西南民族大学

实验报告

2018 ------2019 学年第　2　学期

课程名称：Web 应用开发（Java）

学　　院：计算机科学与技术学院

年级：2016　　班级：计科 1602

学号：201631102013　　姓名：杜安娜

同组人：无

西南民族大学学生实验报告

教学单位：计科学院　　实验室名称：计算机应用实验室　　实验时间：2019 年 3 月 1 日

实验名称	第一个 Java Web 应用	实验编号	1
实验成绩		教师签名	
教师评阅：			

实验项目报告内容

1、 实验目的

A. 搭建 Java Web 项目开发环境
B. 开发第一个 Java Web 应用

2、 实验内容

2.1. 搭建 Java Web 开发环境

2.1.1. 配置环境变量

至少保证 JAVA_HOME 环境变量被正确设置。

2.1.2. 安装 Tomcat 服务器

可安装绿色版 Tomcat 服务器，将其解压到磁盘即可。有的实验室机房已将 Tomcat 绿色版安装到 C 盘根目录下，位于 C:\apache-tomcat-7.x.xx（若没有该文件夹则表示机房尚未安装，需要从 FTP 上面下载后解压安装）。C:\apache-tomcat-7.x.xx\bin 目录下为 Tomcat 的可执行文件，双击 startup.bat 批处理文件，即可启动 Tomcat。需要注意的是确保 2.1.1 设置正确。启动后会有一个常驻的命令行窗口（运行期间不会消失），如图 1 所示。

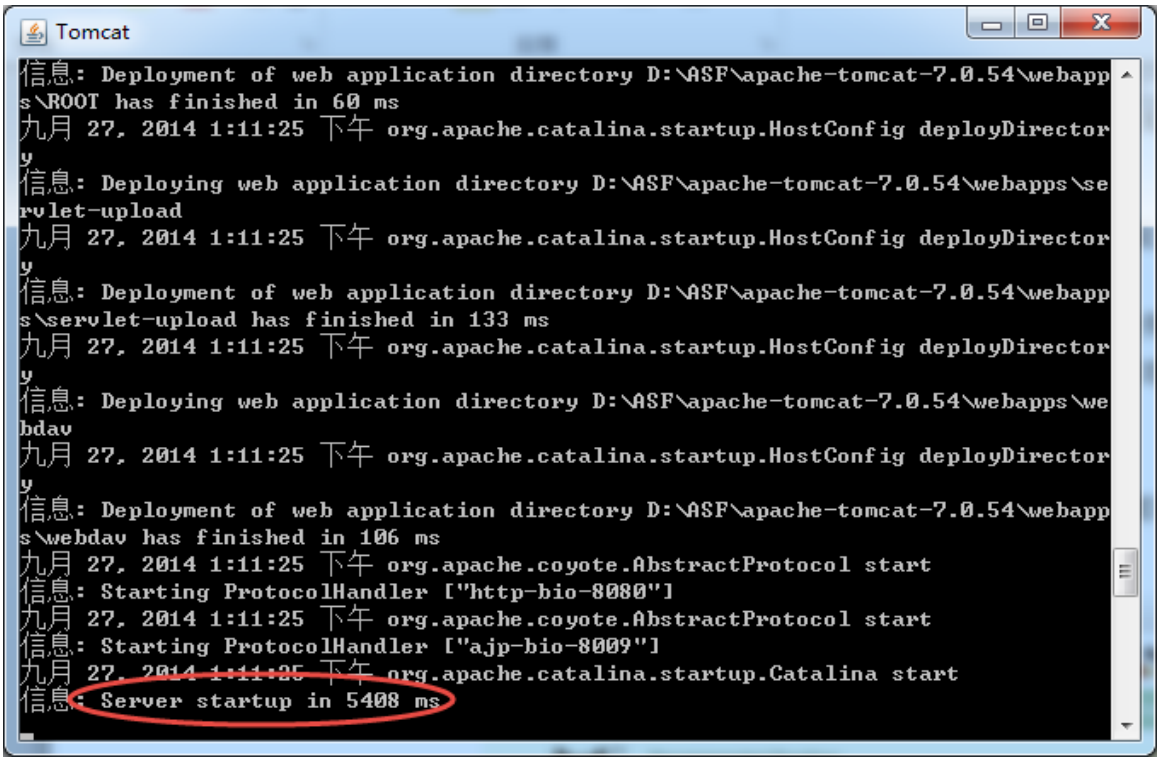


图 1

开启浏览器，键入 http://localhost:8080/，出现如图 2 界面即代表启动成功。

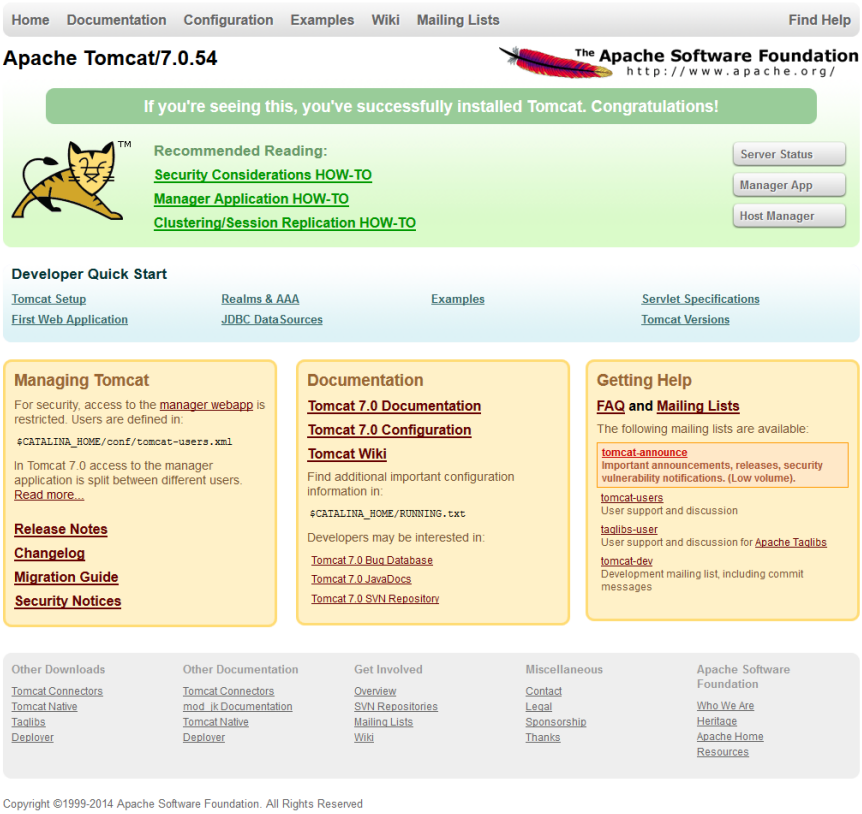


图 2

确保 Tomcat 能正确运行之后，我们将所示的命令行窗口关闭，解除 Tomcat 对 8080 端口的占用。本步骤只是为了保证 Tomcat 被正确安装，下面介绍更高效的方法。

2.1.3. 使用 Eclipse 连接 Tomcat

在进行项目开发时，一般直接利用 Eclipse 启动 Tomcat，这样调试更方便，也更高效。

A. 配置服务器运行时环境

首先需要配置服务器运行时环境，即在 Eclipse 中指向 Tomcat 的安装目录。配置如图 3-图 6 所示，其中图 6 需要使用 Browse 指向自己机器 Tomcat 所安装的位置，即 C:\apache-tomcat-7.x.xx，而不一定是图中蓝色方框显示目录。

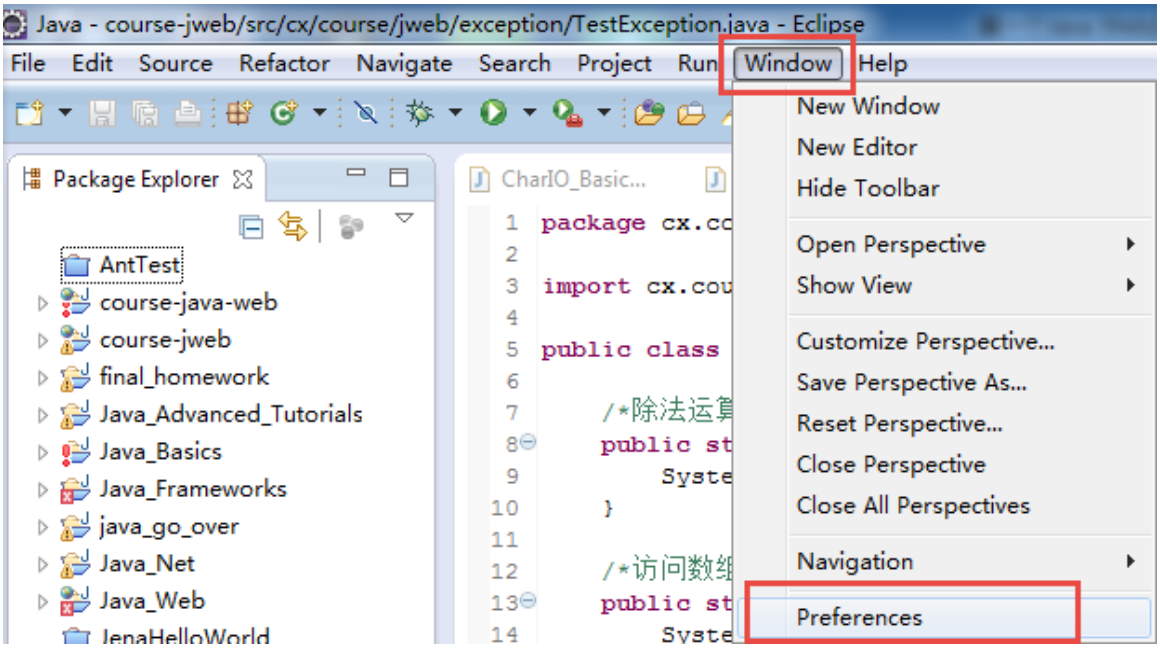

Java - course-jweb/src/cx/course/jweb/exception/TestException.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
New Window
New Editor
Hide Toolbar
Open Perspective
Show View
Customize Perspective...
Save Perspective As...
Reset Perspective...
Close Perspective
Close All Perspectives
Navigation
Preferences
Package Explorer
AntTest
course-java-web
course-jweb
final_homework
Java_Advanced_Tutorials
Java_Basics
Java_Frameworks
java_go_over
Java_Net
Java_Web
JenaHelloWorld

图 3

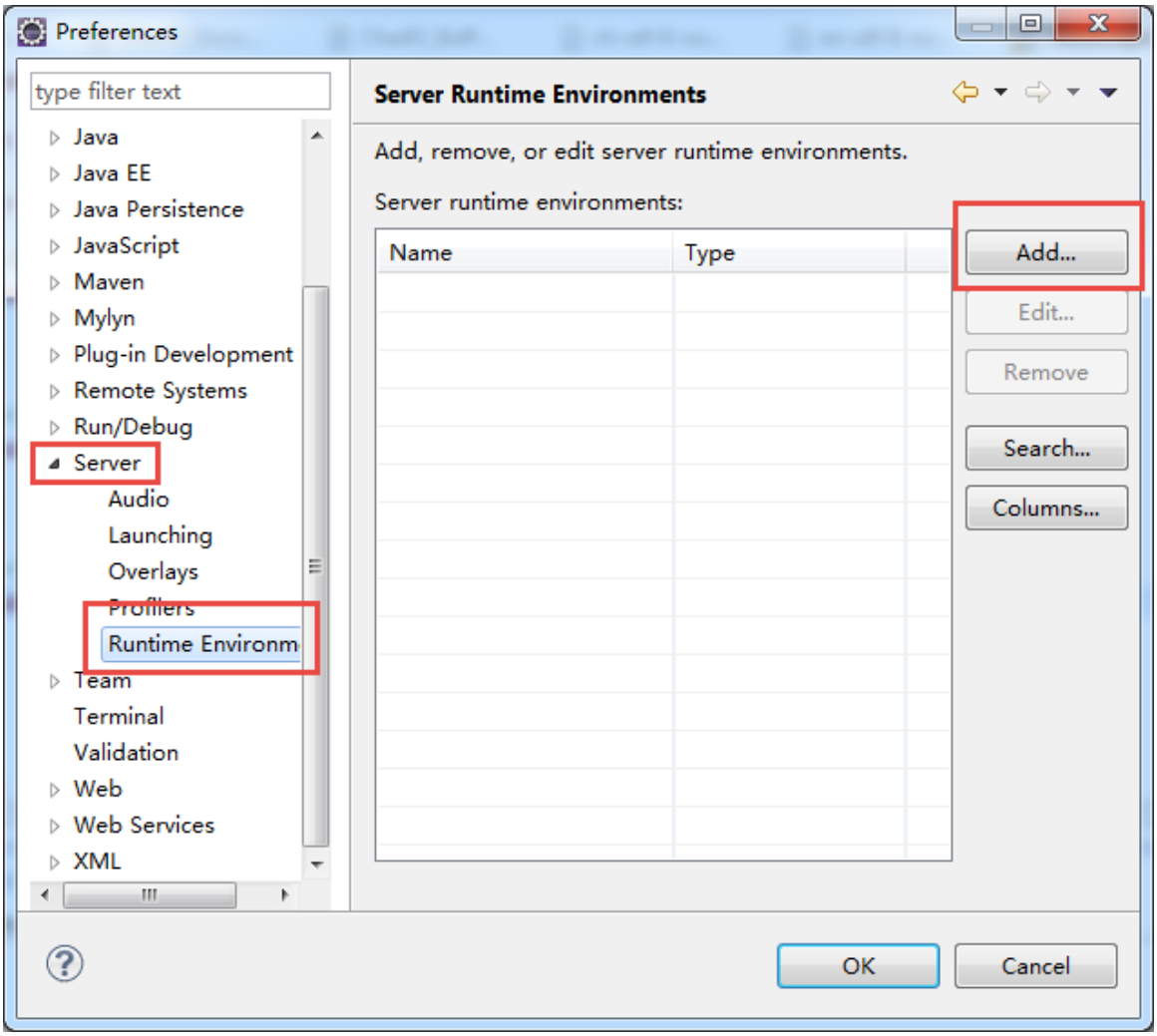

Preferences
type filter text
Java
Java EE
Java Persistence
JavaScript
Maven
Mylyn
Plug-in Development
Remote Systems
Run/Debug
Server
Audio
Launching
Overlays
Profilers
Runtime Environm
Team
Terminal
Validation
Web
Web Services
XML
Server Runtime Environments
Add, remove, or edit server runtime environments.
Server runtime environments:
Name
Type
Add...
Edit...
Remove
Search...
Columns...
OK
Cancel

