# 帕累托图webAPP设计手册

## 逻辑部分

### 功能设计

根据帕累托图的功能要求，将其划分为以下四个功能模块：新建项目、项目管理、帕累托图绘制、报告编辑，具体各部分如图1‑1所示。

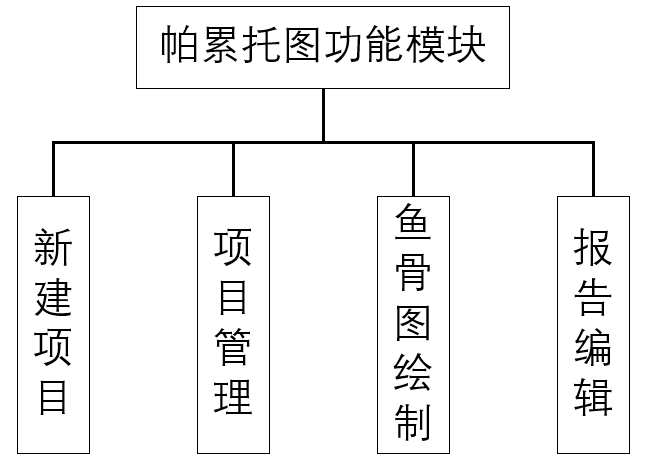


图1‑1 帕累托图功能模块划分

1. 新建项目

该部分主要功能是新建空的帕累托图项目。

1. 项目管理

该部分主要功能是管理和查看当前所有项目。

1. 帕累托图绘制

该部分主要功能是通过模板和表格交互绘制帕累托图。

1. 报告编辑

该部分主要功能是根据绘制帕累托图生成word报告。

帕累托图webAPP提供一种实时在线总结“二八定则”的操作工具，操作人员可以编辑自己的帕累托图项目以及项目报告，模板层项目组组长可以查看项目成员的帕累托图项目报告，帕累托图的用例图如图1‑2所示。

