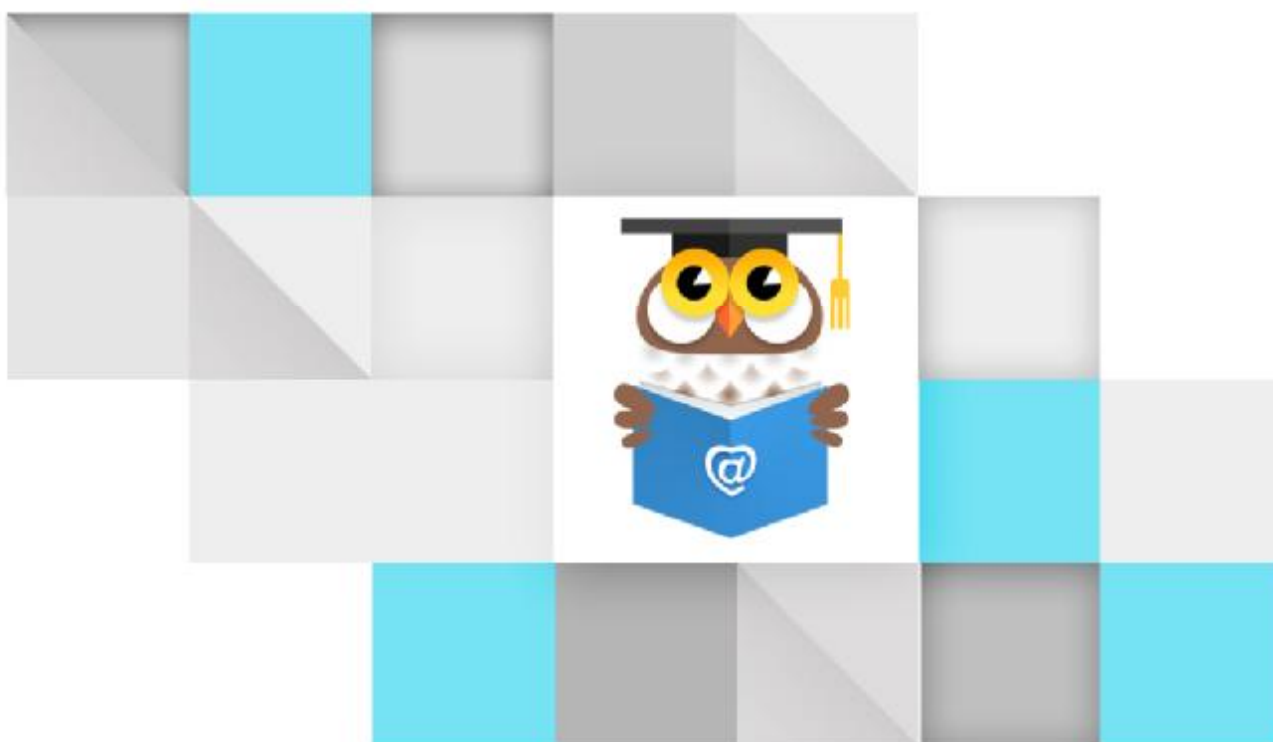


去哪儿玩旅游网（动态 WEB）

教学指导手册

PRJ-WTP-JEE-001 – 监控请求与响



Campus Solution Group

目 录

一、场景说明	1
1、实现效果	1
2、业务描述	1
3、实现思路	1
4、核心组件	2
二、实训技能	2
1、重点演练	2
2、相关技能	2
3、相关知识点	2
4、前置条件	2
5、搭建环境	3
三、场景任务	4
任务 1、创建 Tomcat 测试环境	4
任务 2、打包部署 Web 应用	6
任务 3、Tomcat 路径优化	8
场景总结	10

