;  
  
@Import(library="context:assets/js/robust_textbox_hint.js")  
public class RobustJavaScript {  
    @Property  
    @SuppressWarnings("unused")  
    private String firstName;  
  
    @Property  
    @SuppressWarnings("unused")
```



```
private String lastName;

@property
private String firstNameId;

@property
private String lastNameId;

@Inject
private JavaScriptSupport javaScriptSupport;

public void setupRender() {
    firstNameId = "firstName";
    lastNameId = "lastName";

    JSONObject spec = new JSONObject();
    spec.put("textboxId", firstNameId);
    spec.put("hintText", "Enter First Name");
    spec.put("hintColor", "#ff6600");
    javaScriptSupport.addInitializerCall("textboxHint", spec);

    JSONObject spec2 = new JSONObject();
    spec2.put("textboxId", lastNameId);
    spec2.put("hintText", "Enter Last Name");
    spec2.put("hintColor", "#808080");
    javaScriptSupport.addInitializerCall("textboxHint", spec2);
}
}
```

#### RobustJavaScript.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<form t:type="form" t:autofocus="false">
<input t:type="TextField" t:id="firstName"/>
<input t:type="TextField" t:id="lastName"/><br/>
</form>
</html>
```

#### robust\_textbox\_hint.js

```
TextboxHint = Class.create( {
```



```
initialize : function(textboxId, hintText, hintColor) {
this.textbox = $(textboxId);
this.hintText = hintText;
this.hintColor = hintColor;
this.normalColor = this.textbox.getStyle('color');

Event.observe(this.textbox, 'focus',
this.doClearHint.bindAsEventListener(this));
Event.observe(this.textbox, 'blur',
this.doCheckHint.bindAsEventListener(this));
Event.observe(this.textbox, 'change',
this.doCheckHint.bindAsEventListener(this));
Event.observe(this.textbox.form, 'submit',
this.doClearHint.bindAsEventListener(this));

this.doCheckHint();
},

doClearHint : function(e) {
if (this.textbox.value == this.hintText) {
this.textbox.value = "";
}
this.textbox.setStyle({color: this.normalColor});
},

doCheckHint : function(e) {

// If field is empty, put the hintText in it and set its color to hintColor

if (this.textbox.value.length == 0) {
this.textbox.value = this.hintText;
this.textbox.setStyle({color: this.hintColor});
}

// Else if field contains hintText, set its color to hintColor

else if (this.textbox.value == this.hintText) {
this.textbox.setStyle({color: this.hintColor});
}

// Else, set the field's color to its normal color

else {
```



```
this.textbox.setStyle({color: this.normalColor});
}

}

} )

Tapestry.Initializer.textboxHint = function(spec) {
    new TextboxHint(spec.textboxId, spec.hintText, spec.hintColor);
}
```

## 第三十讲：tapestry 强大的 javaScript 混合 (mixins)

此例还是使用上一篇的实例 JS，看混合的方法，实在是太强大了!!

JavaScriptMixin.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.Property;

public class JavaScriptMixin {
    @Property
    @SuppressWarnings("unused")
    private String firstName;

    @Property
    @SuppressWarnings("unused")
    private String lastName;
}
```



#### JavaScriptMixin.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<form t:type="form" t:autofocus="false">
<input  t:type="TextField"  t:id="firstName"  t:mixins="TextboxHint"
t:hintText="Enter First Name" t:hintColor="#ff6600"/>
<input  t:type="TextField"  t:id="lastName"  t:mixins="TextboxHint"
t:hintText="Enter Last Name" t:hintColor="#808080"/><br/>
</form>
</html>
```

#### TextboxHint.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.mixins;

import org.apache.tapestry5.BindingConstants;
import org.apache.tapestry5.ClientElement;
import org.apache.tapestry5.annotations.Import;
import org.apache.tapestry5.annotations.InjectContainer;
import org.apache.tapestry5.annotations.Parameter;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.json.JSONObject;
import org.apache.tapestry5.services.javascript.JavaScriptSupport;

@Import(library="context:assets/js/robust_textbox_hint.js")
public class TextboxHint {
    @Parameter(required = true, defaultPrefix = BindingConstants.LITERAL)
    private String hintText;

    @Parameter(required = true, defaultPrefix = BindingConstants.LITERAL)
    private String hintColor;

    @Inject
```



```
private JavaScriptSupport javaScriptSupport;

@InjectContainer
private ClientElement clientElement;

public void afterRender() {

    JSONObject spec = new JSONObject();
    spec.put("textboxId", clientElement.getClientId());
    spec.put("hintText", hintText);
    spec.put("hintColor", hintColor);
    javaScriptSupport.addInitializerCall("textboxHint", spec);
}
}
```

## 第三十一讲：tapestry 的 javaScript 混合 (mixins)统计点击数

源码如下：

CreatingMixins1.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.annotations.SessionState;

import com.tapestry.app.entities.MyOrder;

public class CreatingMixins1 {
    @SessionState
```



```
@Property
private MyOrder myOrder;

void setupRender() {
myOrder.setApplesQuantity(0);
myOrder.setOrangesQuantity(0);
myOrder.setBananasQuantity(0);
}

Object onSuccessFromPlainForm() {
orderOneApple();
return CreatingMixins2.class;
}

Object onOrderOneOrange() {
orderOneOrange();
return CreatingMixins2.class;
}

Object onActionFromOrderOneBanana() {
orderOneBanana();
return CreatingMixins2.class;
}

Object onSuccessFromMixinForm() {
orderOneApple();
return CreatingMixins2.class;
}

Object onActionFromOrderOneBananaWithMixin() {
orderOneBanana();
return CreatingMixins2.class;
}

void orderOneApple() {
sleep(1500); // Sleep 1.5 seconds to simulate busy system
myOrder.setApplesQuantity(myOrder.getApplesQuantity() + 1);
}

void orderOneOrange() {
sleep(1500); // Sleep 1.5 seconds to simulate busy system
myOrder.setOrangesQuantity(myOrder.getOrangesQuantity() + 1);
}
```



```
void orderOneBanana() {
    sleep(1500); // Sleep 1.5 seconds to simulate busy system
    myOrder.setBananasQuantity(myOrder.getBananasQuantity() + 1);
}

private void sleep(long duration) {
    try {
        Thread.sleep(duration);
    }
    catch (InterruptedException e) {
    }
}
}
```

#### CreatingMixins1.tml

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<form t:type="form" t:id="plainForm">
<div style="padding: 20px 50px;">
<input t:type="submit" value="Order 1 Apple"/>
<a t:type="eventlink" t:event="orderOneOrange" href="#">Order 1 Orange</a>
<a t:type="actionlink" t:id="orderOneBanana" href="#">Order 1 Banana</a>
</div>
</form>
<form t:type="form" t:id="mixinForm" t:clientValidation="blur">
<div style="padding: 20px 50px;">
<input t:type="submit" value="Order 1 Apple" t:mixins="clickonce"/>
<a t:type="eventlink" t:event="orderOneOrange" t:mixins="clickonce"
href="#">Order 1 Orange</a>
<a t:type="actionlink" t:id="orderOneBananaWithMixin"
t:mixins="clickonce" href="#">Order 1 Banana</a>
</div>
</form>
</html>
```

#### ClickOnce.js

```
var alreadyClickedOnce = false;

ClickOnce = Class.create( {
```





```
initialize: function(elementId) {
    Event.observe($(elementId), 'click',
    this.doClickOnce.bindAsEventListener(this));
},

doClickOnce: function(e) {
    if (alreadyClickedOnce) {
        e.stop();
    }
    alreadyClickedOnce = true;
}

} )

Tapestry.Initializer.clickOnce = function(spec) {
    new ClickOnce(spec.elementId);
}
```

#### ClickOnce.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.mixins;

import org.apache.tapestry5.ClientElement;
import org.apache.tapestry5.annotations.Import;
import org.apache.tapestry5.annotations.InjectContainer;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.json.JSONObject;
import org.apache.tapestry5.services.javascript.JavaScriptSupport;

@Import(library="context:assets/js/ClickOnce.js")
public class ClickOnce {
    @Inject
    private JavaScriptSupport javaScriptSupport;

    @InjectContainer
    private ClientElement clientElement;
```



```
public void afterRender() {  
  
    JSONObject spec = new JSONObject();  
    spec.put("elementId", clientElement.getClientId());  
    javascriptSupport.addInitializerCall("clickOnce", spec);  
}  
  
}
```

#### CreatingMixins2.java

```
/**  
 * 项目名称: TapestryStart  
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql  
 * 网址: http://www.flywind.org  
 * 版本: 1.0  
 * 编写: 飞风  
 * 时间: 2012-02-29  
 */  
package com.tapestry.app.pages;  
  
import org.apache.tapestry5.annotations.Property;  
import org.apache.tapestry5.annotations.SessionState;  
  
import com.tapestry.app.entities.MyOrder;  
  
public class CreatingMixins2 {  
    @SessionState  
    @Property  
    @SuppressWarnings("unused")  
    private MyOrder myOrder;  
}
```

#### CreatingMixins2.tml

```
<html t:type="layout" title="tapestryStart Index"  
t:sidebarTitle="Framework Version"  
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"  
xmlns:p="tapestry:parameter">  
    <div style="padding: 20px 50px;">  
        You have ordered ${myOrder.applesQuantity} apples,  
        ${myOrder.orangesQuantity} oranges, ${myOrder.bananasQuantity} bananas.  
    </div>  
</html>
```



#### MyOrder.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.entities;

public class MyOrder {
    private int applesQuantity;
    private int orangesQuantity;
    private int bananasQuantity;

    public int getApplesQuantity() {
        return applesQuantity;
    }

    public void setApplesQuantity(int applesQuantity) {
        this.applesQuantity = applesQuantity;
    }

    public int getOrangesQuantity() {
        return orangesQuantity;
    }

    public void setOrangesQuantity(int orangesQuantity) {
        this.orangesQuantity = orangesQuantity;
    }

    public int getBananasQuantity() {
        return bananasQuantity;
    }

    public void setBananasQuantity(int bananasQuantity) {
        this.bananasQuantity = bananasQuantity;
    }
}
```



## 第三十二讲：tapestry javaScript 动画

动画效果使用了 [script.aculo.us](http://script.aculo.us) 的 JS 库，源码如下：

Scriptaculous.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.services.javascript.InitializationPriority;
import org.apache.tapestry5.services.javascript.JavaScriptSupport;

public class Scriptaculous {
    @Property
    @SuppressWarnings("unused")
    private String firstName;

    @Property
    @SuppressWarnings("unused")
    private String lastName;

    @Inject
    private JavaScriptSupport javascriptSupport;

    public void setupRender() {

        javascriptSupport.addScript(InitializationPriority.LATE,
            String.format("Effect.Grow('%s');", "expando"));
    }
}
```