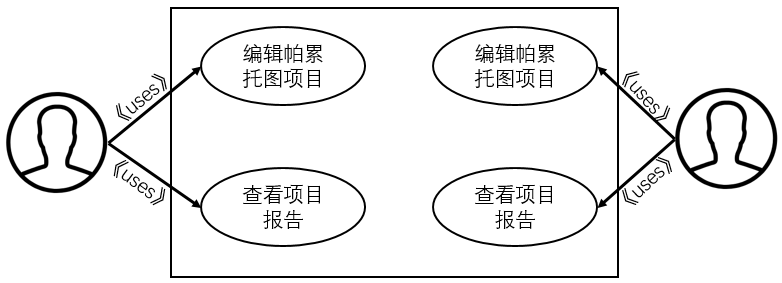


图1‑2 帕累托图用例图

### 流程设计

操作人员可以通过模板和在线表格交互绘图方式绘制帕累托图并生成报告，也可以通过点击项目来查看历史项目及其报告。帕累托图使用流程如图1‑3所示。

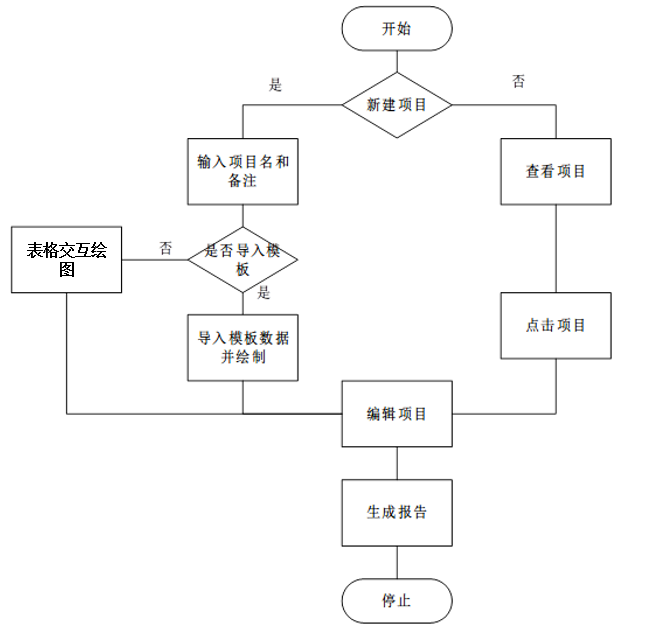


图1‑3 帕累托图webAPP流程图

### 数据库设计

帕累托图webAPP主要通过与后台的数据交互来实现对项目数据的保存。根据帕累托图的操作逻辑，帕累托图webAPP只有一张表，主要包括id、项目名、创建时间、用户名、备注、APP数据、域、word报告等，如表1‑1所示。

表1‑1 帕累托图数据表描述

|  |  |
| --- | --- |
| 表名 | 描述 |
| id | 项目ID |
| projectName | 项目名 |
| createDate | 项目创建时间 |
| username | 用户名 |
| memo | 项目备注 |
| appResult | App数据 |
| domain | 用户群组 |
| appContent | 项目报告 |