一、场景说明

1、实现效果

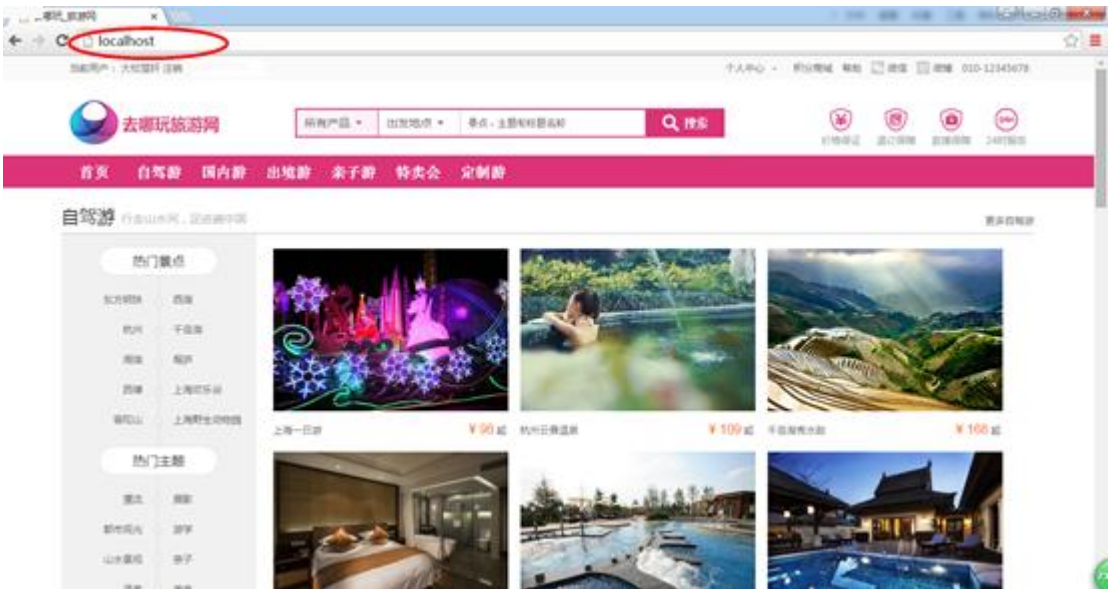


图 1-1-1

2、业务描述

本场景用于演练搭建基于Tomcat的Web应用运行环境和优化URL的技能。

- 2-1. 在Eclipse中创建Server工程，配置Tomcat的测试运行环境。
- 2-2. 将PRJ_WTP_JEE_001项目部署入测试环境中，运行《去哪儿旅游网》。
- 2-3. 打包《去哪儿旅游网》站点（war包），并将其部署入Tomcat真实运行环境中。
- 2-4. 确保通过<http://localhost>可以直接访问《去哪儿旅游网》。

3、实现思路

无。

4、核心组件

无。

二、实训技能

1、重点演练

- 1-1. 搭建基于Eclipse的Tomcat测试运行环境。
- 1-2. 在测试环境中部署、运行Web应用程序。
- 1-3. 实现Web应用的打包，并在真实环境中部署、运行。
- 1-4. 优化/简化访问Web应用的URL连接。