Scriptaculous.tml

```
<html t:type="layout" title="tapestryStart Index"
```



```
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<div id="expando" style="padding: 10px; background-color: #eeeeee;">

<form t:type="form" t:autofocus="false">
<input t:type="TextField" t:id="firstName" t:mixins="TextboxHint"
t:hintText="Enter First Name" t:hintColor="#808080"/>
<input t:type="TextField" t:id="lastName" t:mixins="TextboxHint"
t:hintText="Enter Last Name" t:hintColor="#808080"/><br/>
</form>
</div>
</html>
```

## 第三十三讲: tapestry Ajax eventlink 无刷新页面

tapestry 的 Ajax 使用非常简单, Ajax 的页面无刷新, 刚好解决了 eventlink、actionlink 等组件无刷新, 这点在 js 应用中非常的强大。下面就以 eventlink 组件为例。源码如下:

AJAXEventLink.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import java.util.Date;

import org.apache.tapestry5.annotations.InjectComponent;
import org.apache.tapestry5.corelib.components.Zone;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.services.Request;

public class AJAXEventLink {
```



```
@Inject
private Request request;

@InjectComponent
private Zone time2Zone;

@SuppressWarnings("unused")
private Date serverTime1;

@SuppressWarnings("unused")
private Date serverTime2;

void onRefreshPage() {

}

Object onRefreshZone() {
return request.isXHR() ? time2Zone.getBody() : null;
}

public Date getServerTime1() {
return new Date();
}

public Date getServerTime2() {
return new Date();
}

}
```

#### AJAXEventLink.tml

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<div style="margin-left: 50px">
<a t:type="eventlink" t:event="refreshPage" href="#">页面需要刷新</a><br/>
<a t:type="eventlink" t:event="refreshZone" href="#" t:zone="time2Zone">
页面无刷新</a><br/><br/>

serverTime1: ${serverTime1}
<t:zone t:id="time2Zone">
serverTime2: ${serverTime2}
```



```
</t:zone><br/>
</div>
</html>
```

## 第三十四讲：tapestry Ajax actiontlink 无刷新页面

源码如下：

AJAXActionLink.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import java.util.Date;

import org.apache.tapestry5.annotations.InjectComponent;
import org.apache.tapestry5.corelib.components.Zone;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.services.Request;

public class AJAXActionLink {

    @Inject
    private Request request;

    @InjectComponent
    private Zone time2Zone;

    void onActionFromRefreshPage(){}

    Object onActionFromRefreshZone(){
        return request.isXHR()? time2Zone.getBody():null;
    }
}
```



```
}

public Date getServerTime1(){
return new Date();
}

public Date getServerTime2(){
return new Date();
}
}
```

AJAXActionLink.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<div style="margin-left: 50px">
<a t:type="actionlink" t:id="refreshPage" href="#">ActionLink 刷新页面
</a><br/>
<a          t:type="actionlink"          t:id="refreshZone"          href="#"
t:zone="time2Zone">ActionLink AJAX 无刷新</a><br/><br/>

serverTime1: ${serverTime1}
<t:zone t:id="time2Zone">
serverTime2: ${serverTime2}
</t:zone><br/>
</div>
</html>
```

## 第三十五讲: tapestry Ajax zone 组件无黄色闪烁的背景

实现方法很简单, 在 zone 组件里加上 `t:update="show"` 就不会出现黄色背景了。源码如下:

ZoneWithoutYellowHighlight.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
```





```
* 版本: 1.0
* 编写: 飞风
* 时间: 2012-02-29
*/
package com.tapestry.app.pages;

import java.util.Date;

import org.apache.tapestry5.annotations.InjectComponent;
import org.apache.tapestry5.corelib.components.Zone;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.services.Request;

public class ZoneWithoutYellowHighlight {

    @Inject
    private Request request;

    @InjectComponent
    private Zone time2Zone;

    void onActionFromRefreshPage(){}

    Object onActionFromRefreshZone(){
        return request.isXHR()?time2Zone.getBody():null;
    }

    public Date getServerTime1(){
        return new Date();
    }

    public Date getServerTime2(){
        return new Date();
    }
}

ZoneWithoutYellowHighlight.tml
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<div style="margin-left: 50px">
<a t:type="actionlink" t:id="refreshPage" href="#">ActionLink 刷新页面
```



```
</a><br/>
<a      t:type="actionlink"      t:id="refreshZone"      href="#"
t:zone="time2Zone">ActionLink AJAX 无刷新</a><br/><br/>

serverTime1: ${serverTime1}
<t:zone t:id="time2Zone" t:update="show">
serverTime2: ${serverTime2}
</t:zone><br/>
</div>
</html>
```

## 第三十六讲：tapestry Ajax 表单(form)

源码如下：

AJAXForm.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;

import org.apache.tapestry5.annotations.Component;
import org.apache.tapestry5.annotations.InjectComponent;
import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.corelib.components.DateField;
import org.apache.tapestry5.corelib.components.Form;
import org.apache.tapestry5.corelib.components.TextField;
import org.apache.tapestry5.corelib.components.Zone;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.services.Request;

public class AJAXForm {
```



```
@Property
private String firstName;

@Property
private String lastName;

@Property
private Date birthday;

@Inject
private Request request;

@InjectComponent
private Zone formZone;

@Component(id="firstName")
private TextField firstNameField;

@Component(id="lastName")
private TextField lastNameField;

@Component(id = "birthday")
private DateField birthdayField;

@Component(id="ajaxForm")
private Form form;

void setupRender(){
if(firstName == null && lastName == null && birthday == null){
firstName = "飞";
lastName = "风";
birthday = new Date(0);
}
}

void onValidateFromAjaxForm(){
if(firstName == null || firstName.trim().equals("")){
form.recordError(firstNameField, "姓不能为空值! ");
}
if(lastName == null || lastName.trim().equals("")){
form.recordError(lastNameField, "名不能为空值! ");
}
}
```



```
if(birthday == null){
form.recordError(birthdayField, "生日不能为空值!");
Date now = new Date();
if(birthday.after(now)){
form.recordError(birthdayField, "生日必须是过去的时间!");
}
}
}

Object onSuccess(){
return request.isXHR()?formZone.getBody():null;
}

Object onFailue(){
return request.isXHR()?formZone.getBody():null;
}

public String getName(){
return firstName + " " + lastName;
}

public DateFormat getBirthdayFormat(){
return new SimpleDateFormat("yyyy-MM-dd");
}
}
```

#### AJAXForm.tml

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<t:zone t:id="formZone" id="formZone">
<form t:id="ajaxForm" t:type="form" t:zone="formZone" style="border: 2px
solid #eee; padding: 15px">

姓: <input t:type="TextField" t:id="firstName"/><br/>
名: <input t:type="TextField" t:id="lastName"/><br/>
生日: <input t:type="datefield" t:id="birthday"/><br/><br/>

<input type="submit" value="提交"/><br/><br/>

<t:errors/>
```



```
欢迎  ${name}.  你的生日是  <t:output  t:format="birthdayFormat"
value="birthday"/>

</form>
</t:zone>
</html>
```

## 第三十七讲：tapestry Ajax 表单(form)多zone 组件

源码如下：

AJAXFormMultipleZoneUpdate.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;

import org.apache.tapestry5.annotations.Component;
import org.apache.tapestry5.annotations.InjectComponent;
import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.corelib.components.DateField;
import org.apache.tapestry5.corelib.components.Form;
import org.apache.tapestry5.corelib.components.TextField;
import org.apache.tapestry5.corelib.components.Zone;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.services.Request;
import org.apache.tapestry5.services.ajax.AjaxResponseRenderer;

public class AJAXFormMultipleZoneUpdate {
```



```
@Property
private String firstName;

@property
private String lastName;

@property
private Date birthday;

@Inject
private Request request;

@InjectComponent
private Zone formZone;

@InjectComponent
private Zone outZone;

@Component(id="firstName")
private TextField firstNameField;

@Component(id="lastName")
private TextField lastNameField;

@Component(id = "birthday")
private DateField birthdayField;

@Component(id="ajaxForm")
private Form form;

@Inject
private AjaxResponseRenderer ajaxResponseRenderer;

void setupRender(){
if(firstName == null && lastName == null && birthday == null){
firstName = "飞";
lastName = "风";
birthday = new Date(0);
}
}

void onValidateFromAjaxForm(){
```



```
if(firstName == null || firstName.trim().equals("")){
    form.recordError(firstNameField, "姓不能为空值!");
}
if.lastName == null || lastName.trim().equals("")){
    form.recordError(lastNameField, "名不能为空值!");
}
if(birthday == null){
    form.recordError(birthdayField, "生日不能为空值!");
    Date now = new Date();
    if(birthday.after(now)){
        form.recordError(birthdayField, "生日必须是过去的时间!");
    }
}
}

void onSuccess(){
    if(request.isXHR()){
        ajaxResponseRenderer.addRender(formZone).addRender(outZone);
    }
}

void onFailue(){
    if(request.isXHR()){
        ajaxResponseRenderer.addRender(formZone);
    }
}

public String getName(){
    return firstName + " " + lastName;
}

public DateFormat getBirthdayFormat(){
    return new SimpleDateFormat("yyyy-MM-dd");
}
}
```

#### AJAXFormMultipleZoneUpdate.tml

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<t:zone t:id="formZone" id="formZone">
<form t:id="ajaxForm" t:type="form" t:zone="formZone" style="border: 2px
```



```
solid #eee; padding: 15px">

姓: <input t:type="TextField" t:id="firstName"/><br/>
名: <input t:type="TextField" t:id="lastName"/><br/>
生日: <input t:type="datefield" t:id="birthday"/><br/><br/>