### 页面设计

帕累托图webAPP主要由侧边栏、主功能区以及报告编辑区组成，同时还包含项目管理（编辑、增加、删除操作）。

1. 新建项目点击按钮进入项目新建弹框，如图1‑4所示。



图1‑4 帕累托图新建项目页面

1. 主绘图区域，在主功能区可以看出分为操作区、表格区和帕累托图显示区，如图1‑5所示。

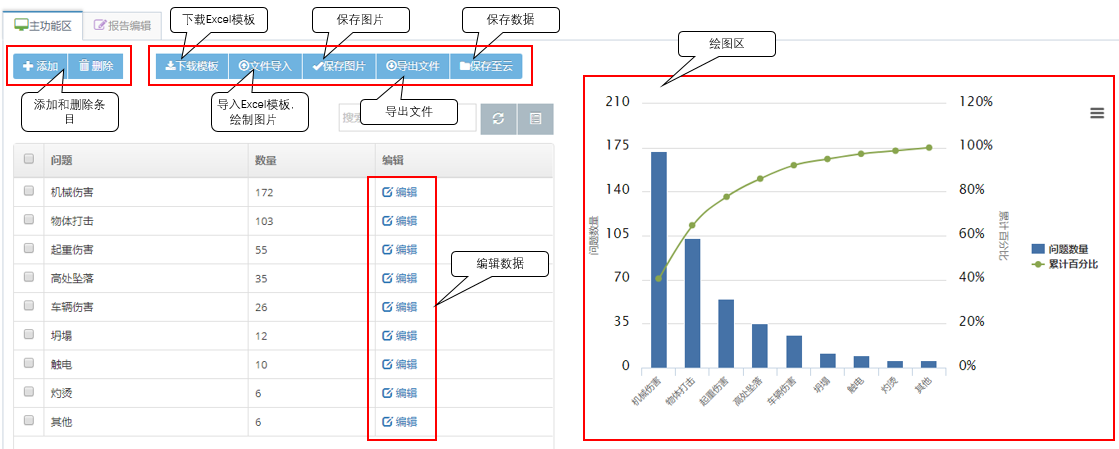


图1‑5 帕累托图主绘图区

1. 通过点击我的帕累托图或者图标跳转到项目管理页面，可以查看当前用户的的所有项目，并进行查看和删除操作，帕累托图的项目管理如图1‑7所示。

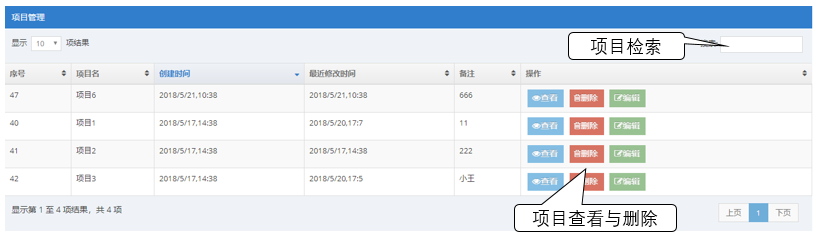


图1‑6 帕累托图项目管理

## 物理部分

帕累托图webAPP的前台文件如图1‑8所示，前台页面为index.jsp，文件的其他css和js文件都是动态修饰页面，以及帕累托图的算法实现。

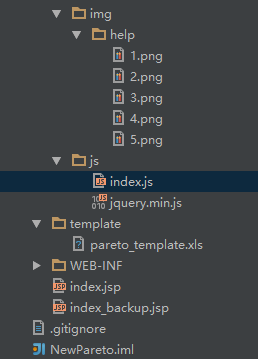
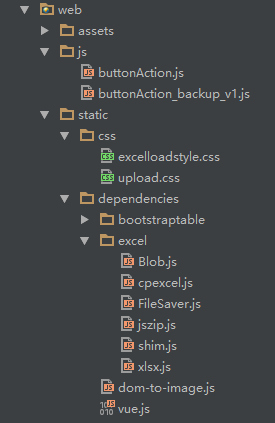


图1‑8 帕累托图的前台文件

前台相关函数如表1‑2所示。

表1‑2 帕累托图webAPP函数说明

|  |  |  |
| --- | --- | --- |
| 文件名 | 函数名 | 功能 |
| buttonAction.js | removeProject（） | 移除项目 |
| checkProject（index） | 查看项目 |
| saveProject（） | 保存项目 |
| saveAsProject（） | 另存项目 |
| getWordContext（） | 获取项目内容 |
| setCustomContext（） | 初始化word内容 |