2、相关技能

- I Web 开发环境搭建

3、相关知识点

- I URL/URI
- I 安装并配置 Tomcat
- I 启动与停止 Tomcat
- I 打包部署 Web 应用
- I 优化访问

4、前置条件

4-1. 前置场景：无

4-2. 前置技能：

4-2.1. Java开发工具 (Eclipse for J2EE)。

5、搭建环境



三、场景任务

任务 1、创建 Tomcat 测试环境

1. 任务说明：

1-1. 完成效果：无

1-2. 任务目标：

1-2.1. 搭建Tomcat测试环境。

1-3. 任务要求：

1-3.1. 在Eclipse中创建Server工程（Tomcat测试环境）。

1-3.2. 配置Tomcat的测试运行环境。

1-3.3. 运行《去哪玩旅游网》项目。

2. 实现思路：

2-1. 在Eclipse中创建一个Tomcat类型的Server工程。

2-2. 将PRJ_WTP_JEE_001项目部署入Tomcat环境中运行。

2-3. 实现流程如下图：



图 3-1-1

3. 推荐步骤：

3-1. 在Eclipse中创建Server（Tomcat测试环境）：

3-1.1. 打开Eclipse，选择File->New->Others；

3-1.2. 选择Server文件 à Server工程，点击Next；

3-1.3. 选择Apache文件夹 -> Tomcat v7.0 Server；

3-1.4. Server's Host Name : localhost；

3-1.5. Server Name : Tomcat v7.0 Server at localhost；

3-1.6. 点击Finish。

3-2. 部署并运行《去哪玩旅游网》项目：

3-2.1. 右键选择项目工程PRJ_WTP_JEE_001，Run As->Run on Server；

3-2.2. 在Server选择对话框中，选择Tomcat v7.0 Server at localhost；

3-2.3. 点击Finish完成运行。

+ 提示

1) 始终使用Tomcat v7.0作为《去哪玩旅游网》项目的Server。

如果希望每次运行《去哪玩旅游网》项目时不再出现Server选择对话框，可在第一次出现Server选择对话框时，勾选对话框左下角的

“Always use this server when running this project” 的复选框。

4. 验证与测试：

4-1. 使用浏览器（Chrome）访问《去哪玩旅游网》项目：

4-1.1. 打开Chrome浏览器访问：

http://localhost:8080/PRJ_WTP_JEE_001

4-1.2. 观察是否可以正常访问《去哪玩旅游网》的首页。

4-2. 停止Tomcat服务：

4-2.1. 在Eclipse的底部找到Servers快速窗口。

4-2.2. 选择Tomcat v7.0 Server at localhost，右键 -> Stop。

+ 提示

1) 没有找到Server快速窗口？

如果在Eclipse下方没有找到Servers，可使用以下方法打开快速窗口：

点击菜单：Window-> Show View ->Others，选择Server文件夹 – Servers快速窗口。

任务 2、打包部署 Web 应用

1. 任务说明：

1-1. 完成效果：见场景完成效果。

1-2. 任务目标：

1-2.1. 打包Web应用并部署入Tomcat真实运行环境中。

1-3. 任务要求：

1-3.1. 打包PRJ_WTP_JEE_001Web项目（war包）。

1-3.2. 将WAR包部署入Tomcat真实运行环境中。

1-3.3. 运行Web项目查看结果。

2. 实现思路：

2-1. 利用Eclipse中的简易打包工具：WAR Export打包站点，

站点取名：PRJ_WTP_JEE_001.war

2-2. 将打包完毕的站点（war包）部署入Tomcat真实运行环境中。

Tomcat的真实运行环境目录：C:/dev/Tomcat7_WTP/webapps

2-3. 实现流程如下图：



图 3-2-1

3. 推荐步骤：

3-1. 打包项目工程：

3-1.1. 选择项目工程PRJ_WTP_JEE_001。

3-1.2. 右键->Export->Web文件夹->WAR file，点击Next；

3-1.3. 在WAR包输出对话框中，选择WAR的输出保存路径：

Destination->Browse，选择桌面点击保存（默认：PRJ_WTP_JEE_001.war）。

3-1.4. 点击Finish。

3-2. 部署WAR包：

3-2.1. 将导出的PRJ_WTP_JEE_001.war拷贝到Tomcat目录下的Webapps文件夹