<input type="submit" value="提交"/><br/><br/>

<t:errors/>

</form>
</t:zone>
<t:zone t:id="outZone" id="outZone" style="border: 2px solid #eee; padding:
15px">
欢迎  ${name}.  你的生日是  <t:output t:format="birthdayFormat"
value="birthday"/>
</t:zone>
</html>
```

## 第三十八讲: **tapestry Ajax 关联下拉选框 (select)组件**

源码如下:

AJAXSelect1.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.ComponentResources;
import org.apache.tapestry5.annotations.InjectComponent;
import org.apache.tapestry5.annotations.InjectPage;
import org.apache.tapestry5.annotations.Property;
```





```
import org.apache.tapestry5.corelib.components.Zone;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.services.Request;

public class AJAXSelect1 {
    static final private String[] ALL_MAKES = new String[] { "梅州市", "河源市" };
    static final private String[] NO_MODELS = new String[] {};
    static final private String[] MEIZHOU_MODELS = new String[] { "五华县", "兴宁县" };
    static final private String[] HEYUAN_MODELS = new String[] { "紫金县", "龙川县" };

    @Property
    private String areaMake;

    @Property
    private String cithModel;

    @Property
    private String keywords;

    @Property
    private String[] areaMakes;

    @Property
    @SuppressWarnings("unused")
    private String[] cithModels;

    @InjectPage
    private AJAXSelect2 page2;

    @InjectComponent
    private Zone cithModelZone;

    @Inject
    private ComponentResources componentResources;

    @Inject
    private Request request;

    void setupRender() {
```



```
if (areaMakes == null) {
    areaMakes = ALL_MAKES;
    cithModels = NO_MODELS;
}
}

Object onValueChangedFromAreaMake(String areaMake) {
    this.areaMake = areaMake;
    areaMakes = ALL_MAKES;

    if (areaMake == null) {
        cithModels = NO_MODELS;
    }
    else if (areaMake.equals(areaMakes[0])) {
        cithModels = MEIZHOU_MODELS;
    }
    else if (areaMake.equals(areaMakes[1])) {
        cithModels = HEYUAN_MODELS;
    }
    else {
        cithModels = NO_MODELS;
    }
    cithModel = null;

    return request.isXHR() ? cithModelZone.getBody() : null;
}

Object onSuccess() {
    page2.set(areaMake, cithModel, keywords);
    componentResources.discardPersistentFieldChanges();
    return page2;
}

Object onGoHome() {
    componentResources.discardPersistentFieldChanges();
    return Index.class;
}
}
```

#### AJAXSelect1.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
```



```
xmlns:p="tapestry:parameter">
<form t:type="form" t:id="searchCriteria">

<t:label for="areaMake"/>:
<select      t:type="select"      t:id="areaMake"      t:model="areaMakes"
t:zone="cithModelZone"
t:blankOption="ALWAYS" t:blankLabel="请选择" t:validate="required"/>

<t:zone t:id="cithModelZone" id="cithModelZone" style="display: inline;">
<t:label for="cithModel"/>:
<select t:type="select" t:id="cithModel" t:model="cithModels"
t:blankOption="ALWAYS" t:blankLabel="请选择" t:validate="required"/>
</t:zone>

说明: <input t:type="TextField" t:id="keywords"/> (optional)<br/>

<input type="submit" value="提交"/>
<t:errors/>

</form><br/>
</html>
```

#### AJAXSelect2.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.PersistenceConstants;
import org.apache.tapestry5.annotations.Persist;
import org.apache.tapestry5.annotations.Property;

public class AJAXSelect2 {
    @Persist(PersistenceConstants.FLASH)
    @Property
    @SuppressWarnings("unused")
    private String areaMake;

    @Persist(PersistenceConstants.FLASH)
```



```
@Property
@SuppressWarnings("unused")
private String cithModel;

@Persist(PersistenceConstants.FLASH)
@Property
@SuppressWarnings("unused")
private String keyWords;

public void set(String areaMake, String cithModel, String keyWords) {
    this.areaMake = areaMake;
    this.cithModel = cithModel;
    this.keyWords = keyWords;
}
}
```

AJAXSelect2.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
    您选择的地区是: ${areaMake} ${cithModel} ${keyWords}!<br/><br/>
</html>
```

## 第三十九讲：tapestry 页面加载等待过程 (ProgressiveDisplay)组件

ProgressiveDisplay 组件其实就是 loading。在请求没完成时调出一个等待的 gif 动画，告诉用户请求没完成请等待！代码如下：

ProgressiveDisplay.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
```



```
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.Property;

public class ProgressiveDisplay {
    static final private String[] ALL_THINGS = { "吃饭", "睡觉", "绘画" } ;

    @Property
    @SuppressWarnings("unused")
    private String[] things;

    @Property
    @SuppressWarnings("unused")
    private String thing;

    public void onProgressiveDisplayFromShowThings() {

        things = ALL_THINGS;

        sleep(5000);

    }

    private void sleep(long millis) {
        try {
            Thread.sleep(millis);
        }
        catch (InterruptedException e) {

        }
    }
}
```

ProgressiveDisplay.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<t:ProgressiveDisplay t:id="showThings" style="margin-left: 10px;">
<div t:type="Loop" t:source="things" t:value="thing">
    ${thing}
</div>
</t:ProgressiveDisplay><br/>
```



</html>

## 第四十讲：tapestry ProgressiveDisplay 组件详细使用

源码如下：

ProgressiveDisplayVariations.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.Block;
import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.ioc.annotations.Inject;

public class ProgressiveDisplayVariations {
    @Property
    @SuppressWarnings("unused")
    private int sleepTimeMillis;

    @Inject
    private Block resultSixBlock;

    public void onProgressiveDisplayFromShowOne() {
        sleep(1000);
    }

    public void onProgressiveDisplayFromShowTwo() {
        sleep(2000);
    }
}
```



```
public void onProgressiveDisplayFromShowThree(int sleepTimeMillis) {
    this.sleepTimeMillis = sleepTimeMillis;
    sleep(sleepTimeMillis);
}

public void onProgressiveDisplayFromShowFour() {
    sleep(4000);
}

public void onProgressiveDisplayFromShowFive() {
    sleep(5000);
}

public Block onProgressiveDisplayFromShowSix() {
    sleep(6000);
    return resultSixBlock;
}

private void sleep(long millis) {
    try {
        Thread.sleep(millis);
    }
    catch (InterruptedException e) {
    }
}
}
```

#### ProgressiveDisplayVariations.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<t:ProgressiveDisplay t:id="showOne" style="margin-left: 10px;">
第一个
    </t:ProgressiveDisplay><br/>

<t:ProgressiveDisplay t:id="showTwo" style="margin-left: 10px;">
第二个
    </t:ProgressiveDisplay><br/>

<br/><br/>
```



```
<t:ProgressiveDisplay      t:id="showThree"      t:context="literal:3000"
style="margin-left: 10px;">
等待 ${sleepTimeMillis} 毫秒.
</t:ProgressiveDisplay><br/>

<br/>

<t:ProgressiveDisplay      t:id="showFour"      t:initial="block:loadingBlock"
style="margin-left: 10px;">
第四个
</t:ProgressiveDisplay><br/>

<t:ProgressiveDisplay      t:id="showFive"      t:update="slidedown"
style="margin-left: 10px;">
第五个<br/>
第五个<br/>
第五个
</t:ProgressiveDisplay><br/>

<t:ProgressiveDisplay t:id="showSix" style="margin-left: 10px;"/>

<t:block t:id="loadingBlock">
<div style="font-weight: bold; text-decoration: blink;">
请等待...4000毫秒
</div>
</t:block>

<t:block t:id="resultSixBlock">
第六个
</t:block>
</html>
```

## 第四十一讲：tapestry Ajax 周期性更新

源码如下：

AJAXPeriodicUpdate.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
```





```
* 版本: 1.0
* 编写: 飞风
* 时间: 2012-02-29
*/
package com.tapestry.app.pages;

import java.util.Date;

import org.apache.tapestry5.annotations.Import;
import org.apache.tapestry5.annotations.InjectComponent;
import org.apache.tapestry5.corelib.components.Zone;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.json.JSONObject;
import org.apache.tapestry5.services.Request;
import org.apache.tapestry5.services.javascript.JavaScriptSupport;

@Import(library = "context:assets/js/AJAXPeriodicUpdate.js")
public class AJAXPeriodicUpdate {

    @InjectComponent
    private Zone timeZone;

    @Inject
    private JavaScriptSupport javascriptSupport;

    @Inject
    private Request request;

    public void setupRender() {
        javascriptSupport.addInitializerCall("periodicTimeZoneUpdater", new
        JSONObject());
    }

    Object onRefreshTimeZone() {
        return request.isXHR() ? timeZone.getBody() : null;
    }

    public Date getServerTime() {
        return new Date();
    }
}
```



#### AJAXPeriodicUpdate.tml

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<div style="margin-left: 50px">
<a t:type="eventlink" t:id="refreshTimeZone" href="#" t:zone="timeZone"
style="display: none;">Hidden EventLink</a>

<t:zone t:id="timeZone">
serverTime: ${serverTime}
</t:zone><br/>
</div>
</html>
```

#### AJAXPeriodicUpdate.js

```
PeriodicTimeZoneUpdater = Class.create( {

initialize: function() {
var eventLink = $('refreshTimeZone');

var updatesCount = 0;

new PeriodicalExecuter(function(pe) {
if (updatesCount++ < 4) {
Tapestry.findZoneManager(eventLink).updateFromURL(eventLink.href);
}
else {
pe.stop();
}
}, 3);
}

})

Tapestry.Initializer.periodicTimeZoneUpdater = function(spec) {
new PeriodicTimeZoneUpdater();
}
```



## 第四十二讲：tapestry Ajax 混合(mixins)周期性更新

源码如下：

AJAXPeriodicUpdateMixin.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import java.util.Date;

import org.apache.tapestry5.annotations.InjectComponent;
import org.apache.tapestry5.corelib.components.Zone;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.services.Request;

public class AJAXPeriodicUpdateMixin {
    @InjectComponent
    private Zone timeZone;

    @Inject
    private Request request;

    Object onRefreshTimeZone() {
        return request.isXHR() ? timeZone.getBody() : null;
    }

    public Date getServerTime() {
        return new Date();
    }
}
```

AJAXPeriodicUpdateMixin.tml



```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<div style="margin-left: 50px">
<t:zone          t:id="timeZone"          t:mixins="PeriodicZoneUpdater"
t:event="refreshTimeZone" t:frequencySecs="3" t:maxUpdates="4">
serverTime:  ${serverTime}
</t:zone>
</div><br/>
</html>
```