| index.js | clearAllData（） | 清空所有数据 |
| drawPareto（） | 显示帕累托图 |
| mockExcelData（） | 模拟一份excel数据 |
| loadDataFromExcel（） | 从Excel中把数据加载到模型中 |
| displayDataInTable（） | 在表格中展示数据 |
| displayDataInPareto（） | 使用帕累托图展示数据 |
| getSerializableData（） | 将数据序列化成为字符串以便保存 |
| hideDisplayElements（） | 隐藏显示元素 |
| addElement（） | 添加元素 |
| reDisplay（） | 重新绘制表格与图像 |
| getIndexById（） | 根据Id找出index |
| delDataById（） | 根据id删除元素 |
| delElement（） | 删除元素 |
| delElements（） | 批量删除元素 |
| showAddElementPanel（） | 显示增加元素模态框 |
| showEditElementPanel（） | 显示编辑元素面板 |
| editElement（） | 编辑元素 |
| saveToFile（） | 保存至文件 |
| saveAsPicture（） | 保存为图片 |
| saveProject（） | 保存至云端 |
| saveAsProject（） | 另存项目 |

## 生产案例

1. 新建项目，如图1‑9所示。



图1‑9 新建帕累托图项目

1. 下载并导入模板，表格填写需要遵循一定规则，表格填写方式必须按照帕累托图树结构进行填写（非常重要），模板中每一行只能有一个数据，列数据必须相邻，否则将会导致无法绘制帕累托图，如图1‑10所示。