图 4

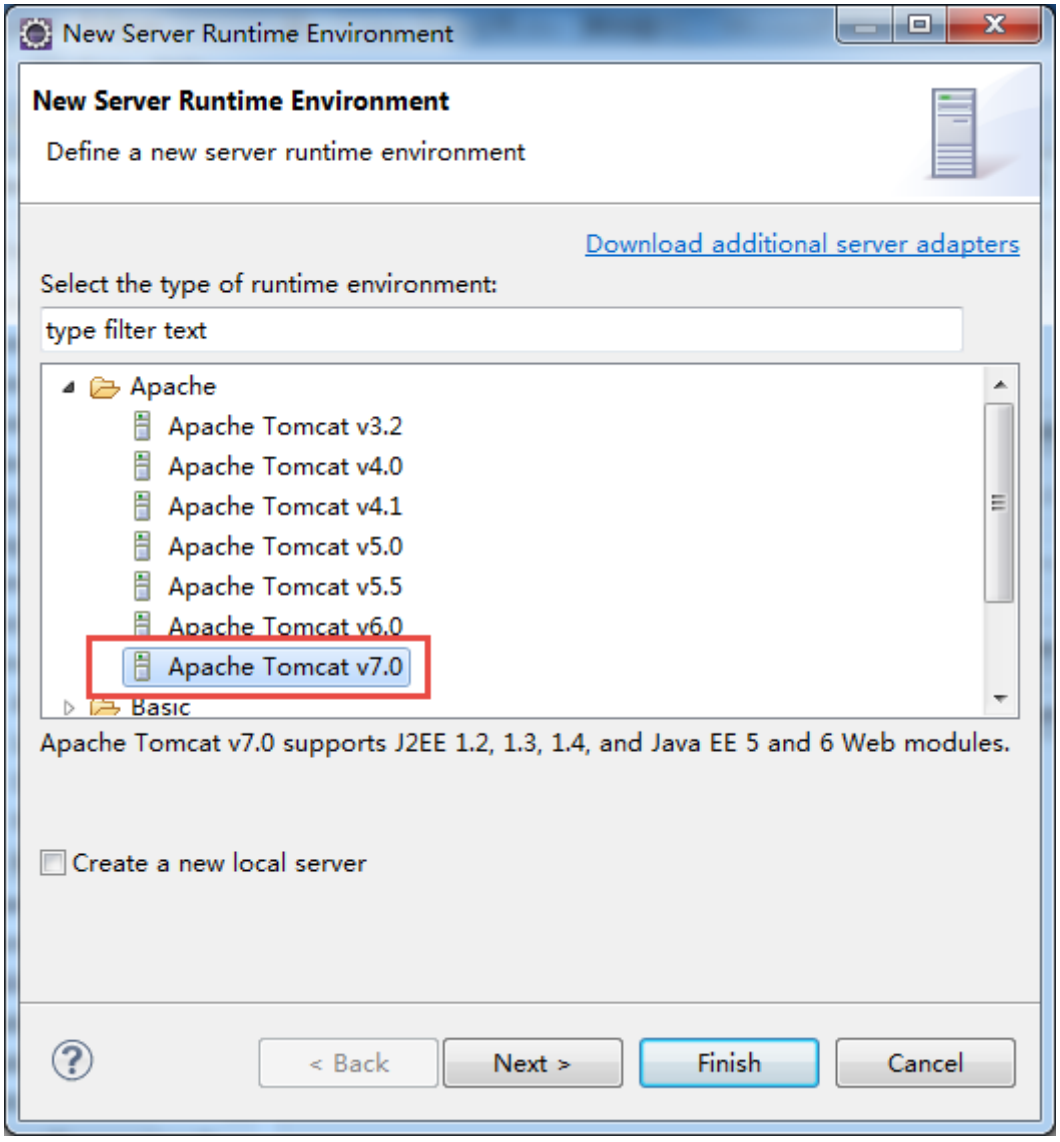

New Server Runtime Environment
New Server Runtime Environment
Define a new server runtime environment
Download additional server adapters
Select the type of runtime environment:
type filter text
Apache
Apache Tomcat v3.2
Apache Tomcat v4.0
Apache Tomcat v4.1
Apache Tomcat v5.0
Apache Tomcat v5.5
Apache Tomcat v6.0
Apache Tomcat v7.0
Basic
Apache Tomcat v7.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5 and 6 Web modules.
Create a new local server
< Back
Next >
Finish
Cancel

图 5

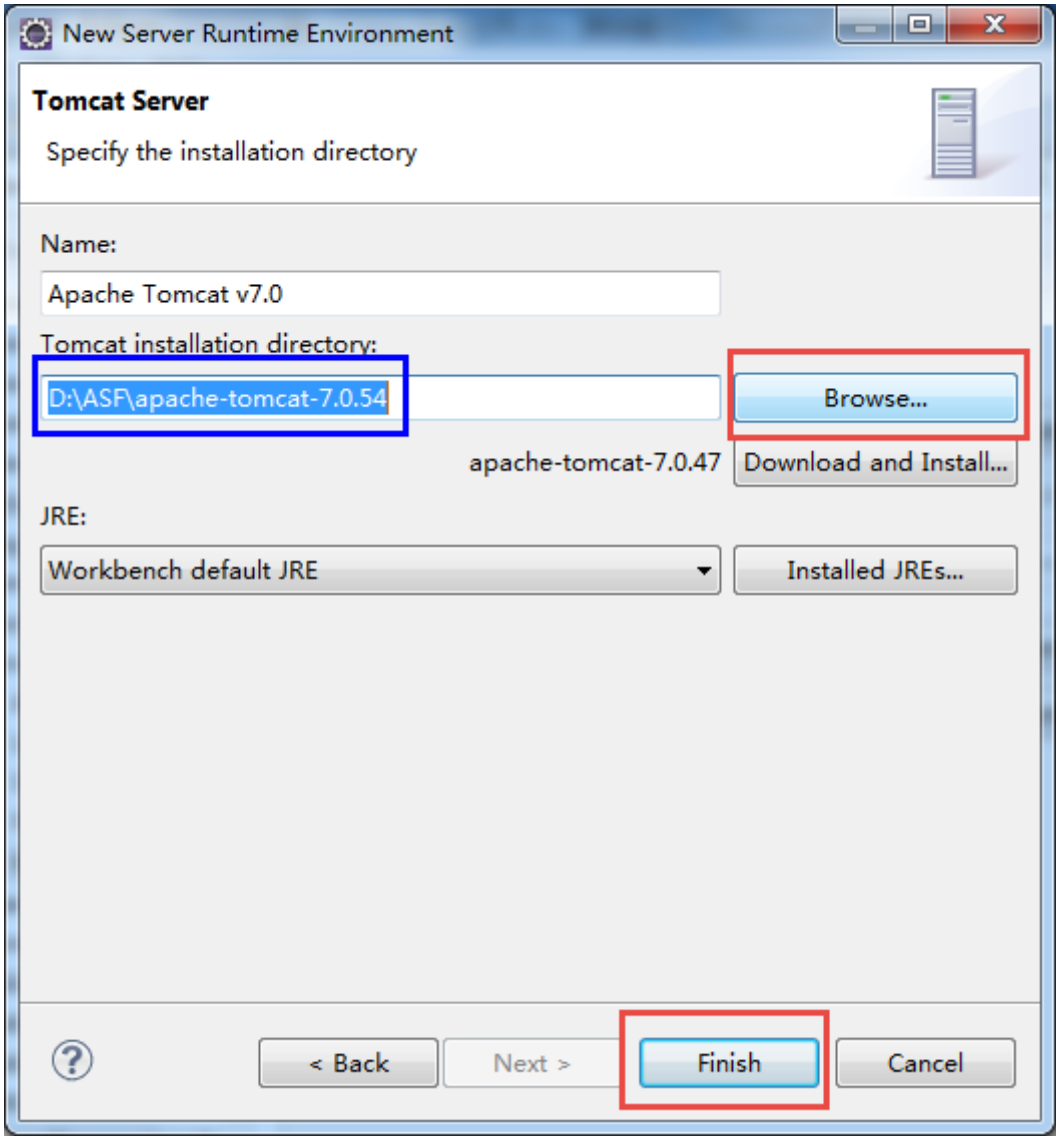
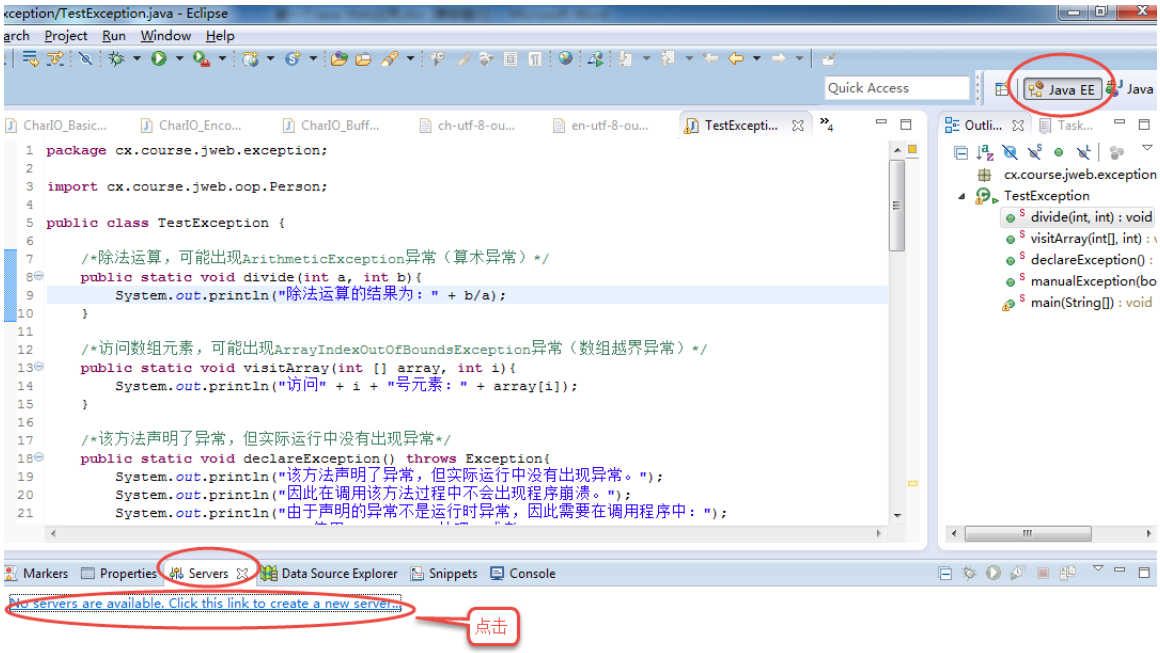


图 6

B. 配置服务器

本步骤将新建一个 Tomcat 服务器实例，配置如图 7-图 9 所示。一定注意图 9 中的设置顺序。




```
1  package cx.course.jweb.exception;
2
3  import cx.course.jweb.oop.Person;
4
5  public class TestException {
6
7      /*除法运算，可能出现ArithmeticException异常（算术异常）*/
8      public static void divide(int a, int b){
9          System.out.println("除法运算的结果为：" + b/a);
10     }
11
12     /*访问数组元素，可能出现ArrayIndexOutOfBoundsException异常（数组越界异常）*/
13     public static void visitArray(int [] array, int i){
14         System.out.println("访问" + i + "号元素：" + array[i]);
15     }
16
17     /*该方法声明了异常，但实际运行中没有出现异常*/
18     public static void declareException() throws Exception{
19         System.out.println("该方法声明了异常，但实际运行中没有出现异常。");
20         System.out.println("因此在调用该方法过程中不会出现程序崩溃。");
21         System.out.println("由于声明的异常不是运行时异常，因此需要在调用程序中：");
```