#### PeriodicZoneUpdater.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.mixins;

import org.apache.tapestry5.BindingConstants;
import org.apache.tapestry5.ClientElement;
import org.apache.tapestry5.ComponentResources;
import org.apache.tapestry5.annotations.Import;
import org.apache.tapestry5.annotations.InjectContainer;
import org.apache.tapestry5.annotations.Parameter;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.json.JSONObject;
import org.apache.tapestry5.services.javascript.JavaScriptSupport;

@Import(library="context:assets/js/PeriodicZoneUpdater.js")
public class PeriodicZoneUpdater {
    @Parameter(required = true, defaultPrefix = BindingConstants.LITERAL)
    private String event;

    @Parameter
    private Object[] context;

    @Parameter(defaultPrefix = BindingConstants.LITERAL, required = true)
    private int frequencySecs;
```



```
@Parameter(defaultPrefix = BindingConstants.LITERAL, required = true)
private int maxUpdates;

@Inject
private ComponentResources componentResources;

@Inject
private JavaScriptSupport javascriptSupport;

@InjectContainer
private ClientElement clientElement;

void afterRender() {

    JSONObject spec = new JSONObject();
    spec.put("zoneElementId", clientElement.getClientId());
    spec.put("eventURL",          componentResources.createEventLink(event,
context).toAbsoluteURI());
    spec.put("frequencySecs", frequencySecs);
    spec.put("maxUpdates", maxUpdates);
    javascriptSupport.addInitializerCall("periodicZoneUpdater", spec);
}
}
```

#### PeriodicZoneUpdater.js

```
PeriodicZoneUpdater = Class.create( {

    initialize : function(spec){
        this.zoneElement = $(spec.zoneElementId).getStorage();
        this.eventURL = spec.eventURL;
        this.frequencySecs = spec.frequencySecs;
        this.maxUpdates = spec.maxUpdates;

        // Use Prototype's PeriodicalExecuter to run updateZone() every 3 seconds,
        this.maxUpdates times.

        var updatesCount = 0;

        new PeriodicalExecuter(function(pe) {
            if (updatesCount++ < this.maxUpdates) {
                this.updateZone();
            }
        })
    }
}
```



```
pe.stop();
}
}.bind(this), this.frequencySecs);

},

updateZone : function() {
var zoneManager = this.zoneElement.zoneManager;
if(!zoneManager){
return;
}

zoneManager.updateFromURL(this.eventURL);
}

} )

// Extend the Tapestry.Initializer with a static method that instantiates
a PeriodicZoneUpdater.

Tapestry.Initializer.periodicZoneUpdater = function(spec) {
    new PeriodicZoneUpdater(spec);
}
```

## 第四十三讲：tapestry Ajax 表单验证

tapestry Ajax 表单验证，作用非常大可做检查用户名是否存在。源码如下：

AJAXValidators1.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import java.util.List;
```



```
import org.apache.tapestry5.annotations.InjectPage;
import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.json.JSONObject;
import org.apache.tapestry5.services.Request;

import com.tapestry.app.entities.Person;
import com.tapestry.app.services.StartDAO;

public class AJAXValidators1 {
    @Property
    private String firstName;

    @Property
    private String lastName;

    @Property
    @SuppressWarnings("unused")
    private List<Person> persons;

    @InjectPage
    private AJAXValidators2 page2;

    @Inject
    private Request request;

    @Inject
    private StartDAO dao;

    void setupRender() {
        StringBuffer sql = new StringBuffer();
        sql.append("from Person");
        persons = dao.findWithQuery(sql.toString());
    }

    JSONObject onAjaxValidateFromFirstName() {
        String firstName = request.getParameter("param");

        try {
            validateFirstNameIsUnique(firstName);
        }
        catch (Exception e) {
            return new JSONObject().put("error", e.getMessage());
        }
    }
}
```



```
}

return new JSONObject();
}

Object onSuccess() {
    page2.set(firstName, lastName);
    return page2;
}

void validateFirstNameIsUnique(String firstName) throws Exception {
    if (firstName != null) {
        List<Person> persons = dao.searchQuery(firstName);

        if (persons.size() > 0) {
            throw new Exception("姓已经存在.");
        }
    }
}

}
```

#### AJAXValidators1.tml

```
<html          t:type="layout"          title="tapestryStart      Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<div style="margin: 20px;">
<form t:type="CustomForm" t:id="inputs">
<t:errors/>
<table>
<tr>
<td><t:label for="firstName"/></td>
<td><input  t:type="TextField"  t:id="firstName"  t:validate="required,
maxLength=10" size="10"
t:mixins="ajaxValidator"/></td>
<td>(required, maxLength=10, new name only)</td>
</tr>
<tr>
<th></th>
<td colspan="2"><t:customerror for="firstName"/></td>
</tr>
<tr>
```





```
<td><t:label for="lastName"/></td>
<td><input t:type="TextField" t:id="lastName" t:validate="required,
maxLength=10" size="10"/></td>
<td>(required, maxLength=10, new name only)</td>
</tr>
<tr>
<th></th>
<td colspan="2"><t:customerror for="lastName"/></td>
</tr>
<tr>
<td></td>
<td><input type="submit" value="Display"/></td>
<td></td>
</tr>
</table>
</form><br/>
<table t:type="grid" t:source="persons" t:row="var:person"
include="firstName,lastName">[Grid here]</table>
</div>
</html>
```

#### AjaxValidator.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.mixins;

import org.apache.tapestry5.ClientElement;
import org.apache.tapestry5.ComponentResources;
import org.apache.tapestry5.annotations.Import;
import org.apache.tapestry5.annotations.InjectContainer;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.json.JSONObject;
import org.apache.tapestry5.services.javascript.JavaScriptSupport;

@Import(library = "context:assets/js/AjaxValidator.js")
public class AjaxValidator {
    @Inject
    private ComponentResources componentResources;
```



```
@Inject
private JavaScriptSupport javaScriptSupport;

@InjectContainer
private ClientElement clientElement;

void afterRender() {

    JSONObject spec = new JSONObject();
    spec.put("elementId", clientElement.getClientId());
    spec.put("listenerURI",
        componentResources.createEventLink("ajaxValidate").toAbsoluteURI());
    javaScriptSupport.addInitializerCall("ajaxValidator", spec);
}
}
```

#### AjaxValidator.js

```
AjaxValidator = Class.create( {

    initialize : function(spec) {
        this.field = $(spec.elementId);
        this.listenerURI = spec.listenerURI;

        // Set up a listener that validates the field - asynchronously in the server
        // - on change of focus

        document.observe(Tapestry.FOCUS_CHANGE_EVENT, function(event) {

            // If changing focus *within the same form* then perform validation.
            // Note that Tapestry.currentFocusField does not change
            // until after the FOCUS_CHANGE_EVENT notification.

            if (Tapestry.currentFocusField == this.field && this.field.form ==
                event.memo.form) {
                this.asyncValidateInServer();
            }

        }).bindAsEventListener(this) );
    },

    asyncValidateInServer : function() {
        var value = this.field.value;
```



```
var listenerURIWithValue = this.listenerURI;

if (value) {
    listenerURIWithValue = addRequestParameter('param', value,
    this.listenerURI);

    new Ajax.Request(listenerURIWithValue, {
        method: 'get',
        onFailure: function(t) {
            alert('Error communication with the server: ' +
            t.responseText.stripTags());
        },
        onException: function(t, exception) {
            alert('Error communication with the server: ' + exception.message);
        },
        onSuccess: function(t) {
            if (t.responseJSON.error) {
                this.field.showValidationMessage(t.responseJSON.error);
            }
        }.bind(this)
    });
}

} )

function addRequestParameter(name, value, url) {
    if (url.indexOf('?') < 0) {
        url += '?';
    } else {
        url += '&';
    }
    value = escape(value);
    url += name + '=' + value;
    return url;
}

// Extend the Tapestry.Initializer with a static method that instantiates
an AjaxValidator.

Tapestry.Initializer.ajaxValidator = function(spec) {
    new AjaxValidator(spec);
}
```



#### AJAXValidators2.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

public class AJAXValidators2 {
    private String firstName;

    private String lastName;

    public void set(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    String[] onPassivate() {
        return new String[] { firstName, lastName };
    }

    void onActivate(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public String getName() {
        return firstName + " " + lastName;
    }
}
```

#### AJAXValidators2.tml

```
<html t:type="layout" title="tapestryStart" Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
你好, ${name}
```



```
</html>
```

## 第四十四讲：tapestry upload 组件

tapestry 提供了上传组件，使用起来非常简单。代码如下：

先在 pom.xml 中引进 tapestry-upload，

```
<dependency>
    <groupId>org.apache.tapestry</groupId>
    <artifactId>tapestry-upload</artifactId>
    <version>${tapestry-release-version}</version>
</dependency>
```

然后修改服务配置 AppModule.java 把 UploadModule 类引进去，如果需要可以设置下文件上传的大小限制 UploadSymbols.FILESIZE\_MAX。

```
package com.tapestry.app.services;

import java.io.IOException;

import org.apache.tapestry5.*;
import org.apache.tapestry5.hibernate.HibernateCoreModule;
import org.apache.tapestry5.hibernate.HibernateModule;
import org.apache.tapestry5.hibernate.HibernateSymbols;
import org.apache.tapestry5.hibernate.HibernateTransactionAdvisor;
import org.apache.tapestry5.ioc.MappedConfiguration;
import org.apache.tapestry5.ioc.MethodAdviceReceiver;
import org.apache.tapestry5.ioc.OrderedConfiguration;
import org.apache.tapestry5.ioc.ServiceBinder;
import org.apache.tapestry5.ioc.annotations.Local;
import org.apache.tapestry5.ioc.annotations.Match;
import org.apache.tapestry5.ioc.annotations.SubModule;
import org.apache.tapestry5.services.Request;
import org.apache.tapestry5.services.RequestFilter;
import org.apache.tapestry5.services.RequestHandler;
import org.apache.tapestry5.services.Response;
import org.apache.tapestry5.upload.services.UploadModule;
import org.apache.tapestry5.upload.services.UploadSymbols;
import org.slf4j.Logger;

@SubModule({HibernateModule.class,
HibernateCoreModule.class,UploadModule.class})
public class AppModule
```



```
{
    public static void bind(ServiceBinder binder)
    {
        binder.bind(StartDAO.class, StartDAOImpl.class);
    }

    public static void contributeFactoryDefaults(
        MappedConfiguration<String, Object> configuration)
    {
        configuration.override(SymbolConstants.APPLICATION_VERSION,
"1.0");
    }

    public static void contributeApplicationDefaults(
        MappedConfiguration<String, Object> configuration)
    {
        configuration.add(SymbolConstants.SUPPORTED_LOCALES, "en");
        configuration.add(SymbolConstants.PRODUCTION_MODE, "false");
        configuration.add(UploadSymbols.FILESIZE_MAX, "50000000");
        configuration.add(SymbolConstants.APPLICATION_VERSION,
"1.0-SNAPSHOT");
        configuration
.add(HibernateSymbols.ENTITY_SESSION_STATE_PERSISTENCE_STRATEGY_ENABLED
,
"true");
    }

    @Match("*.DAO")
    public static void adviseTransactions(HibernateTransactionAdvisor advisor,
MethodAdviceReceiver receiver) {
        advisor.addTransactionCommitAdvice(receiver);
    }

    public RequestFilter buildTimingFilter(final Logger log)
    {
        return new RequestFilter()
        {
            public boolean service(Request request, Response response,
RequestHandler handler)
                throws IOException
            {
                long startTime = System.currentTimeMillis();
```



```
        try
        {
            return handler.service(request, response);
        } finally
        {
            long elapsed = System.currentTimeMillis() - startTime;

            log.info(String.format("Request    time:    %d    ms",
elapsed));
        }
    }
};
}

public                                     void
contributeRequestHandler(OrderedConfiguration<RequestFilter>
configuration,

                           @Local
                           RequestFilter filter)

{
    configuration.add("Timing", filter);
}
}
```