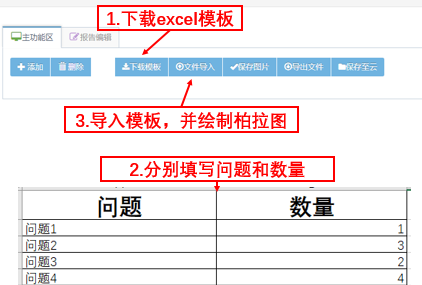


图1‑10 下载并导入模板

1. 编辑帕累托图，如图1‑11所示。



图1‑11 帕累托图表格编辑

1. 报告编辑，如图1‑12所示。



图1‑12 帕累托图报告编辑

最终通过输入问题名和对应数量，输出当前问题的帕累托图分析结果，绘制如图1‑13所示的帕累托图。

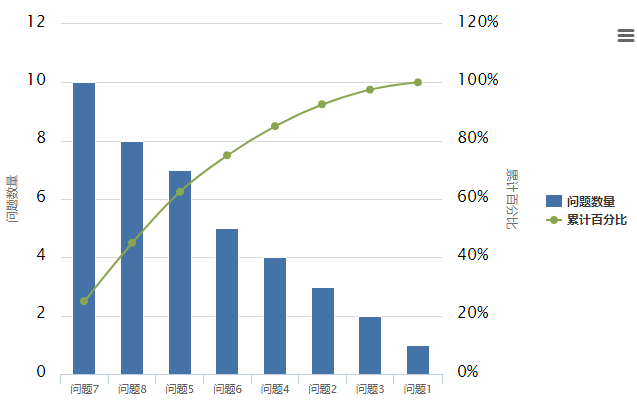


图1‑13 加工问题分析帕累托图

## 安装与调试

### jdk安装说明

1. 点击jdk-6u14-windows-i586.exe，直接安装
2. 右键点击计算机选择“属性”，找到“高级系统设置”，点击环境变量。在系统变量中兴建变量名：classpath；变量值为：

.;d:\%JAVA\_HOME%\lib\dt.jar;d:\%JAVA\_HOME%\lib\tools.jar;d:\%JAVA\_HOME%\jre\lib

1. 同样在系统变量中新建变量名JAVA\_HOME；变量值为jdk的安装目录，一般为C:\Program Files\Java\jdk1.6.0\_14。安装目录如图1‑14所示。

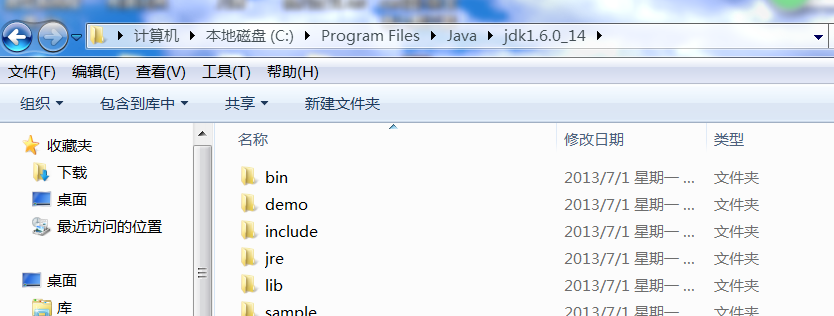


图1‑14 jdk安装目录

1. 同样在系统变量中新建变量名path，变量值为C:\Program Files\Java\jdk1.6.0\_14\bin。若已经存在path变量名，直接添加：;C:\Program Files\Java\jdk1.6.0\_14\bin。
2. 验证jdk是否安装成功。新建一个Test.txt文件，复制以下程序到Test.txt文件：

Public class Test{

public static void main{

System.out.println("successful");