图 7

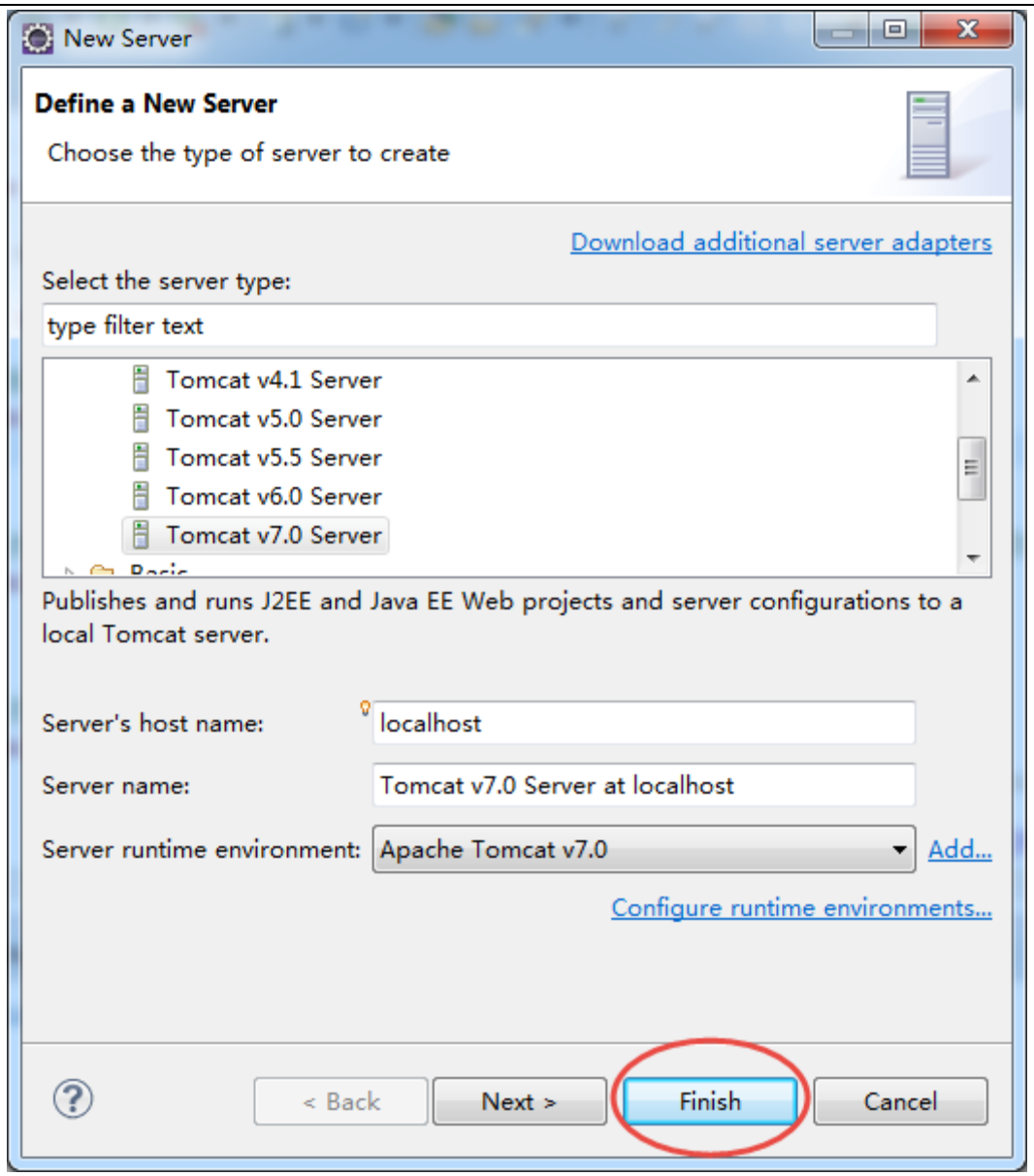

New Server
Define a New Server
Choose the type of server to create
Download additional server adapters
Select the server type:
type filter text
Tomcat v4.1 Server
Tomcat v5.0 Server
Tomcat v5.5 Server
Tomcat v6.0 Server
Tomcat v7.0 Server
Publishes and runs J2EE and Java EE Web projects and server configurations to a local Tomcat server.
Server's host name:
localhost
Server name:
Tomcat v7.0 Server at localhost
Server runtime environment:
Apache Tomcat v7.0
Add...
Configure runtime environments...
< Back
Next >
Finish
Cancel

图 8

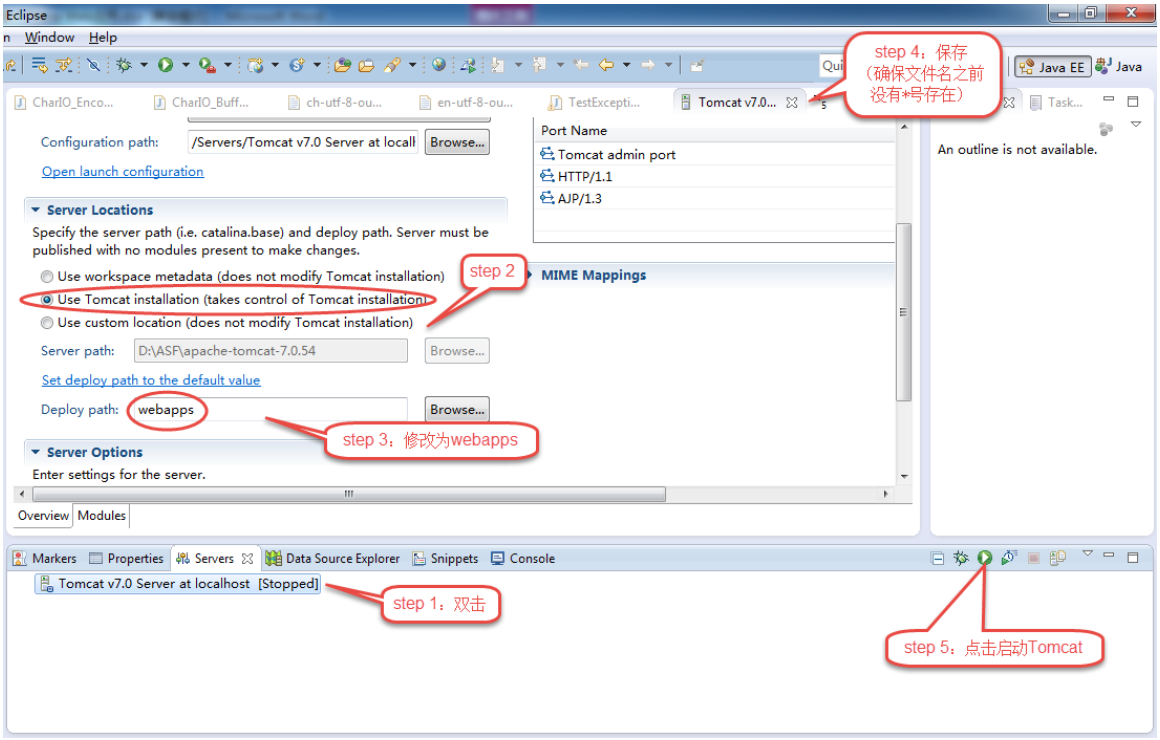


图 9

开启浏览器，键入 http://localhost:8080/，出现如图 2 界面即代表启动成功。

2.2. 开发第一个 Java Web 应用

本节的若干步骤实际上是一般 Java Web 应用开发的通用步骤，请对这些步骤有所认知。

2.2.1. 构建项目

请将 Eclipse 切换到 Java EE 透视图（图 7）。构建一个 Dynamic Web Project，如图 10-图 12 所示，其中“下一步”过程中，有一步需做如图 12 的操作，即让 Eclipse 自动生成 web.xml 文件。

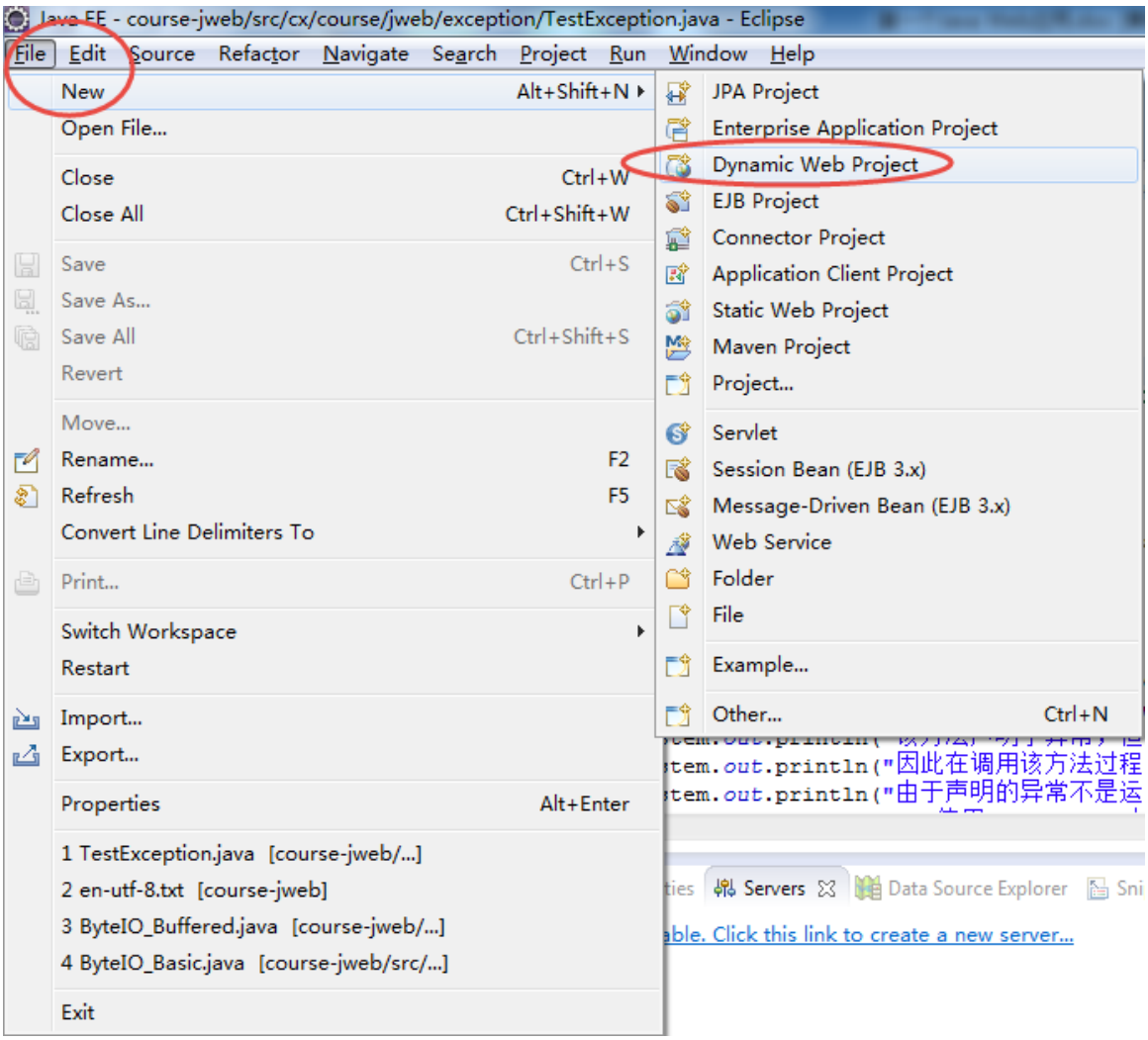

Java EE - course-jweb/src/cx/course/jweb/exception/TestException.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
New Alt+Shift+N
Open File...
Close Ctrl+W
Close All Ctrl+Shift+W
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... Ctrl+P
Switch Workspace
Restart
Import...
Export...
Properties Alt+Enter
1 TestException.java [course-jweb/...]
2 en-utf-8.txt [course-jweb]
3 ByteIO_Buffered.java [course-jweb/...]
4 ByteIO_Basic.java [course-jweb/src/...]
Exit
JPA Project
Enterprise Application Project
Dynamic Web Project
EJB Project
Connector Project
Application Client Project
Static Web Project
Maven Project
Project...
Servlet
Session Bean (EJB 3.x)
Message-Driven Bean (EJB 3.x)
Web Service
Folder
File
Example...
Other... Ctrl+N
tem.out.println("因此在调用该方法过程
tem.out.println("由于声明的异常不是运
Servers
Data Source Explorer
able. Click this link to create a new server...