实现上传如下: TapestryUpload.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import java.io.File;

import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.upload.services.UploadedFile;

public class TapestryUpload {
```



```
@Property
private UploadedFile file;

void onSuccess(){
//上传文件到 d 盘
File copied = new File("d:\\\" + file.getFileName());
file.write(copied);
}
}
```

TapestryUpload.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<t:form>
    <t:errors/>
    <input          t:type="upload"          t:id="file"          t:value="file"
validate="required"/>
    <br/>
    <input type="submit" value="上传"/>
</t:form>
</html>
```

## 第四十五讲: tapestry upload 组件使用 input type="file" 实现上传

此教程我们使用常用的 `multipart/form-data` 同过 `post` 方式实现上传, 上传实现依旧使用了 `tapestry` 的 `UploadedFile` 组件, 我随便写了个字符串 `to`, 这个字符串可以随便命名的, 没有具体实际意义, 作用就是传一个值到上传处理页面让它知道执行了上传这个动作。还需要创建一个解码器。代码如下:

UploadTypeFile.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<form          action="UploadTypeFile/to/"          method="post"
enctype="multipart/form-data" >
```





```
<input type="file" name="filename"/>
<input type="submit" value="Submit" />
</form>
</html>
```

#### UploadTypeFile.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import java.io.File;

import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.upload.services.MultipartDecoder;
import org.apache.tapestry5.upload.services.UploadedFile;

public class UploadTypeFile {

    @Property
    private UploadedFile file;

    @Inject
    private MultipartDecoder decoder;

    void onActivate(String to){
        file = decoder.getFileUpload("filename");
        File copied = new File("d:\\", file.getFileName());
        file.write(copied);
    }

}
```



## 第四十六讲：tapestry upload 组件常用的配置详解

tapestry 的上传提供了很简洁的代码，但很多时候并不是我们想要的，例如上到的本工程目录的文件夹下怎么实现？虽然在服务里面设置了文件上传的大小，但那是所有上传的控制，能不能在某个上传的时候进行文件大小上传限制？等等诸如此类的问题。下面看下源码：

UploadDetail.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import java.io.File;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.servlet.http.HttpServletRequest;

import org.apache.tapestry5.annotations.Persist;
import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.services.ApplicationGlobals;
import org.apache.tapestry5.services.RequestGlobals;
import org.apache.tapestry5.upload.services.UploadedFile;

import com.tapestry.app.util.UiUtils;

public class UploadDetail {

    @Inject
    private ApplicationGlobals applicationGlobals;

    @Inject
    private RequestGlobals requestGlobals;
```



```
@Property
private UploadedFile file;

@Persist
@Property
private String dataUrl;

@Persist
@Property
private String tooBig;

protected final HttpServletRequest getRequest() {
    return requestGlobals.getHttpServletRequest();
}

void onSuccess(){
    uploadImage();
}

void uploadImage(){
    // 文件保存目录路径
    String savePath =
    applicationGlobals.getServletContext().getRealPath("/uploadImages/") +
    "\\\";

    // 文件保存目录 URL 地址
    String saveUrl = getRequest().getContextPath() + "/uploadImages/";

    //最大文件大小
    long maxSize = 1048576;

    // 检查目录
    File uploadDir = new File(savePath);

    if (!uploadDir.exists()) {
        uploadDir.mkdirs();
    }

    // 检查目录写权限
    if (!uploadDir.canWrite()) {
        System.out.println("上传目录没有写权限。");
        return;
    }
}
```



```
}

//文件夹名
String dirName = "uploadData";
// 创建文件夹
savePath += dirName + "/";
saveUrl += dirName + "/";
File saveDirFile = new File(savePath);
if (!saveDirFile.exists()) {
    saveDirFile.mkdirs();
}

String fileName = file.getFileName();
//转换文件名为时间_随机数的名字格式如21120229_8888745544.gif
SimpleDateFormat df = new SimpleDateFormat("yyyyMMddHHmmss");
String newFileName = df.format(new Date()) + "_" +
    UiUtils.createFileName(fileName);

File copied = new File(savePath + newFileName);

long fileSize = file.getSize();

//判断文件大小
if(fileSize > maxSize){
    tooBig = "上传失败！文件大小超过1M。";
    System.out.println("上传失败！文件大小超过1M。");
    return;
}else{
    tooBig = "";
}

file.write(copied);

//图片上传成功之后返回的途径
dataUrl = saveUrl + newFileName;

}

}
```

UploadDetail.tml

<html t:type="layout" title="tapestryStart Index"



```
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
文件路径: ${dataUrl}<br/>

<t:form t:id="uploadForm">
    <t:errors/>
    <input t:type="upload" t:id="file" t:value="file"
validate="required"/><span style="color:red">${tooBig}</span>
    <br/>
    <input type="submit" value="上传"/>
</t:form>

</html>
```

#### UiUtils.java

```
package com.tapestry.app.util;

import java.util.Date;
import java.util.Random;

public class UiUtils {
    public static String splitString(String str, String split) {
        int index = str.lastIndexOf(split);
        return str.substring(index + 1);
    }

    public static String splitStringPath(String str) {
        str = str.replace("/", "\\");
        int index = str.lastIndexOf("\\");
        return str.substring(index + 1);
    }

    public static String createFileName(String name) {
        Long randomNum = new Random().nextLong();
        String radomStr = randomNum.toString().substring(2, 5);
        String accessoryAutoName = new Date().getTime() + radomStr;
        String extendedName = splitString(name, ".");
        String fileName = accessoryAutoName;
        if (extendedName != null) {
            fileName = accessoryAutoName + "." + extendedName;
        }
        return fileName;
    }
}
```



```
}
```

## 第四十七讲：tapestry 与 swfupload 上传

tapestry upload 上传虽然简单易用，但并不能显示上传进度。swfupload 是 flash 结合 js 写的上传，上传进度一目了然，下面是 tapestry 整合 swfupload 上传的代码：

首先去 swfupload 官网 <http://www.swfupload.org/> 下载文件，解压之后可看到，上传例子很多就以 Simple Upload Demo 为例，使用 tapestry 制作 flash 上传。

第一步：把 swfupload.js、swfupload.queue.js、fileprogress.js、handlers.js、swfupload.swf 拷贝到工程静态目录 webapp\assets\js\swfupload 目录下，文件夹没有的话可以自己在 webapp 下创建它。

第二步：把 TestImageNoText\_65x29.png 图片拷贝到 webapp\assets\images 目录下，把 default.css 拷贝到 webapp\assets\style 目录下面。

以上步骤完成之后就下面代码：

SWFUpload.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.Import;

@Import(library={"context:assets/js/swfupload/swfupload.js",
"context:assets/js/swfupload/swfupload.queue.js",
"context:assets/js/swfupload/fileprogress.js",
"context:assets/js/swfupload/handlers.js"},
stylesheet="context:assets/style/default.css")
public class SWFUpload {

}
```



#### SWFUpload.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<script type="text/javascript">
var swfu;

window.onload = function() {
var settings = {
flash_url : "${context:assets/js/swfupload/swfupload.swf}",
upload_url: "SWFUploadHandler/",
//post_params: {"PHPSESSID" : "<?php echo session_id(); ?>"},
file_size_limit : "100 MB",
file_types : " *.*",
file_types_description : "All Files",
file_upload_limit : 100,
file_queue_limit : 0,
custom_settings : {
progressTarget : "fsUploadProgress",
cancelButtonId : "btnCancel"
},
debug: false,

// Button settings
button_image_url: "${context:assets/images/TestImageNoText_65x29.png}",
button_width: "65",
button_height: "29",
button_placeholder_id: "spanButtonPlaceHolder",
button_text: '<span class="theFont">上传</span>',
button_text_style: ".theFont { font-size: 16; }",
button_text_left_padding: 12,
button_text_top_padding: 3,

// The event handler functions are defined in handlers.js
file_queued_handler : fileQueued,
file_queue_error_handler : fileQueueError,
file_dialog_complete_handler : fileDialogComplete,
upload_start_handler : uploadStart,
upload_progress_handler : uploadProgress,
upload_error_handler : uploadError,
upload_success_handler : uploadSuccess,
upload_complete_handler : uploadComplete,
```



```
queue_complete_handler : queueComplete // Queue plugin event
};

swfu = new SWFUpload(settings);
};
</script>
<div id="content">
<h2>Simple Demo</h2>
<form id="form1" action="index.php" method="post"
enctype="multipart/form-data">
<p>This page demonstrates a simple usage of SWFUpload. It uses the Queue
Plugin to simplify uploading or cancelling all queued files.</p>

<div class="fieldset flash" id="fsUploadProgress">
<span class="legend">Upload Queue</span>
</div>
<div id="divStatus">0 Files Uploaded</div>
<div>
<span id="spanButtonPlaceHolder"></span>
<input id="btnCancel" type="button" value="Cancel All Uploads"
onclick="swfu.cancelQueue();" disabled="disabled" style="margin-left: 2px;
font-size: 8pt; height: 29px;" />
</div>