}

}

将Test.txt文件保存为Test.java文件。然后打开命令提示符窗口，cd到你的Test.java所在目录，然后依次键入下面的命令并回车：

javac Test.java

java Test

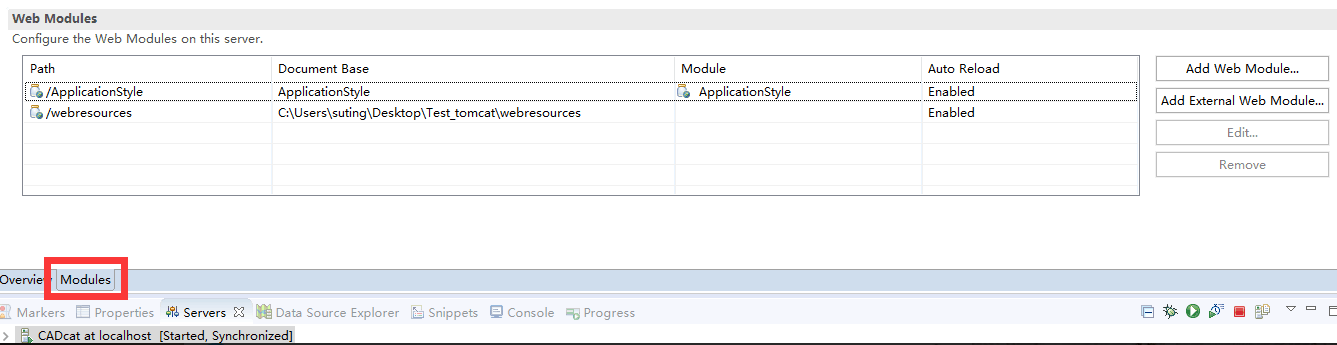
此时如果看到打印出来successful的话说明安装成功了，如果没有打印出这句话，你需要仔细检查一下你的配置情况。

### Tomcat和静态资源配置

首先，更新教研室专属Tomcat，以及添加webresources目录，操作细节参考庆贺的静态资源使用文档；

其次，将web文件夹下的资源复制到项目的WebContent（eclipse）下或者web（Idea）下；

然后，配置后台连接接口程序projectManager，使用方法和静态资源调用一样，在eclipse中，配置如图1‑15所示：



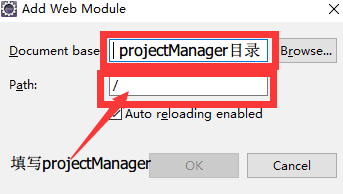
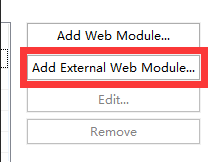
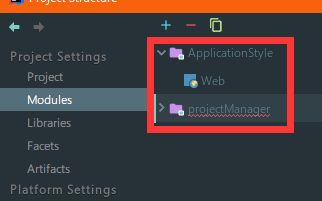
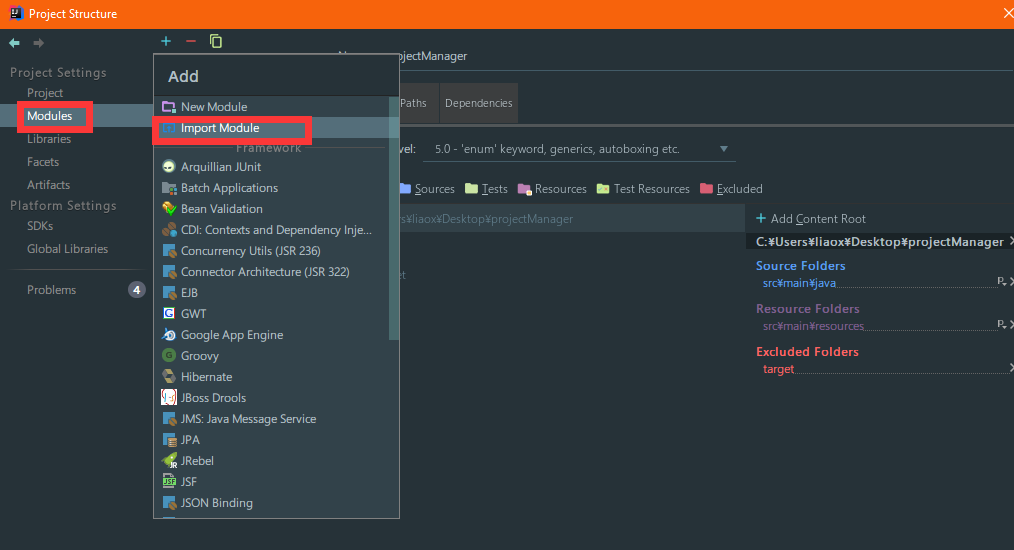
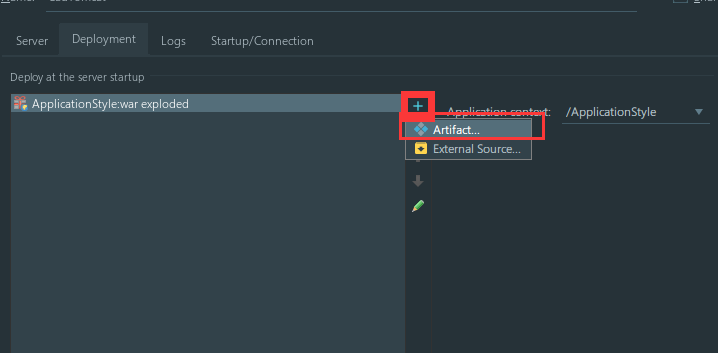