Tomcat本地路径：C:\dev\Tomcat7_WTP

3-2.2. 启动并运行Tomcat，双击Tomcat/bin/startup.bat。

4. 验证与测试：

4-1. 使用浏览器（Chrome）访问《去哪玩旅游网》项目：

4-1.1. 打开Chrome浏览器访问：

http://localhost:8080/PRJ_WTP_JEE_001

4-1.2. 观察是否可以正常访问《去哪玩旅游网》的首页。

任务 3、Tomcat 路径优化

1. 任务描述：

1-1. 完成效果：见场景完成效果。

1-2. 任务目标：

1-2.1. 优化《去哪玩旅游网》访问连接（URL）。

1-2.2. 优化后的访问URL，类似于通过域名直接访问站点的形式。

1-3. 任务要求：

1-3.1. 通过域名或IP地址直接访问《去哪玩旅游网》。

1-3.2. 精简Web应用的URL链接，形式如下：

优化前：http://localhost:8080/PRJ_WTP_JEE_001

优化后：<http://localhost>

2. 实现思路：

2-1. 通过Tomcat的server.xml配置文件优化URL中的HTTP端口号。

2-2. 通过Tomcat的server.xml配置文件优化URL中的站点路径。

2-3. 实现流程如下图：



图 3-3-1

3. 推荐步骤：

3-1. 缺省端口号设置：

3-1.1. 文件定位：找到Tomcat目录/conf/server.xml；

Tomcat本地路径：**C:\dev\Tomcat7_WTP**

3-1.2. 找到Connector标签节，将该节点中的port属性从8080修改为80端口。

+ 提示：

- 1) 修改前需要通过Tomcat/bin/shutdown.bat关闭Tomcat。
- 2) Connector节点必须是有效节点，而不是注释块中的节点。

3-2. 缺省（默认）站点设置：

3-2.1. 将Tomcat/webapps路径下的PRJ_WTP_JEE_001.war重命名为ROOT.war；

3-2.2. 删除之前生成的PRJ_WTP_JEE_001文件夹。

3-2.3. 删除ROOT文件夹（如果有的话）。

3-3. 运行Tomcat：

3-3.1. 运行Tomcat/bin/startup.bat。

4. 验证与测试：

4-1. 使用浏览器（Chrome）访问《去哪玩旅游网》项目：

4-1.1. 打开Chrome浏览器访问：

<http://localhost>

4-1.2. 观察是否可以正常访问《去哪玩旅游网》的首页。

场景总结

Q1. 什么是Web容器（Server），请谈谈Web容器作用。

1. Web容器又被称为服务器容器，它是Web应用程序运行的基础环境，所有Web应用程序都需要基于容器才可以运行。

2. Web容器负责接收多个不同客户端的请求（例如：浏览器地址栏、站点超级链接、按钮点击、键盘回车等提交操作），并将请求交给站点中对应的请求处理组件，随后Web容器将处理组件的计算结果发送到客户端以响应客户的请求操作（Web容器的职责如下图）。

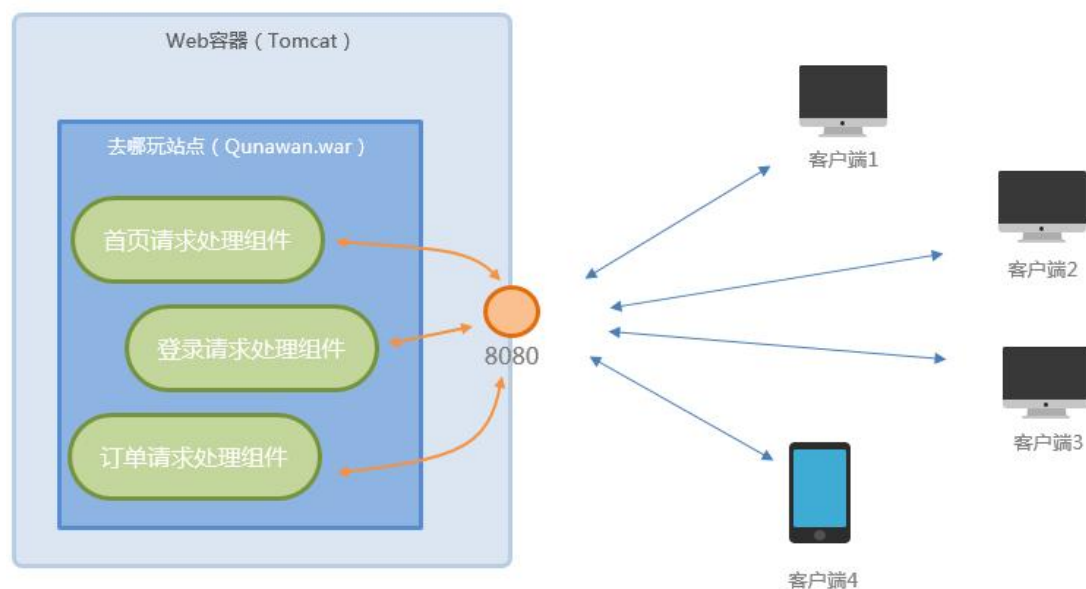


图 9-1

3. 利用Java语言开发的Web应用有大量的Web容器可供选择，常用的有：

Tomcat 普通的轻量级Web容器。

Jetty 性能优越扩展性更强的Web容器。

GlassFish Oracle公司标准轻量级Web容器。

JBOSS 支持分布式、高并发访问的重量级Web容器。

Q2. 如何打包Web应用？与Java应用的打包方式有何差别？

1. Web应用基础打包方式与Java应用的打包方式类同。
2. 区别在于Web应用打包完成后产生的是WAR包，而Java应用打包完成后产生的是JAR包。
3. WAR包必须基于Web容器才可以运行，JAR包仅需JVM运行环境即可独立运行。

Q3. 为何需要优化Web应用的访问连接（URL）？

1. 实际生产环境中，单个Web容器一般只部署一个核心Web产品（站点）。
2. 为保证用户可以通过域名直接访问Web容器中的Web产品（站点），我们需要将访问站点的URL链接优化成类似于：【协议://服务器IP地址】 这样的格式。
3. 经过优化的URL，用户可以直接通过域名访问到站点，而DNS（域名解析服务器）负责将域名解析成IP地址并找到Web服务器的所在位置，例如：

用户输入：<http://www.qunawan.com>

DNS将其解析为：<http://202.56.41.329>

4. 未经优化的URL，用户在访问时很可能会以如下方式输入URL：

<http://www.qunawan.com:8080/Qunawan/index.jsp>

以上的URL是任何一个用户都不可能记住的。

作者：Fox.Sun