</form>
</div>
</html>
```

#### SWFUploadHandler.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import java.io.File;
import java.text.SimpleDateFormat;
import java.util.Date;

import javax.servlet.http.HttpServletRequest;
```





```
import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.services.ApplicationGlobals;
import org.apache.tapestry5.services.RequestGlobals;
import org.apache.tapestry5.upload.services.MultipartDecoder;
import org.apache.tapestry5.upload.services.UploadedFile;

import com.tapestry.app.util.UiUtils;

public class SWFUploadHandler {

    @Inject
    private ApplicationGlobals applicationGlobals;

    @Inject
    private RequestGlobals requestGlobals;

    @SuppressWarnings("unused")
    @Property
    private UploadedFile file;

    @Inject
    private MultipartDecoder decoder;

    void onActivate() throws Exception{
        UploadedFile file = decoder.getFileUpload("Filedata");

        // 文件保存目录路径
        String savePath = applicationGlobals.getServletContext().getRealPath("/uploadImages/") + "\\";

        // 文件保存目录 UR
        //String saveUrl = getRequest().getContextPath() + "/uploadImages/";

        //最大文件大小
        long maxSize = 1048576;

        // 检查目录
        File uploadDir = new File(savePath);

        if (!uploadDir.exists()) {
```



```
uploadDir.mkdirs();
}

// 检查目录写权限
if (!uploadDir.canWrite()) {
    System.out.println("上传目录没有写权限。");
    return;
}

//文件夹名字
String dirName = "swfData";
// 创建文件夹
savePath += dirName + "/";
//saveUrl += dirName + "/";
File saveDirFile = new File(savePath);
if (!saveDirFile.exists()) {
    saveDirFile.mkdirs();
}

String fileName = file.getFileName();

SimpleDateFormat df = new SimpleDateFormat("yyyyMMddHHmmss");
String newFileName = df.format(new Date()) + "_" +
    UiUtils.createFileName(fileName);

File copied = new File(savePath + newFileName);

long fileSize = file.getSize();

if(fileSize > maxSize){
    System.out.println("上传文件大小超过限制。");
    return;
}

file.write(copied);

}

protected final HttpServletRequest getRequest() {
    return requestGlobals.getHttpServletRequest();
}

}
```



#### SWFUploadHandler.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">

</html>
```

#### UiUtils.java

```
package com.tapestry.app.util;

import java.util.Date;
import java.util.Random;

public class UiUtils {
    public static String splitString(String str, String split) {
        int index = str.lastIndexOf(split);
        return str.substring(index + 1);
    }

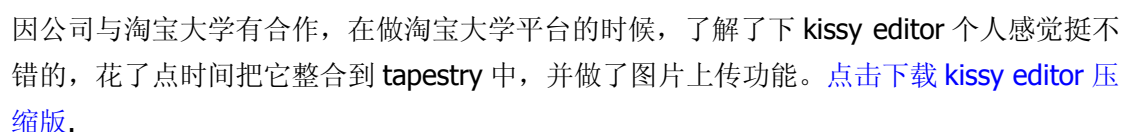
    public static String splitStringPath(String str) {
        str = str.replace("/", "\\");
        int index = str.lastIndexOf("\\");
        return str.substring(index + 1);
    }

    public static String createFileName(String name) {
        Long randomNum = new Random().nextLong();
        String radomStr = randomNum.toString().substring(2, 5);
        String accessoryAutoName = new Date().getTime() + radomStr;
        String extendedName = splitString(name, ".");
        String fileName = accessoryAutoName;
        if (extendedName != null) {
            fileName = accessoryAutoName + "." + extendedName;
        }
        return fileName;
    }
}
```



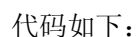
tapestry 没有富文本编辑器组件，就是官方推荐的 easyfck 也是鸡肋，上传图片经常出错，

浏览器兼容性也不好！最让人难受的是版本兼容性太差！！



下载之后解压，把 **build** 放到 **webapp/assets/**目录下面。之后就看下面源码吧！

最后效果如图:



KissyEditor.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+MySQL
 * 网址: http://www.flywind.org
 * 版本: 1.0
```



```
* 编写: 飞风
* 时间: 2012-02-29
*/
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.Import;
import org.apache.tapestry5.annotations.PageActivationContext;
import org.apache.tapestry5.annotations.Persist;
import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.services.javascript.JavaScriptSupport;

@Import(library = { "context:assets/build/kissy-min.js",
"context:assets/build/uibase/uibase-pkg-min.js",
"context:assets/build/dd/dd-pkg-min.js",
"context:assets/build/overlay/overlay-pkg-min.js",
"context:assets/build/editor-all-pkg-min.js",
"context:assets/build/biz/ext/editor-plugin-pkg-min.js",
"context:assets/build/TEstart.js"})
public class KissyEditor {

@Inject
private JavaScriptSupport javaScriptSupport;

@Property
@Persist
private String content;

/*这个 eid 是传给上传请求页面的, 当多个文本编辑器同是使用一个上传请求时
*通过这个值的设置可以区别他们是那个文本编辑器的请求
*这里让它上传图片到 work 目录下
**/
@PageActivationContext
private static String eid = "work";

void afterRender() {
javaScriptSupport.addScript("start('" + eid + "');");
}

void onSuccess(){

}

}
```



#### KissyEditor.tml

```
<html          t:type="layout"          title="tapestryStart          Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<link          href="${context:assets/build/theme/base-min.css}"
rel="stylesheet"/>
<link
href="${context:assets/build/theme/cool/editor-pkg-sprite-min.css}"
id="cssLink" rel="stylesheet"/>
<t:form>
<t:textarea t:id="editor" tabindex="0" value="content" style="height:260px;
width:100%"></t:textarea>
<input type="submit" value="提交"/>
</t:form>
<t:OutputRaw value="content"></t:OutputRaw>
</html>
```

#### KissyEditorUploadHandler.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 网址: http://www.flywind.org
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import java.io.File;
import java.text.SimpleDateFormat;
import java.util.Date;

import javax.servlet.http.HttpServletRequest;

import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.json.JSONObject;
import org.apache.tapestry5.services.ApplicationGlobals;
import org.apache.tapestry5.services.Context;
import org.apache.tapestry5.services.RequestGlobals;
import org.apache.tapestry5.upload.services.MultipartDecoder;
```



```
import org.apache.tapestry5.upload.services.UploadedFile;

import com.tapestry.app.util.UiUtils;

public class KissyEditorUploadHandler {
    @SuppressWarnings("unused")
    @Inject
    private Context context;

    private String outUrl;

    @Inject
    private ApplicationGlobals applicationGlobals;

    @Inject
    private RequestGlobals requestGlobals;

    @SuppressWarnings("unused")
    @Property
    private UploadedFile file;

    @Inject
    private MultipartDecoder decoder;

    void onActivate(String eid) throws Exception {

        UploadedFile file = decoder.getFileUpload("Filedata");

        // 文件保存目录路径
        String savePath = applicationGlobals.getServletContext().getRealPath(
            "/uploadImages/")
            + "\\\\";

        // 文件保存目录 UR
        String saveUrl = getRequest().getContextPath() + "/uploadImages/";

        //最大文件大小
        long maxSize = 1048576;//1M

        // 检查目录
        File uploadDir = new File(savePath);
```



```
if (!uploadDir.exists()) {
    uploadDir.mkdirs();
}

// 检查目录写权限
if (!uploadDir.canWrite()) {
    System.out.println(getError("上传目录没有写权限。"));
    return;
}

String dirName = eid;
// 创建文件夹
savePath += dirName + "/";
saveUrl += dirName + "/";
File saveDirFile = new File(savePath);
if (!saveDirFile.exists()) {
    saveDirFile.mkdirs();
}

String fileName = file.getFileName();

SimpleDateFormat df = new SimpleDateFormat("yyyyMMddHHmmss");
String newFileName = df.format(new Date()) + "_"
+ UiUtils.createFileName(fileName);

File copied = new File(savePath + newFileName);

long fileSize = file.getSize();

if(fileSize > maxSize){
    System.out.println(getError("上传文件大小超过限制。"));
    return;
}

file.write(copied);

System.out.println(saveUrl);

try {
    String url = saveUrl + newFileName;
    setOutUrl("{\"status\":\"0\" , \"imgUrl\" :\"" + url + "\"}");
```





```
} catch (Exception e) {
    setOutUrl("{\"status\": \"1\" , \"error\" : \" error \"}");
}

}

private String getError(String message) {
    JSONObject obj = new JSONObject();
    obj.put("error", 1);
    obj.put("message", message);
    return obj.toString();
}

protected final HttpServletRequest getRequest() {
    return requestGlobals.getHttpServletRequest();
}

public String getOutUrl() {
    return outUrl;
}

public void setOutUrl(String outUrl) {
    this.outUrl = outUrl;
}
}
```

#### KissyEditorUploadHandler.tml

```
<html xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd">
<t:OutputRaw value="outUrl"></t:OutputRaw>
</html>
```

#### UiUtils.java

```
package com.tapestry.app.util;

import java.util.Date;
import java.util.Random;

public class UiUtils {
    public static String splitString(String str, String split) {
        int index = str.lastIndexOf(split);
        return str.substring(index + 1);
    }
}
```



```
public static String splitStringPath(String str) {
    str = str.replace("/", "\\");
    int index = str.lastIndexOf("\\");
    return str.substring(index + 1);
}

public static String createFileName(String name) {
    Long randomNum = new Random().nextLong();
    String radomStr = randomNum.toString().substring(2, 5);
    String accessoryAutoName = new Date().getTime() + radomStr;
    String extendedName = splitString(name, ".");
    String fileName = accessoryAutoName;
    if (extendedName != null) {
        fileName = accessoryAutoName + "." + extendedName;
    }
    return fileName;
}
}
```

#### TStart.js

```
function userAgent() {
    var ua = navigator.userAgent;
    ua = ua.toLowerCase();
    var match = /(webkit)[ \\/]([\w.]+)/.exec(ua)
    || /(opera)(?:.*version)?[ \\/]([\w.]+)/.exec(ua)
    || /(msie) ([\w.]+)/.exec(ua) || !/compatible/.test(ua)
    && /(mozilla)(?:.*? rv:([\w.]+))?/.exec(ua) || [];
    // 如果需要获取浏览器版本号: match[2]
    switch (match[1]) {
        case "msie" : // ie
            if (parseInt(match[2]) > 8) // ie6
            {
                $('cssLink').href =
                "${context:assets/build/theme/cool/editor-pkg-min-datauri.css}";
            }
            break;
        case "webkit" : // safari or chrome
            break;
        case "opera" : // opera
            break;
        case "mozilla" : // Firefox
            break;
        default :
```