图 10

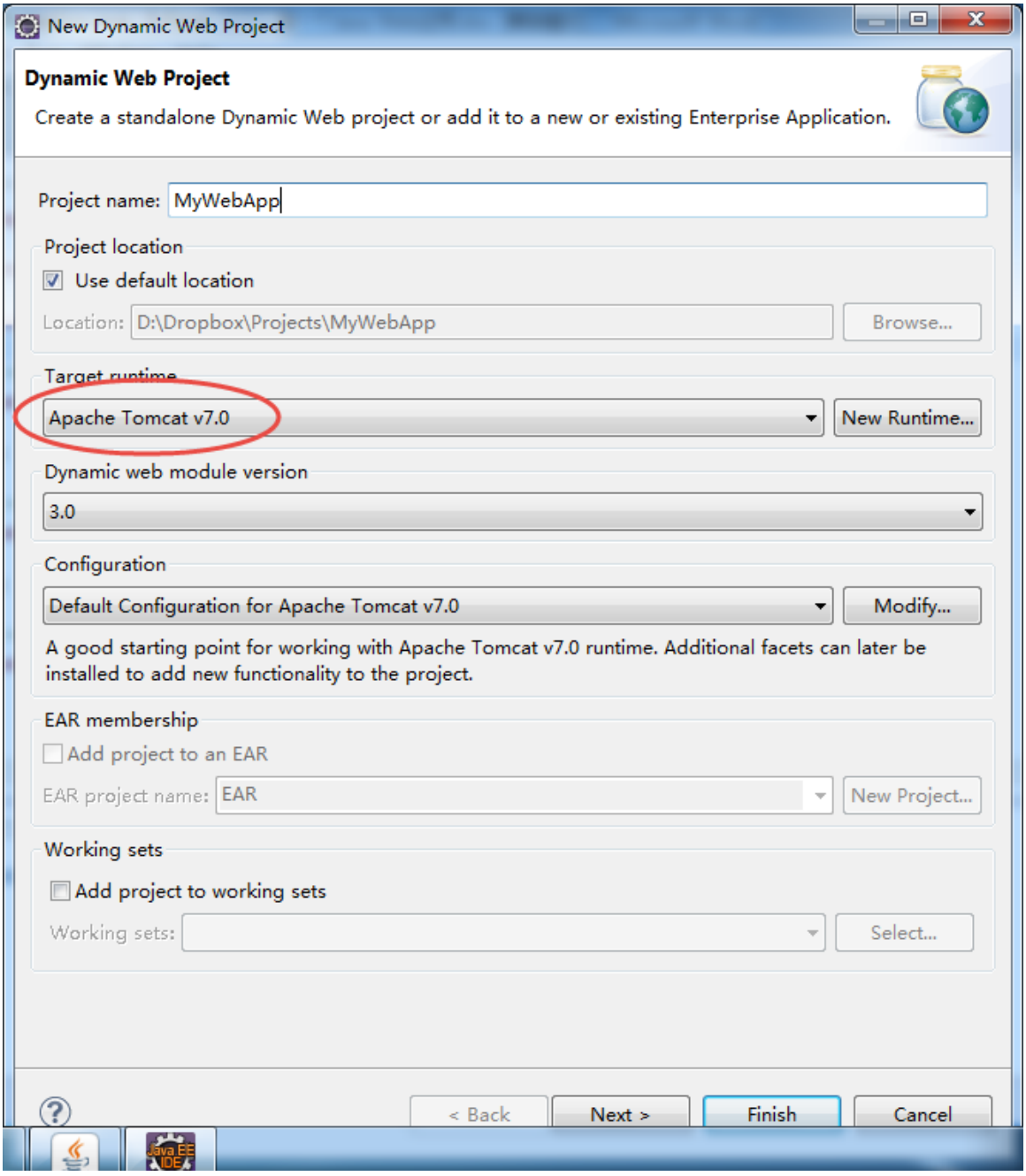

New Dynamic Web Project
Dynamic Web Project
Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.
Project name: MyWebApp
Project location
Use default location
Location: D:\Dropbox\Projects\MyWebApp
Browse...
Target runtime
Apache Tomcat v7.0
New Runtime...
Dynamic web module version
3.0
Configuration
Default Configuration for Apache Tomcat v7.0
Modify...
A good starting point for working with Apache Tomcat v7.0 runtime. Additional facets can later be installed to add new functionality to the project.
EAR membership
Add project to an EAR
EAR project name: EAR
New Project...
Working sets
Add project to working sets
Working sets:
Select...
< Back
Next >
Finish
Cancel

图 11

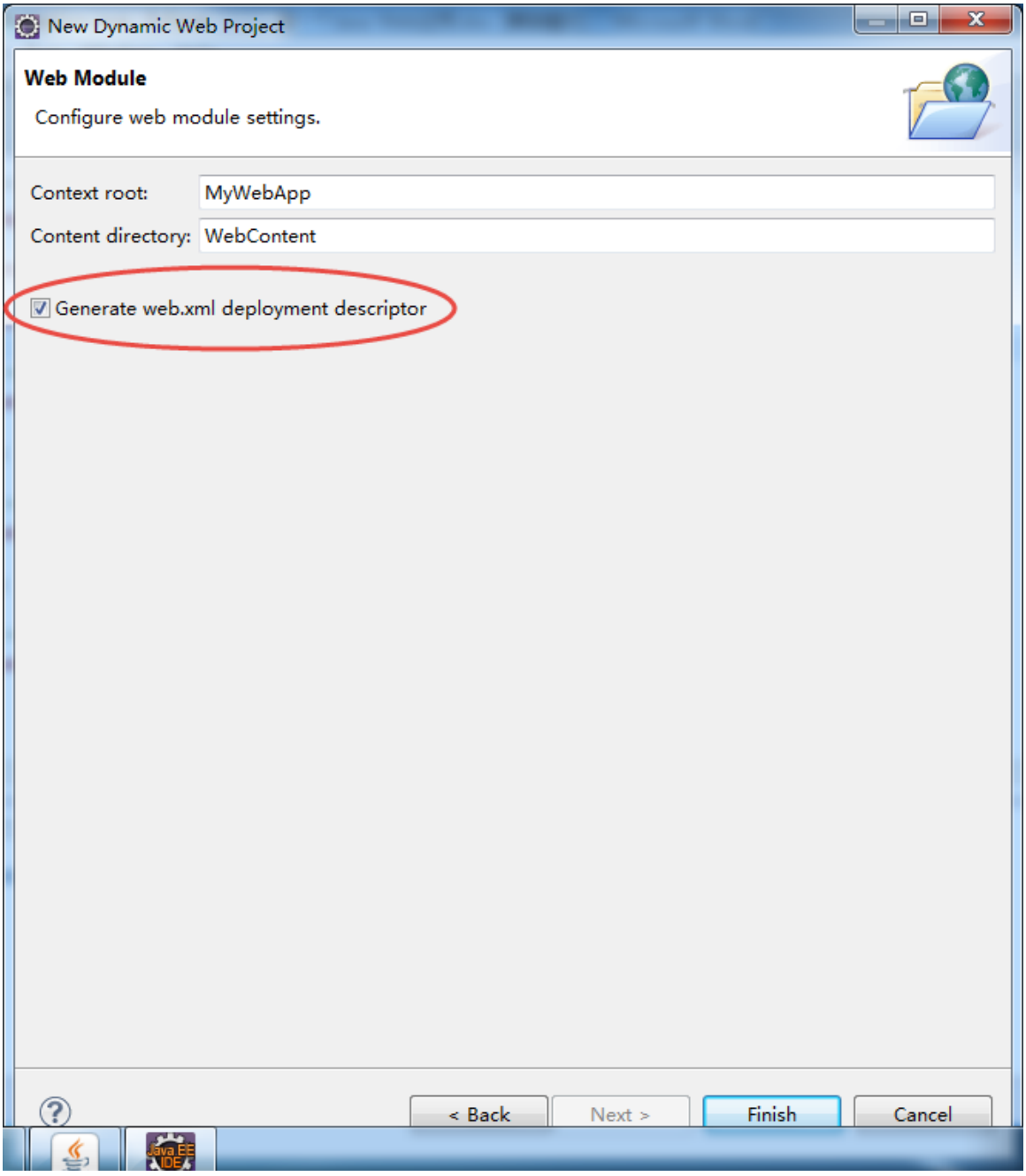


图 12

项目构建完毕后，文件夹结构如图 13 所示。其中 Part 1 用于存放 Java 源代码；Part 2 的 WebContent 文件夹用于存放 Web 应用的内容（包括前端页面、后端.class 文件、配置文件、需要的类库等等）。Part 2 中，web.xml 文件是整个 Web 应用的配置文件，用于指定特定的 URL 应该由哪些 Servlet 负责处理、Servlet 的实现类等关键配置信息。lib 文件夹用于存放需要使用的第三方库，在后续章节中将有讲解。

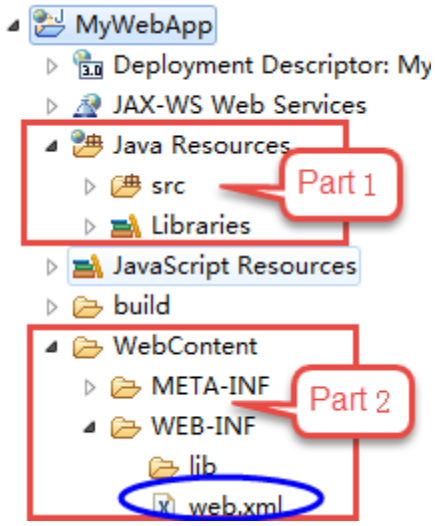


图 13

2.2.2. 部署应用

在开发过程中，可先将 Web 应用部署到 Tomcat 服务器上去，然后逐渐做修改测试。具体做法如图 14-图 16 所示。一般而言，由于我们的应用被部署在本地主机的 Tomcat 服务器上，本地主机的主机名为 localhost，因此应用被部署在 http://localhost:8080/[your_web_app_name]，其中[your_web_app_name]为自己的 Web 应用的名称（一般就是自己构建的项目的名称）。如本项目的名称为 MyWebApp，则应该为 http://localhost:8080/MyWebApp。

而实际上部署的内容就是 WebContent 的内容，WebContent 目录实际上是 Web 应用所包含内容的根目录，该目录的内容即是 Web 应用的内容，故名 WebContent。当 Web 应用被部署到服务器上去之后，该目录被修改为 Web 项目的名称，并被复制到 C:\apache-tomcat-7.x.xx\webapps 目录下。webapps 目录下就是 Tomcat 服务器已经部署的应用。

如图 16 所示，展开“Tomcat v7.0 Server at localhost”，若已有 MyWebApp 项目，说明部署成功。对应地，在 C:\apache-tomcat-7.x.xx\webapps 应该存在一个 MyWebApp 文件夹。在实际开发中，我们可以在本步骤之后进行 2.2.7 步骤，以测试应用是否不是成功。