图1‑15 eclipse环境配置

Idea配置过程如图1‑16所示：





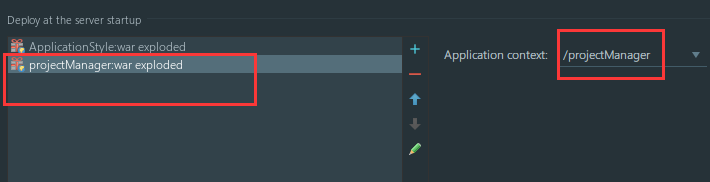


图1‑16 Idea环境配置

由此完成数据库接口配置；

最后，在index.html中，进行功能界面开发以及使用帮助文档完善；

在数据库中建表，并命名为demo（根据app名字确定），与buttonAction.js中appName保持一致。例如帕累托图的appName为fishbone，数据库对应的名称为fishbone\_project。

### 部署与运行

在idea中，勾选deploy application configured in tomcat instance，将webresources加载到项目中，如图1‑17所示；

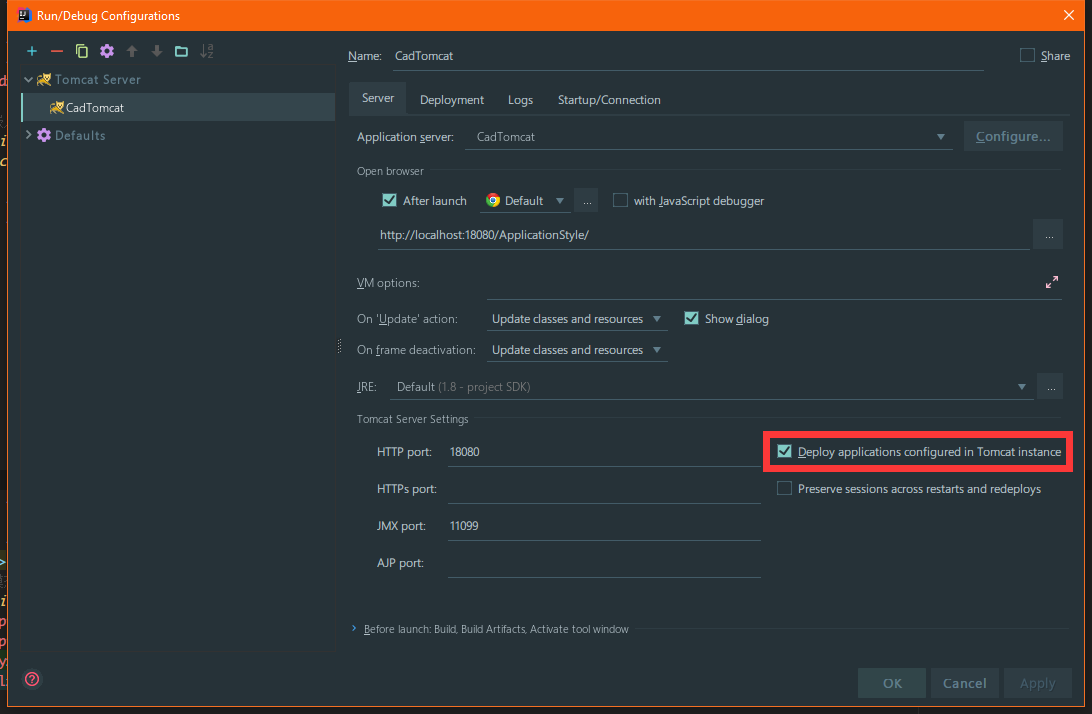
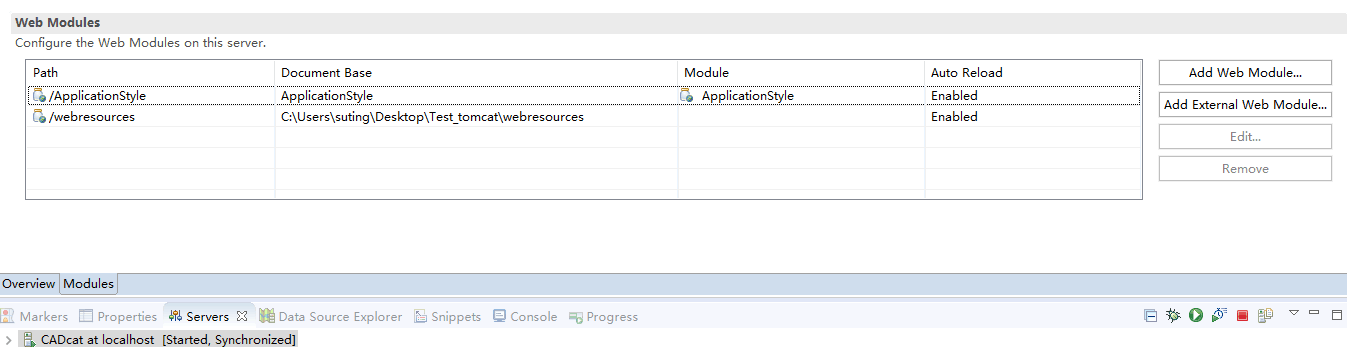