```
break;
}
}
function start(id) {
userAgent();
KISSY.ready(function(S) {
S.use('editor', function() {

var KE = KISSY.Editor;
var cfg = {
attachForm : true,
baseZIndex : 10000,
// 自定义样式
// customStyle:"p{color:purple;}",
// 自定义外部样式
//
customLink:["http://localhost/customLink.css","http://xx.com/y2.css"],
// 是否一开始自动聚焦
// focus:true,
pluginConfig : {
"image" : {
upload : {

serverUrl : "/tapestryStart/KissyEditorUploadHandler/" + id,
serverParams : {

cookie : document.cookie,
waterMark : function() {
return S.one("#ke_img_up_watermark_1")[0].checked;
}
},
surfix : "png,jpg,jpeg,gif",
fileInput : "Filedata",
sizeLimit : 1024,// k
extraHtml      :      "<p      style='margin-top:10px;display:none;'><input
type='checkbox' id='ke_img_up_watermark_1' checked='checked'> 图片加水印,
防止别人盗用</p>"
}
},
"flash" : {
defaultWidth : "300",
defaultHeight : "300"
},
},
```



```
"templates" : [{
  demo : "模板1效果演示 html",
  html : "<div style='border:1px solid red'>模板1效果演示 html</div><p></p>"
}, {
  demo : "模板2效果演示 html",
  html : "<div style='border:1px solid red'>模板2效果演示 html</div>"
}],
// "font-size":false,
// "font-family":false,
// "font-bold":false,
// "font-italic":false,
// "font-underline":false,

// "font-strikeThrough":{
// style:{
// element : 'strike',
// overrides : [
// {element : 'span',
// attributes : { style:'text-decoration: line-through;' }},
// { element : 's' },
// { element : 'del' }
// ]
// }
// },
"multi-upload" : {
  serverUrl : "/code/upload/upload.jsp",
  serverParams : {
    waterMark : function() {
      return S.one("#ke_img_up_watermark_2")[0].checked;
    }
  },
  // previewSuffix:"_60x60",
  previewWidth : "80px",
  sizeLimit : 1000// k,
  ,
  numberLimit : 15,
  extraHtml : "<p style='margin-top:10px;'>"
+ "<input type='checkbox' "
+ "style='vertical-align:middle;margin:0 5px;' "
+ "id='ke_img_up_watermark_2'>"
+ "<span style='vertical-align:middle;'>图片加水印,防止别人盗用</span></p>"
},
"video" : {
```



飞风娱乐网 <http://www.flywind.org>

```
urlCfg : [{
  reg : /tudou\.com/i,
  url : "http://bangpai.taobao.com/json/getTudouVideo.htm?"
  + "url=@url@&callback=@callback@"// "&rand=@rand@"
}],
urlTip : "请输入优酷网、土豆网、酷7网的视频播放页链接...",
providers : [{
  reg : /youku\.com/i,
  width : 480,
  height : 400,
  detect : function(url) {
    var m = url.match(/id_(\[^\.\]+)\.html$/);
    if (m) {
      return "http://player.youku.com/player.php/sid/"
      + m[1] + "/v.swf";
    }
    m = url.match(/v_playlist\/(\[^\.\]+)\.html$/);
    if (m) {
      return;
      // return
      // "http://player.youku.com/player.php/sid/"
      // + m[1] + "/v.swf";
    }
    return url;
  }, {
    reg : /tudou\.com/i,
    width : 480,
    height : 400,
    detect : function(url) {
      return url;
    }, {
      reg : /ku6\.com/i,
      width : 480,
      height : 400,
      detect : function(url) {
        var m = url.match(/show\[^\.\]+\/(\[^\.\]+)\.html$/);
        if (m) {
          return "http://player.ku6.com/refer/"
          + m[1] + "/v.swf";
        }
        return url;
      }
    }
  ]
}
```



```
}
}/*
* , { reg:/taobaocdn\.com/i, width:480, height:400,
* detect:function(url) { return url; } }
*/
]
},
music : {
// 必须和网址 url 同域而不是类库同域
musicPlayer : KE.Config.base + "music/niftyplayer.swf"
},
"draft" : {
interval : 5,
limit : 10,
helpHtml : "<div "
+ "style='width:200px;'"
+ "<div style='padding:5px;'">草稿箱能够自动保存您最新编辑的内容, "
+ "如果发现内容丢失, " + "请选择恢复编辑历史</div></div>"
},
"resize" : {
direction : ["y"]
},

dragupload : {
surfix : "png,jpg,jpeg,gif",
fileInput : "Filedata",
sizeLimit : 1000,
serverUrl : "/code/upload/web/upload.jsp",
serverParams : {
waterMark : function() {
return true;
}
}
}
}
};

// if(1>2)
function test() {
window.editor2 = KE("#editor", S.clone(cfg))
.use("elementpaths," + "sourcearea,preview,"
+ "checkbox-sourcearea,"
+ "templates,separator," + "undo,separator,"
```



```
+ "removeformat,font,format,color,separator,"
+ "list,indent," + "justify,separator,link,"
+ "image,flash," + "video," + "music,"
+ "smiley,separator,table,resize," + "draft,"
+ "pagebreak,separator,maximize,dragupload");

}

test();
});
});
}
```

## 第四十九讲: **tapestry+spring** 整合

tapestry 与 spring 整合超级简单, tapestry ioc 虽然很出众, 但是考虑到系统开发功能越来越多, 数据库访问层(dao)与服务层(业务层)越来越复杂, 安全等等问题。这时候你就会想到 spring 在这方面的突出的表现。于是得出一个结论: 庞大的系统开发首选与 spring 结合(如中兴商旅 e 路通、电子报账、淘宝大学学习平台等等项目), 简单的系统直接就用 tapestry ioc 就好了。下面来看下整合代码:

1, 修改 webapp/WEB-INF 目录下的 web.xml, 改成 tapestry+spring 框架。修改 pom.xml 增加 spring 引入代码如下:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
    <display-name>tapestrySpringStart          Tapestry          5
Application</display-name>
    <context-param>
        <param-name>tapestry.app-package</param-name>
        <param-value>com.tapestry.app</param-value>
    </context-param>
    <context-param>
        <param-name>tapestry.development-modules</param-name>
        <param-value>
            com.tapestry.app.services.DevelopmentModule
```



```
        </param-value>
    </context-param>
    <context-param>
        <param-name>tapestry.qa-modules</param-name>
        <param-value>
            com.tapestry.app.services.QaModule
        </param-value>
    </context-param>
    <filter>
        <filter-name>app</filter-name>

<filter-class>org.apache.tapestry5.spring.TapestrySpringFilter</filter-
class>
    </filter>
    <filter-mapping>
        <filter-name>app</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
</web-app>
```

#### pom.xml

```
<project          xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://maven.apache.org/POM/4.0.0">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.tapestry</groupId>
    <artifactId>tapestrySpringStart</artifactId>
    <version>1.0</version>
    <packaging>war</packaging>
    <name>tapestrySpringStart Tapestry 5 Application</name>
    <dependencies>
        <!-- Too set up an application with a database, change the artifactId
below to
            tapestry-hibernate, and add a dependency on your JDBC driver.
You'll also
            need to add Hibernate configuration files, such as
hibernate.cfg.xml. -->
        <dependency>
            <groupId>org.apache.tapestry</groupId>
            <artifactId>tapestry-core</artifactId>
            <version>${tapestry-release-version}</version>
```





```
</dependency>

<!-- This adds automatic compression of JavaScript and CSS when in
production mode. -->
<dependency>
  <groupId>org.apache.tapestry</groupId>
  <artifactId>tapestry-yuicompressor</artifactId>
  <version>${tapestry-release-version}</version>
</dependency>

<dependency>
  <groupId>org.apache.tapestry</groupId>
  <artifactId>tapestry-spring</artifactId>
  <version>${tapestry-release-version}</version>
</dependency>

<!-- Uncomment this to add support for file uploads:

<dependency>
  <groupId>org.apache.tapestry</groupId>
  <artifactId>tapestry-upload</artifactId>
  <version>${tapestry-release-version}</version>
</dependency>

-->

<!-- A dependency on either JUnit or TestNG is required, or the
surefire plugin (which runs the tests)
will fail, preventing Maven from packaging the WAR. Tapestry includes a
large number
of testing facilities designed for use with TestNG (http://testng.org/),
so it's recommended. -->
<dependency>
  <groupId>org.testng</groupId>
  <artifactId>testng</artifactId>
  <version>${testng-release-version}</version>
  <scope>test</scope>
</dependency>

<dependency>
  <groupId>org.easymock</groupId>
  <artifactId>easymock</artifactId>
  <version>${easymock-release-version}</version>
```



```
<scope>test</scope>
</dependency>

<dependency>
  <groupId>org.apache.tapestry</groupId>
  <artifactId>tapestry-test</artifactId>
  <version>${tapestry-release-version}</version>
  <scope>test</scope>
</dependency>

<!-- Provided by the servlet container, but sometimes referenced
in the application
code. -->
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>servlet-api</artifactId>
  <version>${servlet-api-release-version}</version>
  <scope>provided</scope>
</dependency>

<!-- Provide dependency to the Tapestry javadoc taglet which replaces
the Maven component report -->
<dependency>
  <groupId>org.apache.tapestry</groupId>
  <artifactId>tapestry-javadoc</artifactId>
  <version>${tapestry-release-version}</version>
  <scope>provided</scope>
</dependency>

</dependencies>
<build>
  <finalName>tapestrySpringStart</finalName>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>2.3.2</version>
      <configuration>
        <source>1.5</source>
        <target>1.5</target>
        <optimize>true</optimize>
      </configuration>
    </plugin>
```



```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>2.7.2</version>
  <configuration>
    <systemPropertyVariables>

<tapestry.execution-mode>Qa</tapestry.execution-mode>
    </systemPropertyVariables>
  </configuration>
</plugin>

<!-- Run the application using "mvn jetty:run" -->
<plugin>
  <groupId>org.mortbay.jetty</groupId>
  <artifactId>maven-jetty-plugin</artifactId>
  <version>6.1.16</version>
  <configuration>
    <!-- Log to the console. -->
    <requestLog
implementation="org.mortbay.jetty.NCSARequestLog">
      <!-- This doesn't do anything for Jetty, but is a
workaround for a Maven bug
      that prevents the requestLog from being set. -->
      <append>true</append>
    </requestLog>
    <systemProperties>
      <systemProperty>
        <name>tapestry.execution-mode</name>
        <value>development</value>
      </systemProperty>
    </systemProperties>
  </configuration>
</plugin>
</plugins>
</build>

<reporting/>

<repositories>

  <!-- This repository is only needed when the Tapestry version is
```



```
a preview release, rather
    than a final release. -->
    <repository>
        <id>apache-staging</id>

<url>https://repository.apache.org/content/groups/staging/</url>
    </repository>
</repositories>

<properties>
    <tapestry-release-version>5.3.2</tapestry-release-version>
    <servlet-api-release-version>2.5</servlet-api-release-version>
    <testng-release-version>5.14.9</testng-release-version>
    <easymock-release-version>3.0</easymock-release-version>
</properties>
</project>
```