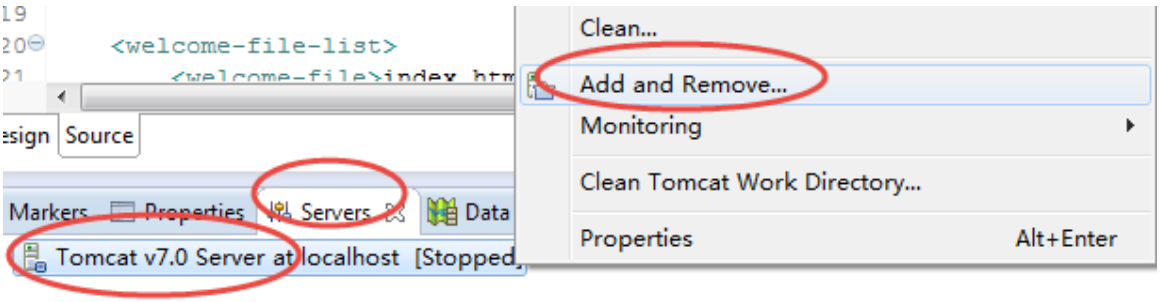


图 14

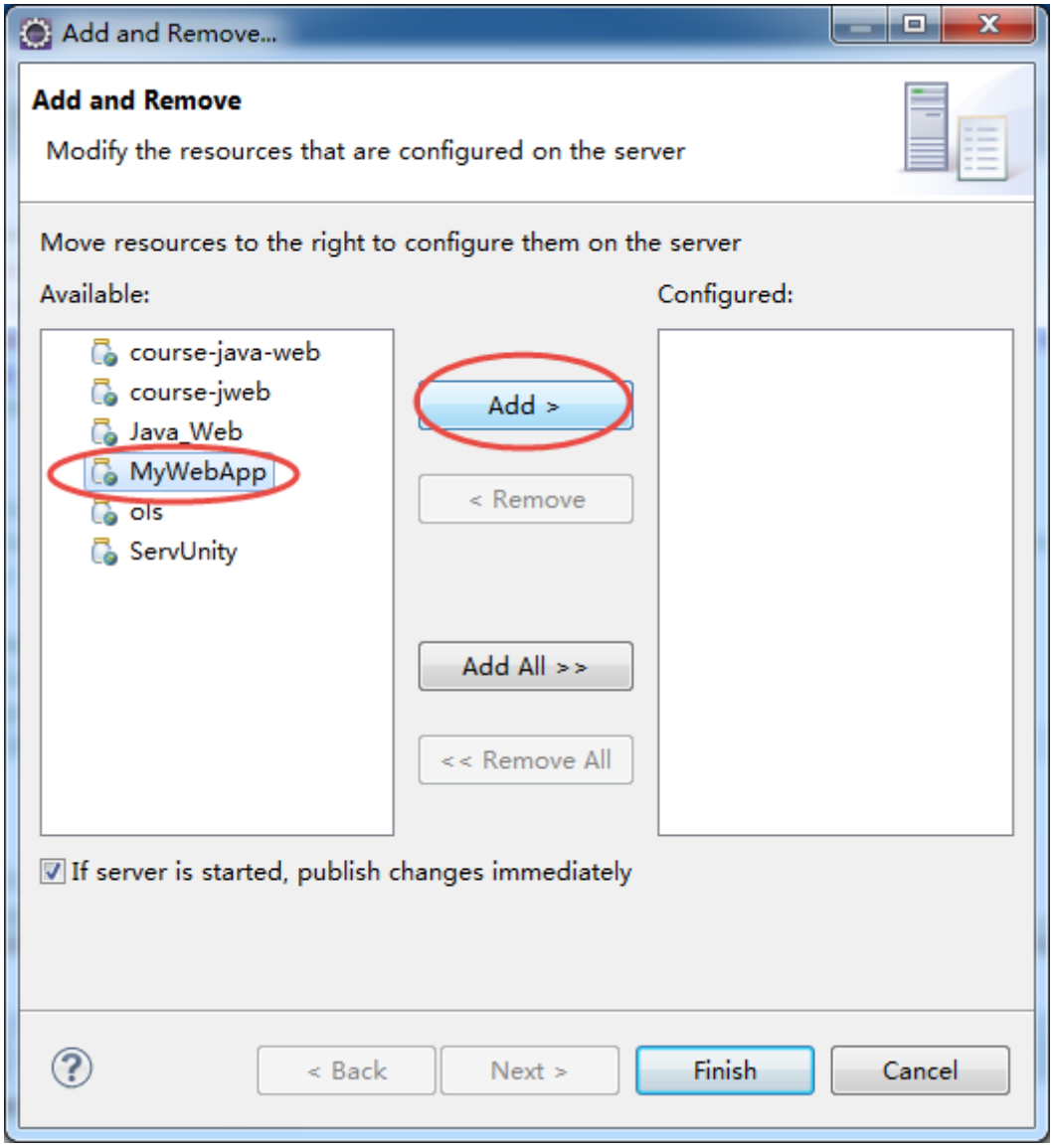


图 15

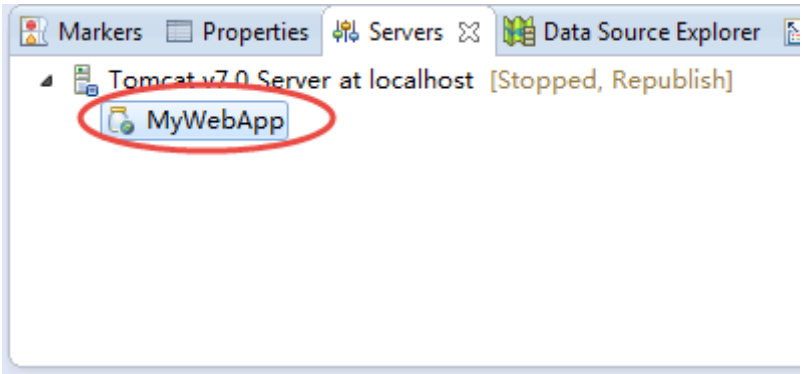


图 16

2.2.3. 编写 Servlet 代码（控制层）

Servlet 是 Java Web 应用对用户请求进行分发的组件，特定的 Servlet 处理特定的 URL 请求，并指定特定的后端 Java 业务逻辑为其服务（即调用其他的 Java 类、接口等组件进行具体的业务操作，如进行数据库的

操作、IO 操作等），最后指定特定的页面展示执行结果。在 Part 1 的 src 文件夹新建一个类（也可在 src 下先新建一个包，将代码按功能分门别类地存放，不做强制要求。程序清单 1 中将该 Servlet 放在了 cx.course.jweb.servlet 包内），命名为 HelloWorldServlet，源代码如下，注意看一看注释中给出的解释，加深理解。

程序清单 1

```
package cx.course.jweb.servlet;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class HelloWorldServlet extends HttpServlet {

    /*doGet用于处理HTTP GET请求，如一般的超链接、指定为GET方法的HTML表单等*/
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse
resp)
            throws ServletException, IOException {
        /*将对应的GET请求转发到jsp/hello.jsp页面
         * 至于GET请求所对应的URL路径在web.xml文件中设定
         * */
        req.getRequestDispatcher("jsp/hello.jsp").forward(req, resp);
    }

    /*doPost用于处理HTTP POST请求，如指定为POST方法的HTML表单等*/
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse
resp)
            throws ServletException, IOException {
        /*doPost方法的业务逻辑和doGet方法相同，直接调用即可*/
        this.doGet(req, resp);
    }
}
```

2.2.4. 编写业务逻辑代码

在 2.2.3 中已经谈到 Servlet 用于响应用户的 URL 请求。较好的设计风格是将具体的业务逻辑代码编写在专门的 Java 类、接口当中，Servlet 在响应用户 URL 请求的过程中，调用具体的 Java 类、接口等。这些 Java 类、接口实现了具体的业务逻辑，如与数据库相关的操作、IO 操作、排序查找等。本实验较简单，尚不涉及复杂的业务逻辑，因此此步内容为空。之所以列出此步骤，主要是为了体现较完整的开发流程和顺序。

2.2.5. 编写 JSP 页面（展示层、前端页面）

在 WebContent 目录下新建 index.jsp 页面（图 17），index.jsp 的代码如图 18 所示。WebContent 下的 index.jsp 文件可被自动识别为 Web 应用的主页。index.jsp 包含一个链接指向 hello，hello 是一个相对路径，根据其所在的位置可知，实际上是一个指向 http://localhost:8080/MyWebApp/hello 的链接，该链接由上述 HelloWorldServlet 处理（在 2.2.6 的 web.xml 文件中将做配置）。

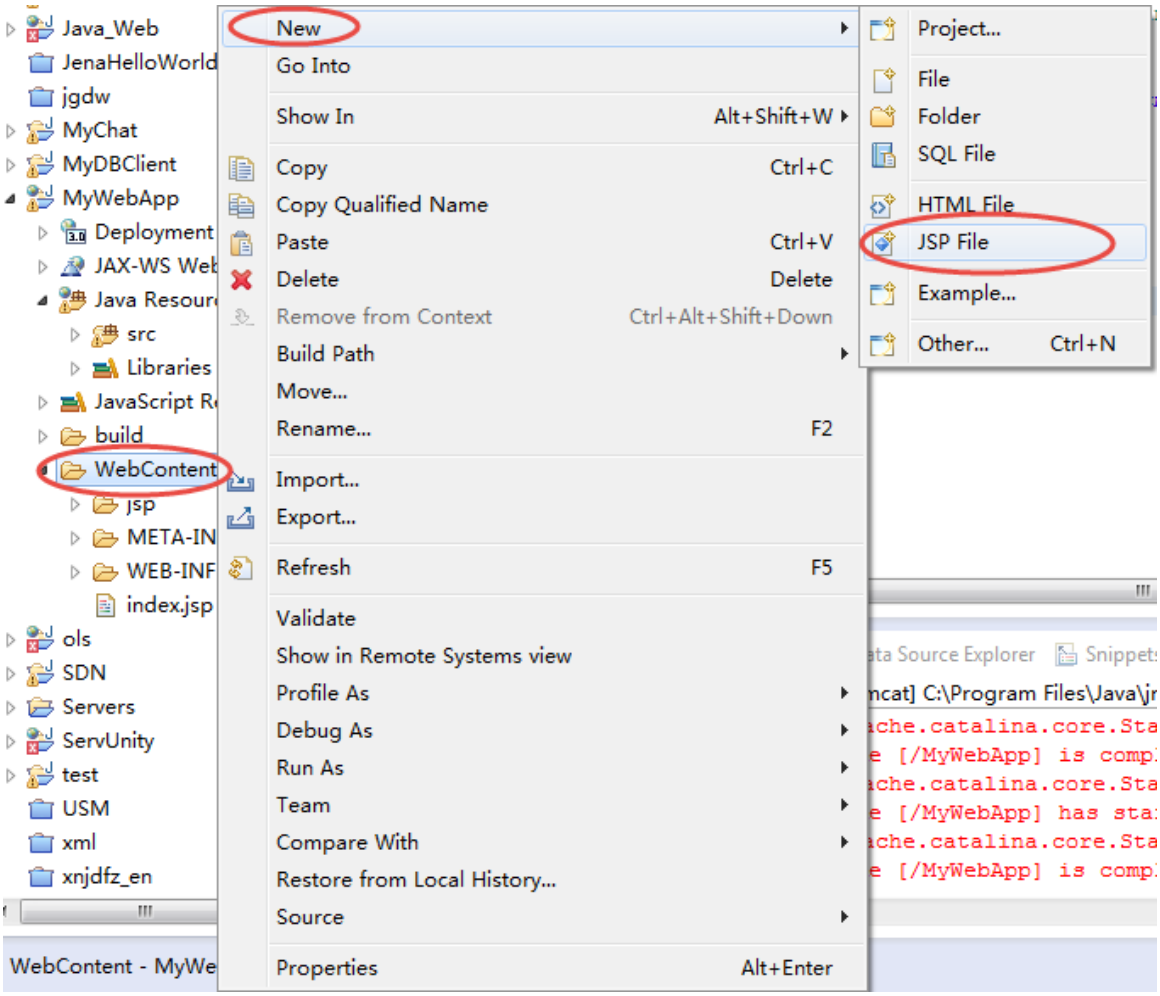


图 17

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dt
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
    这是主页index.jsp

    <!-- 这是注释，下述代码为超链接 -->
    <a href="hello"> 测试HelloWorldServlet </a>
</body>
</html>
```

新建index.jsp之后，此部分是需要手动添加的部分

图 18

然后在 WebContent 文件夹下新建一个文件夹 jsp，在 jsp 文件夹下新建 hello.jsp 页面，用于响应特定的 URL 请求（在 2.2.6 的 web.xml 文件中将做配置）。

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dt
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
    Hello World!
    <br> This is my first Java Web application!
</body>
</html>
```

图 19

2.2.6. 修改 web.xml 文件（配置文件）

web.xml 文件是整个 Web 应用的配置文件，用于指定特定的 URL 应该由哪些 Servlet 负责处理、Servlet 的实现类等关键配置信息。如图 20 所示，简要阐述一下：/hello 是一个 URL 模式，由于我们的项目部署在 http://localhost:8080/MyWebApp，因此/hello 的完整 URL 路径应该是：http://localhost:8080/MyWebApp/hello，该 URL 请求由 HelloWorldServlet 这个 Servlet 处理，该 Servlet 由 cx.course.jweb.servlet.HelloWorldServlet 类来实现。

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-ap
    id="WebApp_ID" version="3.0">
    <display-name>MyWebApp</display-name>

    <servlet>
        <!-- 名字可以随意取，但不得有非法字符 -->
        <servlet-name>HelloWorldServlet</servlet-name>
        <!-- 指向自己编写的Servlet类，注意需要包含包名的完整路径 -->
        <servlet-class>cx.course.jweb.servlet.HelloWorldServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <!-- 指向上述已建立的HelloWorldServlet -->
        <servlet-name>HelloWorldServlet</servlet-name>
        <!-- 用于处理如下URL -->
        <url-pattern>/hello</url-pattern>
    </servlet-mapping>