图1‑17 Idea中webresources配置

在eclipse中，编辑server，在modules中配置webresources，如图1‑18所示：



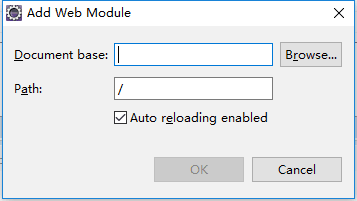
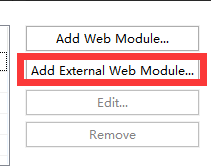


图1‑18 eclipse中webresources配置

选择教研室专属tomcat下的webresources目录，在path处填写/webresources,运行即可。

### 程序发布说明

1. 服务器端部署

在配置完成tomcat之后将fishbone.war复制到tomcat安装目录中的“webapps”文件夹中。如图1‑19所示：

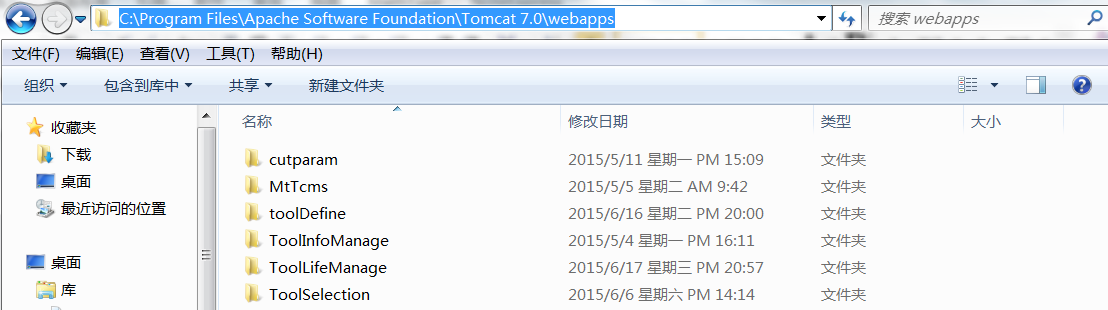


图1‑19 JavaWeb程序发布目录

2）远程tomcat管理页面部署

登录tomcat的管理页面58.206.100.146。页面如图1‑20所示。

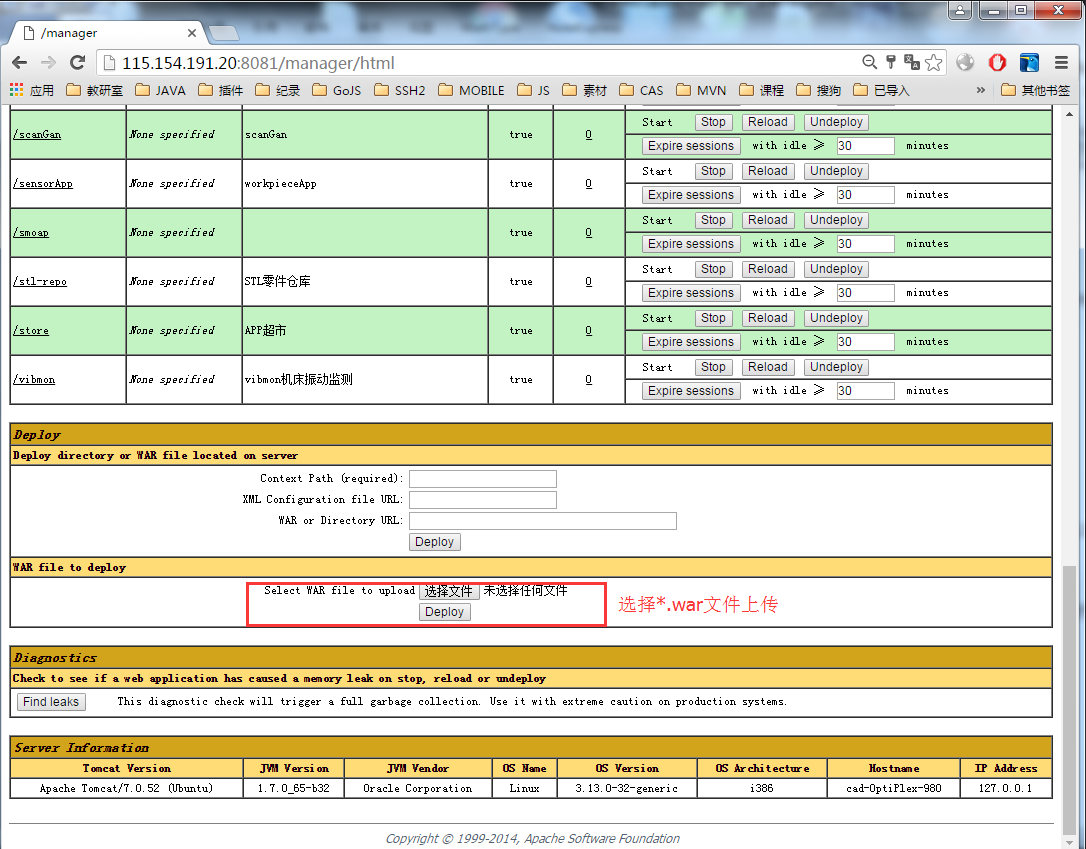


图1‑20 向服务器上传\*.war文件进行部署

选择需要部署到服务器端的\*.war文件，然后上传即可。

**注意：**程序发布一定是发布编译之后的程序，不能使源代码。

### Bug测试

系统常见的错误以及处理方法如表1‑3所示。

表1‑3 系统运行出错及处理措施

|  |  |
| --- | --- |
| 出错信息 | 补救措施 |
| 某些界面不能显示 | 重新开启平台软件界面 |
| 服务器端连接失败 | 检查服务器端口，并正确配置和启动 |
| 登录异常 | 检查改用户名和密码是否正确 |
| 部分界面不能显示 | 刷新或重新开启平台软件界面 |