2, 在 webapp/WEB-INF 目录下增加 applicationContext.xml 文件作为管理 spring bean 的文件, 代码如下:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context-3.0.xsd">

    <context:annotation-config/>
</beans>
```

3, 修改 AppModule.java 服务, 增加 SpringModule.class 类, 代码如下:

```
package com.tapestry.app.services;

import java.io.IOException;

import org.apache.tapestry5.*;
import org.apache.tapestry5.ioc.MappedConfiguration;
import org.apache.tapestry5.ioc.OrderedConfiguration;
```



```
import org.apache.tapestry5.ioc.ServiceBinder;
import org.apache.tapestry5.ioc.annotations.Local;
import org.apache.tapestry5.ioc.annotations.SubModule;
import org.apache.tapestry5.services.Request;
import org.apache.tapestry5.services.RequestFilter;
import org.apache.tapestry5.services.RequestHandler;
import org.apache.tapestry5.services.Response;
import org.apache.tapestry5.spring.ApplicationContextCustomizer;
import org.apache.tapestry5.spring.SpringModule;
import org.slf4j.Logger;

@SubModule(SpringModule.class)
public class AppModule
{
    public static void bind(ServiceBinder binder)
    {

    }

    public static void contributeFactoryDefaults(
        MappedConfiguration<String, Object> configuration)
    {

        configuration.override(SymbolConstants.APPLICATION_VERSION,
"1.0");
    }

    public static void contributeApplicationDefaults(
        MappedConfiguration<String, Object> configuration)
    {
        configuration.add(SymbolConstants.PRODUCTION_MODE, "false");
        configuration.add(SymbolConstants.SUPPORTED_LOCALES, "en,
zh_CN");
    }

    public RequestFilter buildTimingFilter(final Logger log)
    {
        return new RequestFilter()
        {
            public boolean service(Request request, Response response,
RequestHandler handler)
                throws IOException
```



```
        {
            long startTime = System.currentTimeMillis();

            try
            {
                return handler.service(request, response);
            } finally
            {
                long elapsed = System.currentTimeMillis() - startTime;

                log.info(String.format("Request time: %d ms",
elapsed));
            }
        }
    };
}

    public void
contributeRequestHandler(OrderedConfiguration<RequestFilter>
configuration,
                        @Local
                        RequestFilter filter)
    {
        configuration.add("Timing", filter);
    }
}
```

4, 到此 **tapestry+spring** 已经整合完毕, 下一步我们测试下, 在项目中新建 **dao** 包, 用来放数据库访问层代码, 在包内创建一个接口和实现方法类分别为: **PersonDao.java** 与 **PersonDaoBean.java** 代码如下:

**PersonDao.java**

```
package com.tapestry.app.dao;

public interface PersonDao {

    public void add();

}
```

**PersonDaoBean.java**

```
package com.tapestry.app.dao;
```



```
public class PersonDaoBean implements PersonDao {

    public void add(){
        System.out.println("执行 add()方法");
    }
}
```

5, 新建服务接口类与实现类分别为: **PersonService.java** 与 **PersonServiceBean.java**, 并且修改 **applicationContext.xml**, 代码如下:

**PersonService.java**

```
package com.tapestry.app.services;

import java.util.Set;

public interface PersonService {

    public void save();

    public Set<String> getSets();

    public List<String> getLists();

    public Properties getProperties() ;

    public Map<String, String> getMaps();

}
```

**PersonServiceBean.java**

```
package com.tapestry.app.services;

import java.util.HashSet;
import java.util.Set;

import com.tapestry.app.dao.PersonDao;

public class PersonServiceBean implements PersonService {

    //注入 bean
    private PersonDao personDao;

    //注入 set 集合类
```



```
private Set<String> sets = new HashSet<String>();

private List<String> lists = new ArrayList<String>();

private Properties properties = new Properties();

private Map<String, String> maps = new HashMap<String, String>();

public List<String> getLists() {
    return lists;
}

public void setLists(List<String> lists) {
    this.lists = lists;
}

public Properties getProperties() {
    return properties;
}

public void setProperties(Properties properties) {
    this.properties = properties;
}

public Map<String, String> getMaps() {
    return maps;
}

public void setMaps(Map<String, String> maps) {
    this.maps = maps;
}

//注入对象
private String hello;

public PersonDao getPersonDao() {
    return personDao;
}

public void setPersonDao(PersonDao personDao) {
    this.personDao = personDao;
}
```





```
public void save(){
    System.out.println(hello);
    personDao.add();
}

public String getHello() {
    return hello;
}

public void setHello(String hello) {
    this.hello = hello;
}

public Set<String> getSets() {
    return sets;
}

public void setSets(Set<String> sets) {
    this.sets = sets;
}
}
```

#### applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:context="http://www.springframework.org/schema/context"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-3.0.xsd">

    <context:annotation-config/>
    <bean                                id="personService"
class="com.tapestry.app.services.PersonServiceBean">
    <!--
        <constructor-arg                                index="0"
type="com.tapestry.app.dao.PersonDao" ref="personDao"/>
        <constructor-arg index="1" value="通过构造器注入"/>
    -->
    <property name="personDao">
```



```
<bean class="com.tapestry.app.dao.PersonDaoBean"/>
</property>
<property name="hello" value="你好! "/>
<property name="sets">
<set>
<value>第一个</value>
<value>第2个</value>
</set>
</property>
<property name="lists">
<list>
<value>第一个 list 元素</value>
<value>第二个 list 元素</value>
<value>第三个 list 元素</value>
</list>
</property>
<property name="properties">
<props>
<prop key="key1">value1</prop>
<prop key="key2">value2</prop>
<prop key="key3">value3</prop>
</props>
</property>
<property name="maps">
<map>
<entry key="key-1" value="value-1"/>
<entry key="key-2" value="value-2"/>
<entry key="key-3" value="value-3"/>
</map>
</property>
</bean>
</beans>
```

6, 新建测试页面: About.java 与 About.tml 代码如下:

About.java

```
package com.tapestry.app.pages;

import org.apache.tapestry5.ioc.annotations.Inject;
import com.tapestry.app.services.PersonService;

public class About
{
```



```
@Inject
private PersonService personService;

void setupRender(){
personService.save();
for(String str : personService.getSets()){
System.out.println(str);
}
System.out.println("=====list=====");
for(String value : personService.getLists()){
System.out.println(value);
}
System.out.println("=====properties=====");
for(Object key : personService.getProperties().keySet()){
System.out.println(key+"="+
personService.getProperties().getProperty((String)key));
}
System.out.println("=====map=====");
for(String key : personService.getMaps().keySet()){
System.out.println(key+"="+ personService.getMaps().get(key));
}
}
}
```

#### About.tml

```
<html t:type="layout" title="About tapestrySpringStart"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
</html>
```

7, 运行 <http://localhost:8080/tapestrySpringStart/about> 在 eclipse 控制台中可以看到结果为:

你好!

执行 add()方法

第一个

第2个

=====list=====

第一个 list 元素

第二个 list 元素

第三个 list 元素

=====properties=====



```
key3=value3
key2=value2
key1=value1
=====map=====
key-1=value-1
key-2=value-2
key-3=value-3
```

## 第五十讲：tapestry 自定义 404 错误页面

代码如下：

Error404.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.annotations.Property;
import org.apache.tapestry5.ioc.annotations.Inject;
import org.apache.tapestry5.services.Request;

public class Error404 {
    @Property
    @Inject
    private Request request;
}
```

Error404.tml

```
<html t:type="layout" title="tapestryStart Index"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
<h1>Page or resource not found.</h1>
<t:renderobject object="request"/>
```



</html>

最后修改下 web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
    xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
    version="2.4">
    <display-name>tapestryStart Tapestry 5 Application</display-name>
    <context-param>
        <param-name>tapestry.app-package</param-name>
        <param-value>com.tapestry.app</param-value>
    </context-param>
    <context-param>
        <param-name>tapestry.development-modules</param-name>
        <param-value>
            com.tapestry.app.services.DevelopmentModule
        </param-value>
    </context-param>
    <context-param>
        <param-name>tapestry.qa-modules</param-name>
        <param-value>
            com.tapestry.app.services.QaModule
        </param-value>
    </context-param>
    <filter>
        <filter-name>app</filter-name>
        <filter-class>org.apache.tapestry5.TapestryFilter</filter-class>
    </filter>
    <filter-mapping>
        <filter-name>app</filter-name>
        <url-pattern>/*</url-pattern>
        <dispatcher>REQUEST</dispatcher>
        <dispatcher>ERROR</dispatcher>
    </filter-mapping>
    <error-page>
        <error-code>404</error-code>
        <location>/error404</location>
    </error-page>
</web-app>
```



## 第五十一讲：tapestry 自定义异常页面

代码如下：

ExceptionReport.java

```
/**
 * 项目名称: TapestryStart
 * 开发模式: Maven+Tapestry5.x+Tapestry-hibernate+Mysql
 * 版本: 1.0
 * 编写: 飞风
 * 时间: 2012-02-29
 */
package com.tapestry.app.pages;

import org.apache.tapestry5.services.ExceptionReporter;

public class ExceptionReport implements ExceptionReporter
{
    private String error;

    public void reportException(Throwable exception)
    {
        error = exception.getLocalizedMessage();
    }

    public String getError()
    {
        return error;
    }
}
```

ExceptionReport.tml

```
<html t:type="layout" title="tapestryStart" Index="1"
t:sidebarTitle="Framework Version"
xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xsd"
xmlns:p="tapestry:parameter">
    异常: ${error}
</html>
```



飞风娱乐网 <http://www.flywind.org>