    <welcome-file-list>
        <welcome-file>index.html</welcome-file>
        <welcome-file>index.htm</welcome-file>
        <welcome-file>index.jsp</welcome-file>
        <welcome-file>default.html</welcome-file>
        <welcome-file>default.htm</welcome-file>
        <welcome-file>default.jsp</welcome-file>
    </welcome-file-list>
</web-app>
```

新添加的内容

图 20

2.2.7. 测试应用

两种方法可以测试应用，如下所示，使用其中一种即可（个人推荐第一种，这样对 URL 的层次结构会理解更深）。

(1) 打开浏览器，在浏览器键入 http://localhost:8080/MyWebApp，可进入主页（即 index.jsp 文件），如图 24 所示。

(2) 按照图 21-图 23 所示，可在 Eclipse 内部打开一个内置浏览器访问主页，结果类似图 24。

点击图 24 中“测试 HelloWorldServlet”链接，若能链接到图 25 所示页面，表示 HelloWorldServlet 工作正常，实验成功。

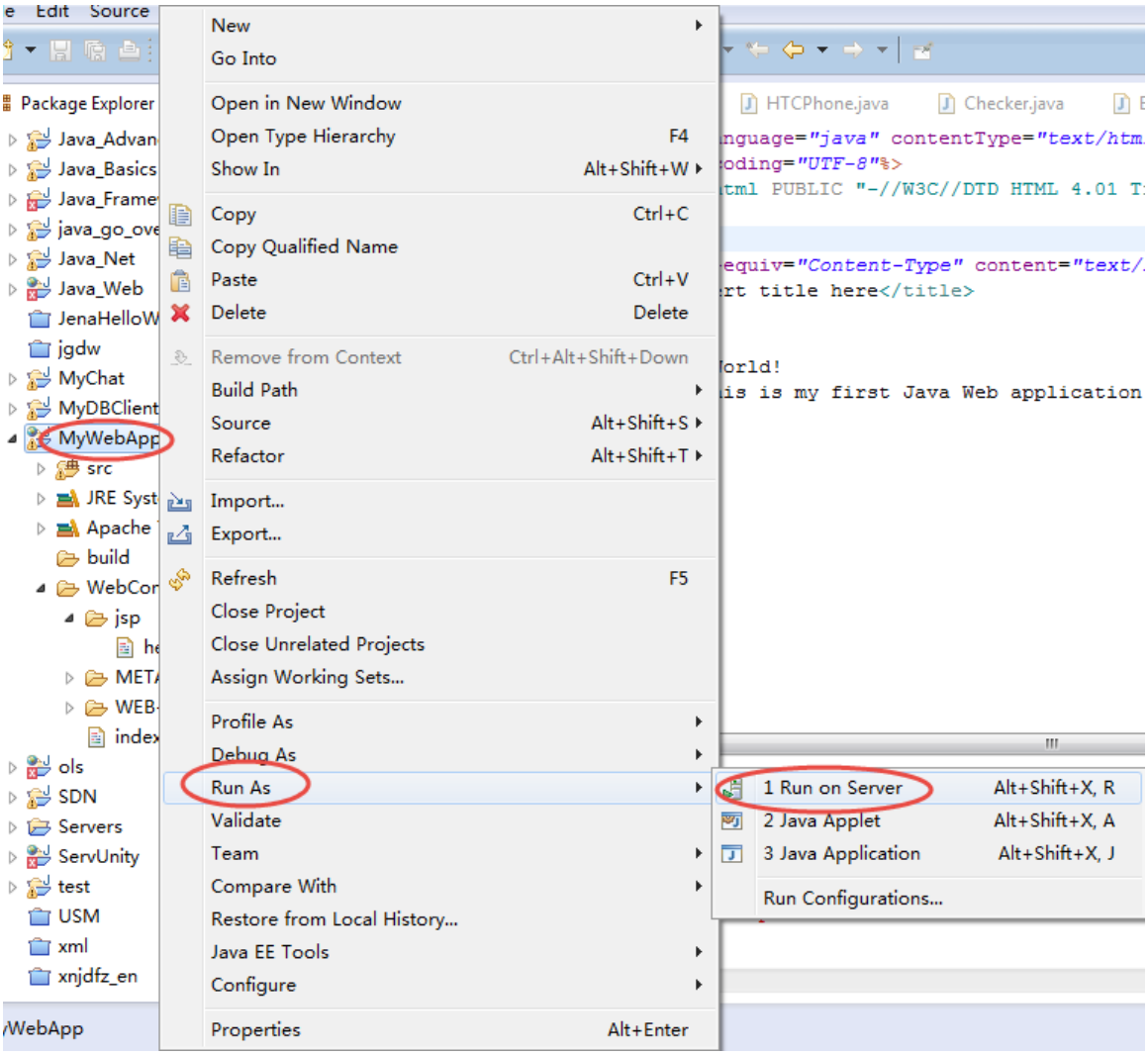


图 21

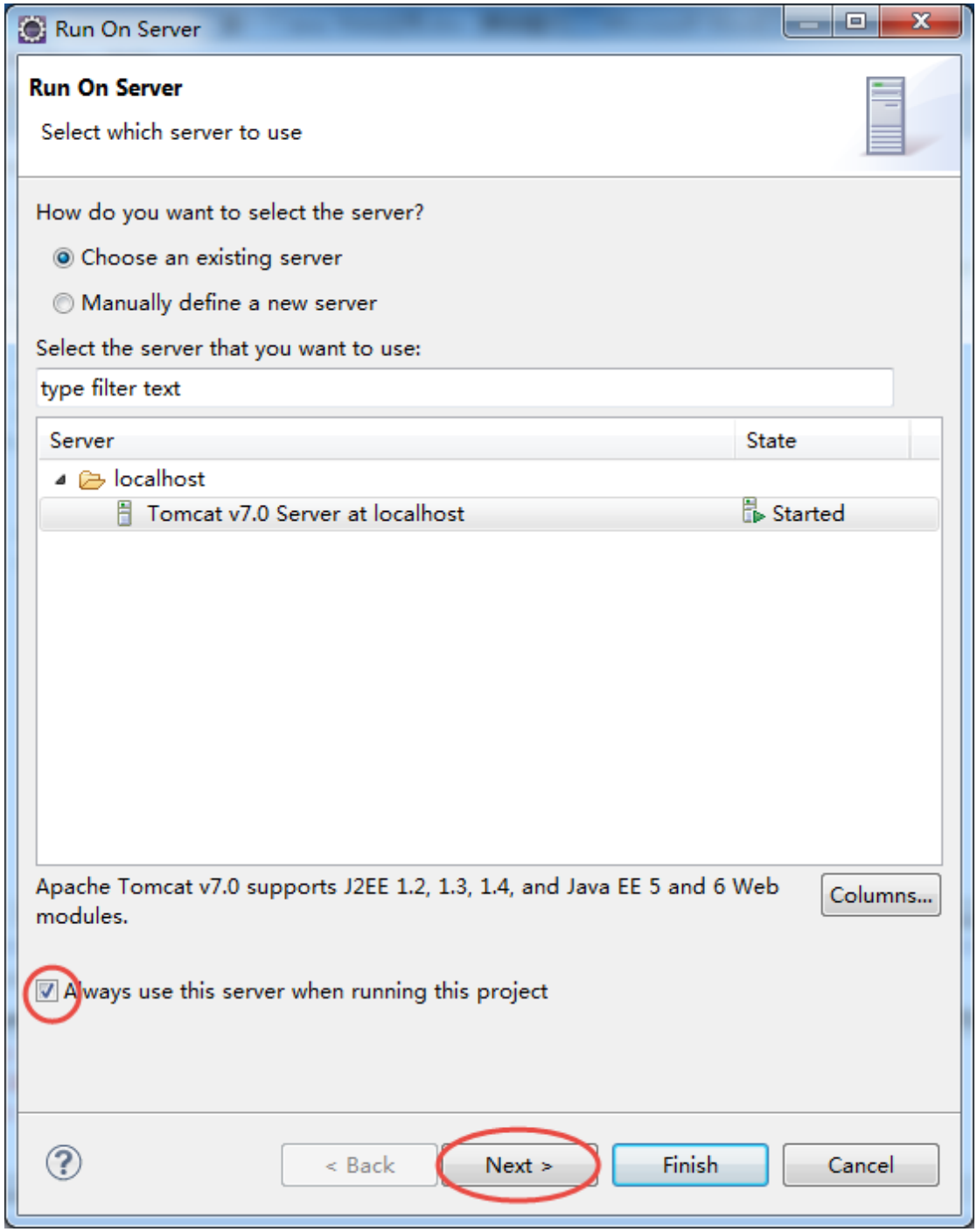

Run On Server
Run On Server
Select which server to use
How do you want to select the server?
Choose an existing server
Manually define a new server
Select the server that you want to use:
type filter text
Server
State
localhost
Tomcat v7.0 Server at localhost
Started
Apache Tomcat v7.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5 and 6 Web modules.
Columns...
Always use this server when running this project
< Back
Next >
Finish
Cancel

图 22

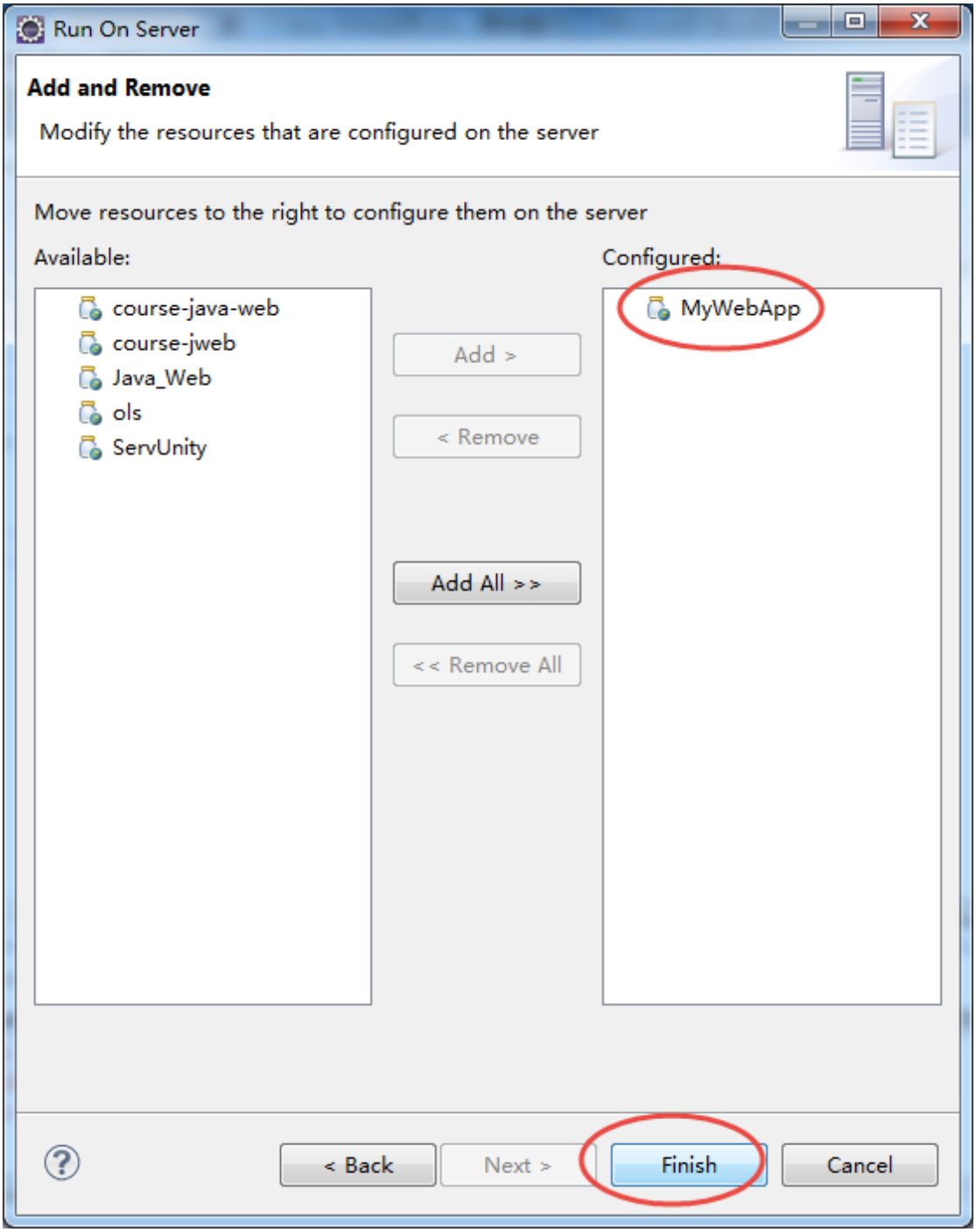

Run On Server
Add and Remove
Modify the resources that are configured on the server
Move resources to the right to configure them on the server
Available:
course-java-web
course-jweb
Java_Web
ols
ServUnity
Add >
< Remove
Add All >>
<< Remove All
Configured:
MyWebApp
< Back
Next >
Finish
Cancel

图 23

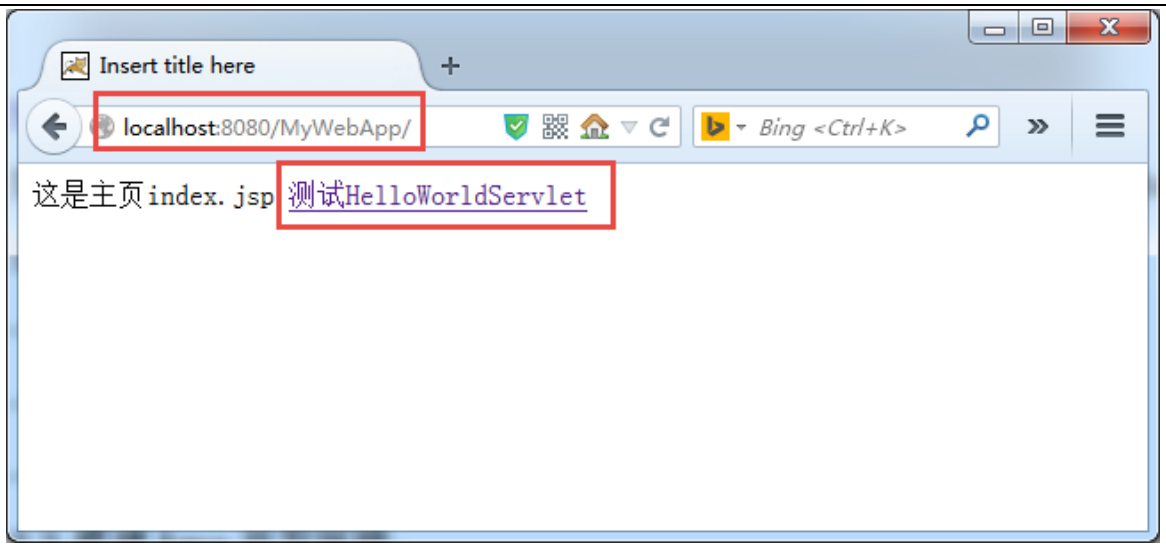


图 24

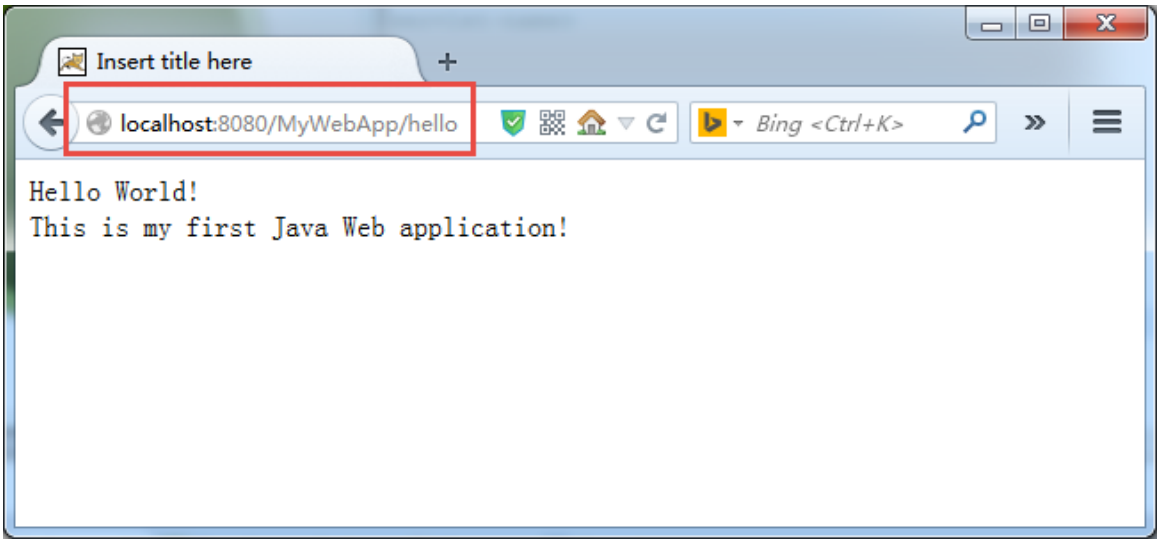


图 25

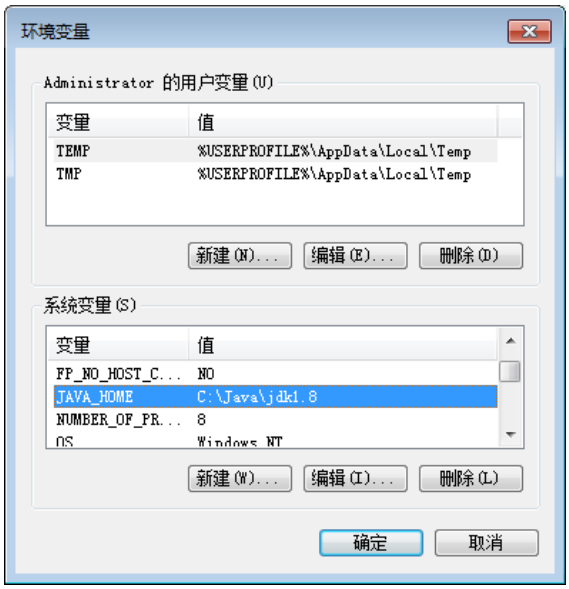
3、实验要求

4、实验主要过程与结果

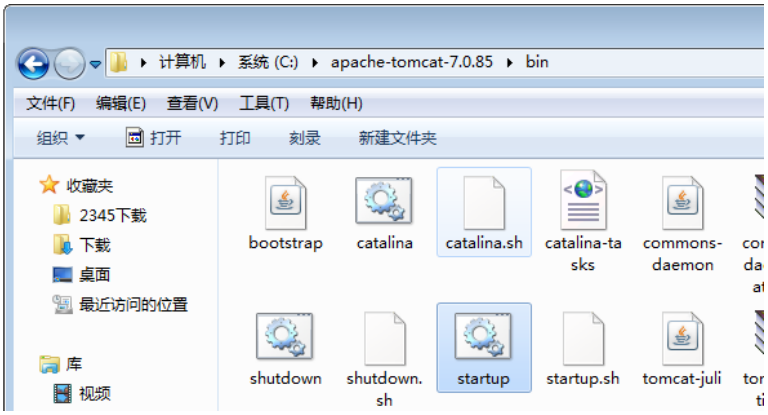
2.1 搭建 Java Web 开发环境

2.1.1 配置环境变量

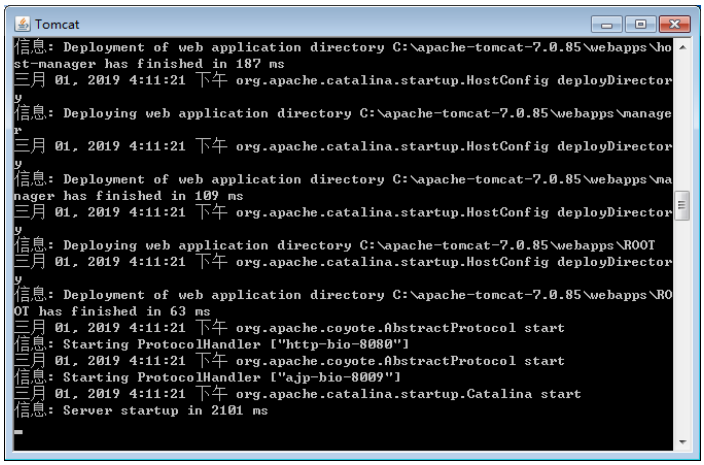
发现 JAVA_HOME 路径配置有误，重新配置



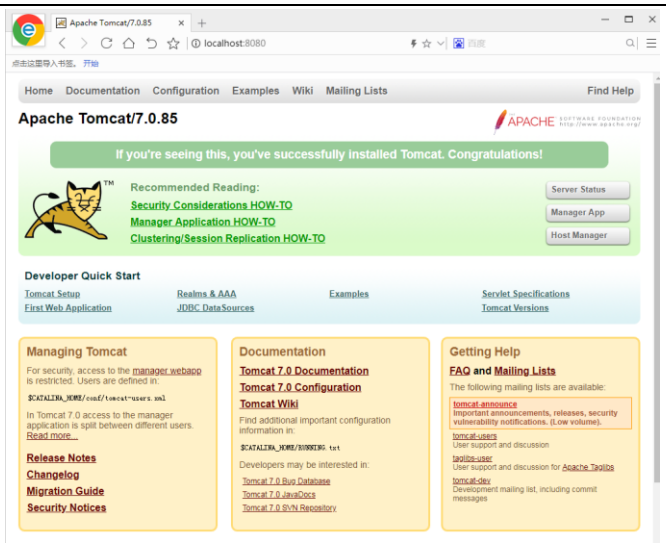
2.1.2 安装 Tomcat 服务器



启动成功

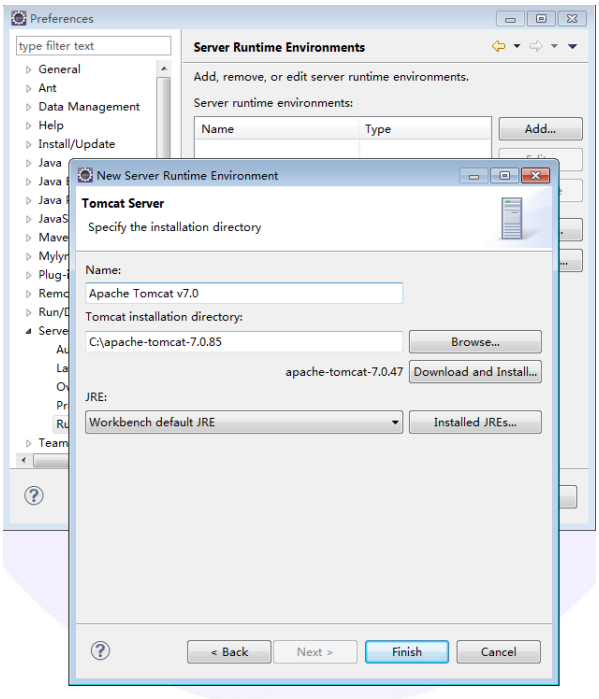


使用浏览器打开 http://localhost:8080/

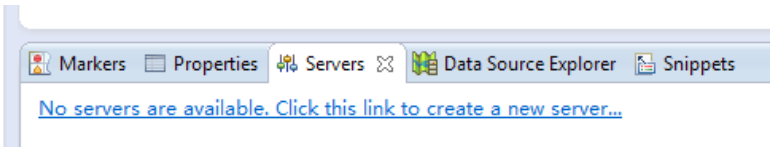


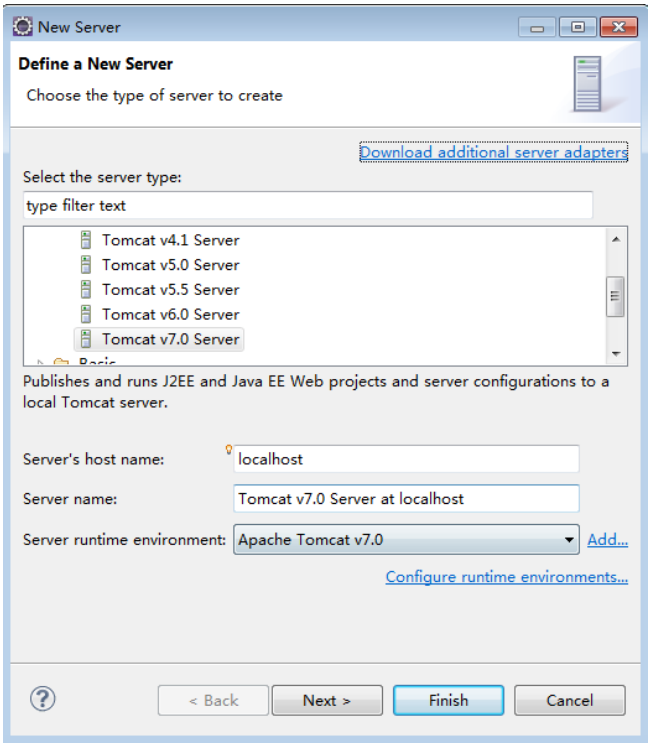
2.1.3 使用 Eclipse 连接 Tomcat

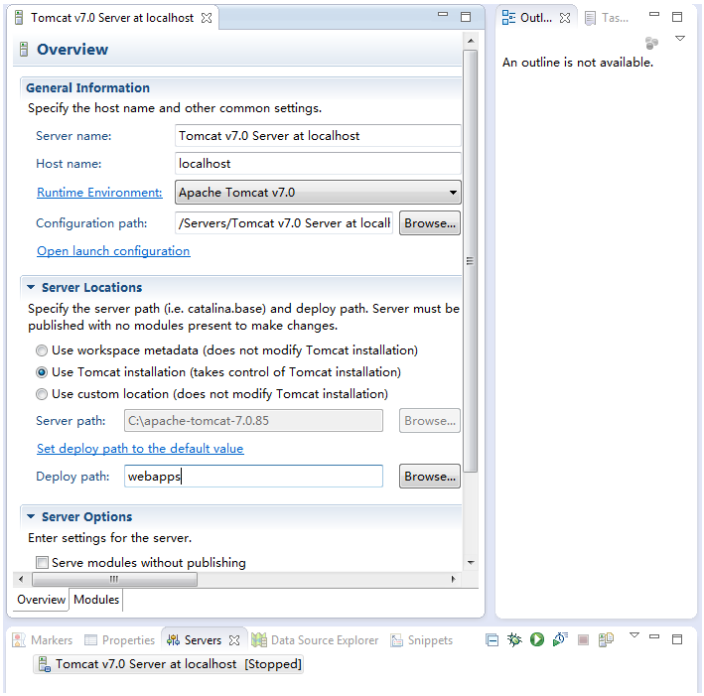
A. 配置服务器运行时环境

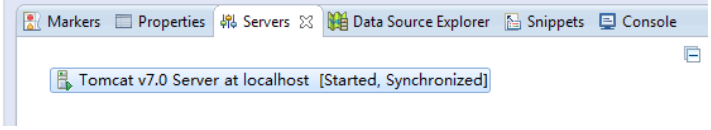


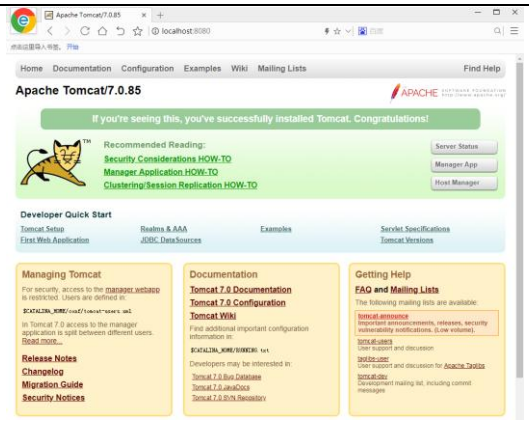
B. 配置服务器




New Server
Define a New Server
Choose the type of server to create
Download additional server adapters
Select the server type:
type filter text
Tomcat v4.1 Server
Tomcat v5.0 Server
Tomcat v5.5 Server
Tomcat v6.0 Server
Tomcat v7.0 Server
Publishes and runs J2EE and Java EE Web projects and server configurations to a local Tomcat server.
Server's host name:
localhost
Server name:
Tomcat v7.0 Server at localhost
Server runtime environment:
Apache Tomcat v7.0
Add...
Configure runtime environments...
< Back
Next >
Finish
Cancel

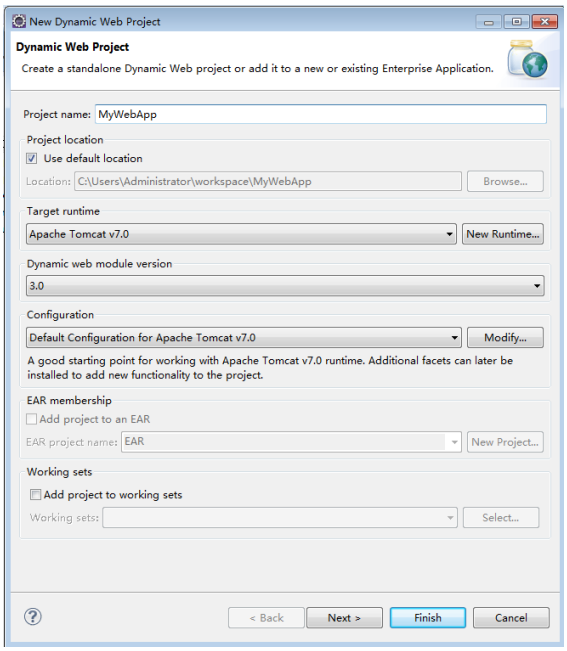

Tomcat v7.0 Server at localhost
Overview
General Information
Specify the host name and other common settings.
Server name:
Tomcat v7.0 Server at localhost
Host name:
localhost
Runtime Environment:
Apache Tomcat v7.0
Configuration path:
/Servers/Tomcat v7.0 Server at locall
Browse...
Open launch configuration
Server Locations
Specify the server path (i.e. catalina.base) and deploy path. Server must be published with no modules present to make changes.
Use workspace metadata (does not modify Tomcat installation)
Use Tomcat installation (takes control of Tomcat installation)
Use custom location (does not modify Tomcat installation)
Server path:
C:\apache-tomcat-7.0.85
Browse...
Set deploy path to the default value
Deploy path:
webapps
Browse...
Server Options
Enter settings for the server.
Serve modules without publishing
Overview
Modules
Outl...
Tas...
An outline is not available.
Markers
Properties
Servers
Data Source Explorer
Snippets
Tomcat v7.0 Server at localhost [Stopped]

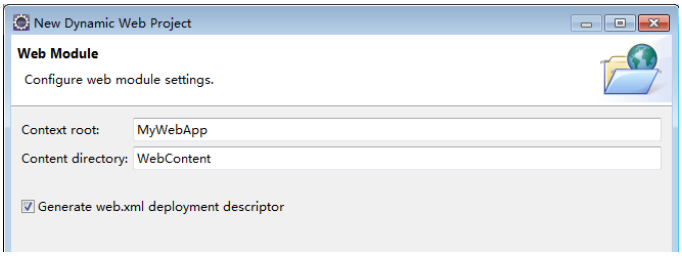

Markers
Properties
Servers
Data Source Explorer
Snippets
Console
Tomcat v7.0 Server at localhost [Started, Synchronized]

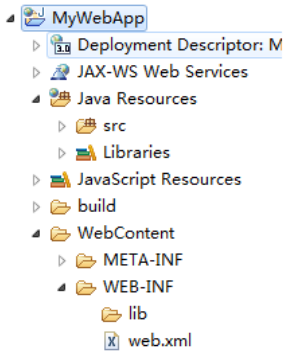


2.2 开发第一个 Java Web 应用

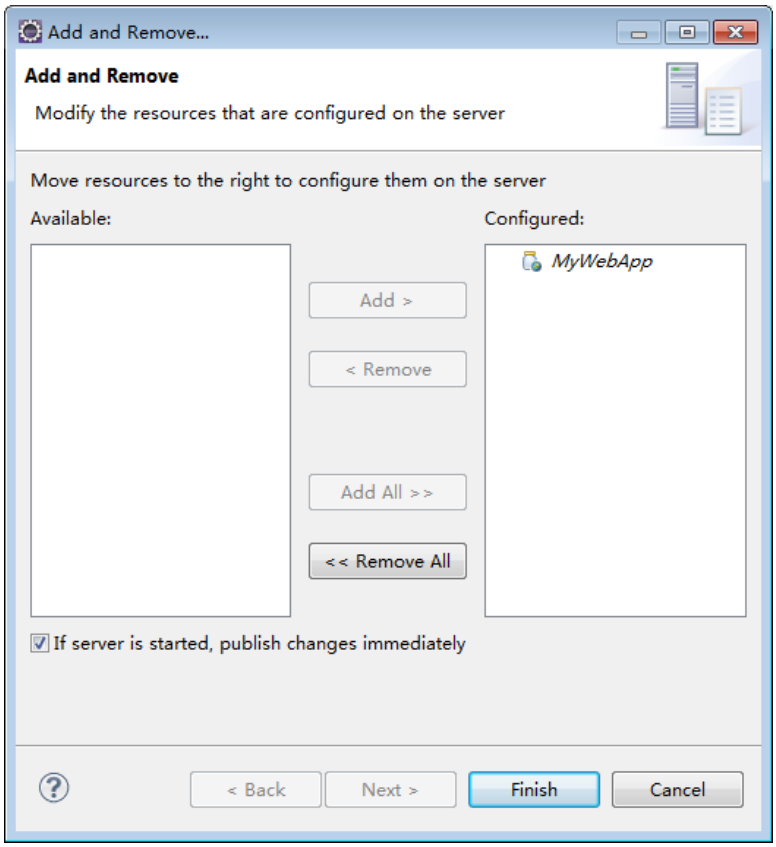
2.2.1 构建项目

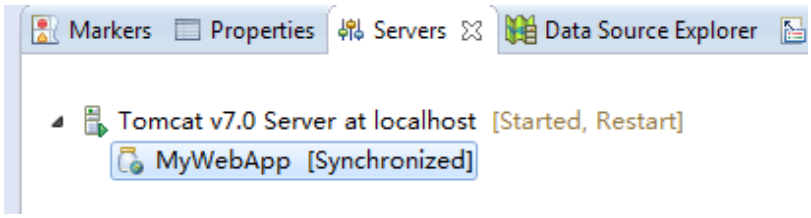




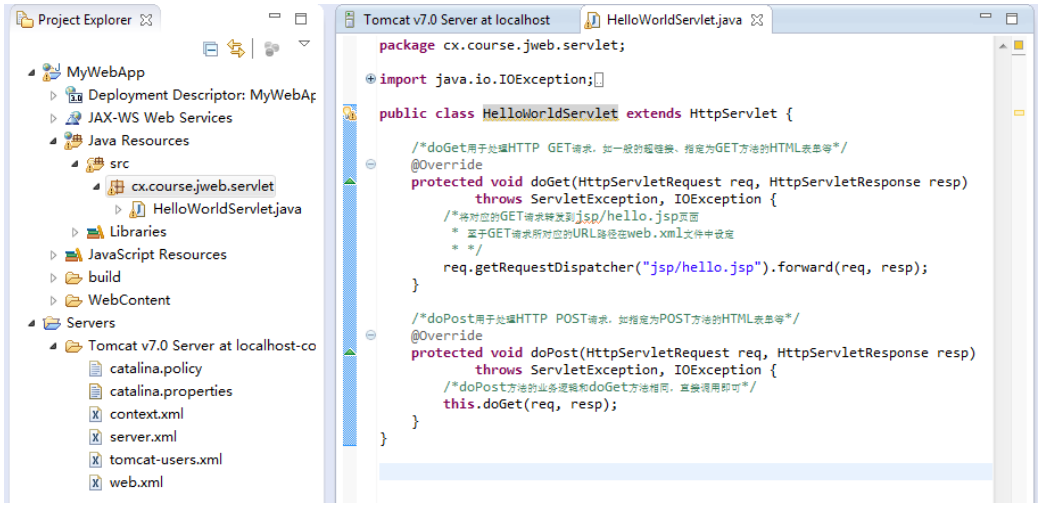


2.2.2 部署应用



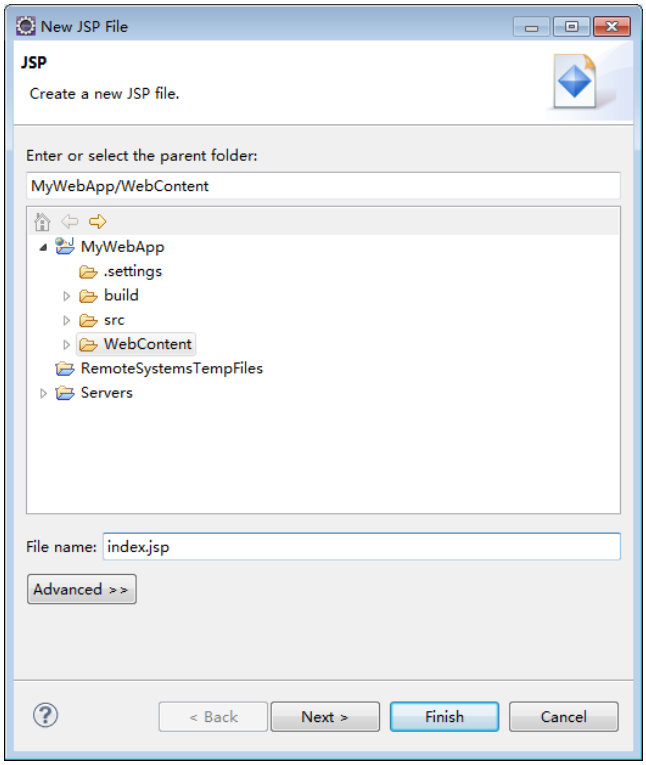


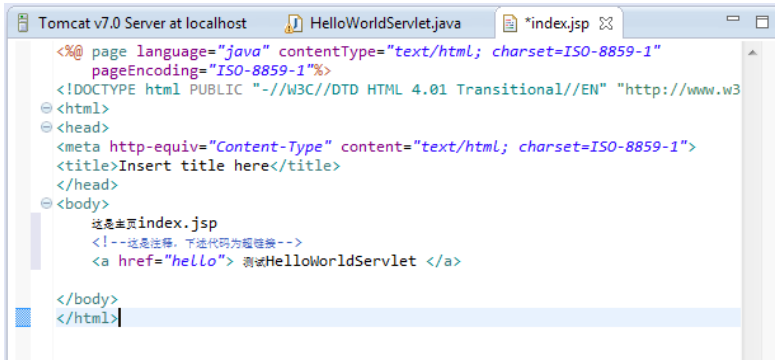
2.2.3 编写 Servlet 代码（控制层）

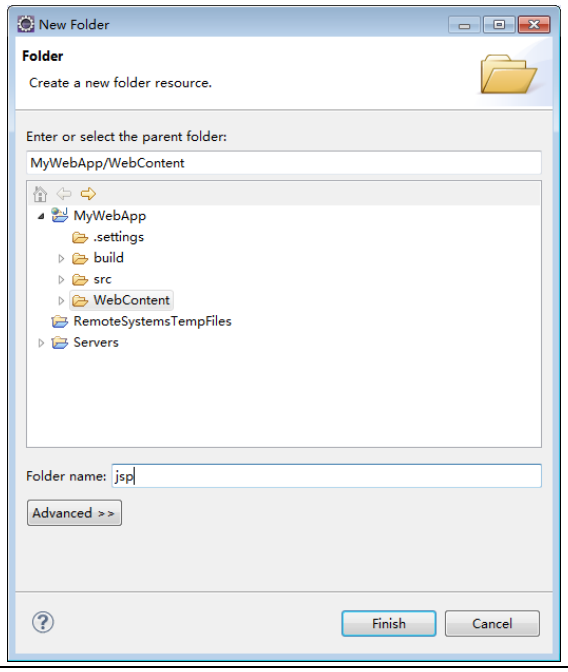


2.2.4 编写业务逻辑代码

2.2.5 编写 JSP 页面（展示层、前端页面）


New JSP File
JSP
Create a new JSP file.
Enter or select the parent folder:
MyWebApp/WebContent
MyWebApp
.settings
build
src
WebContent
RemoteSystemsTempFiles
Servers
File name: index.jsp
Advanced >>
< Back
Next >
Finish
Cancel


Tomcat v7.0 Server at localhost
HelloWorldServlet.java
*index.jsp
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
这是主页index.jsp
<!--这是注释，下述代码为超链接-->
 测试HelloWorldServlet
</body>
</html>


New Folder
Folder
Create a new folder resource.
Enter or select the parent folder:
MyWebApp/WebContent
MyWebApp
.settings
build
src
WebContent
RemoteSystemsTempFiles
Servers
Folder name: jsp
Advanced >>
Finish
Cancel

Tomcat v7.0 S... HelloWorldSer... index.jsp hello.jsp web.xml

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
    这是主页index.jsp
    <!--这是注释，下述代码为超链接-->
    <a href="hello"> 测试HelloWorldServlet </a>

</body>
</html>
```

Tomcat v7.0 S... HelloWorldSer... index.jsp hello.jsp web.xml

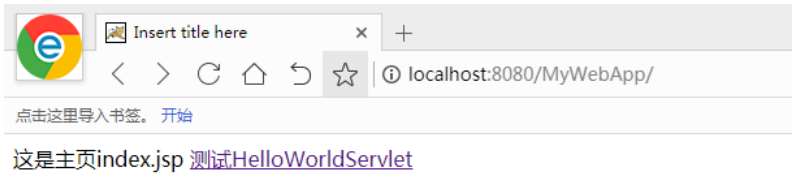
```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
    Hello World!
    <br> This is my first Java Web application!

</body>
</html>
```

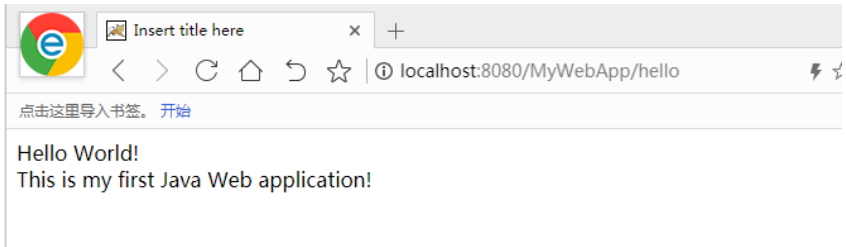
2.2.6 修改 web.xml 文件（配置文件）

Tomcat v7.0 S... HelloWorldSer... index.jsp hello.jsp web.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://
  <display-name>MyWebApp</display-name>

  <servlet>
    <servlet-name>HelloWorldServlet</servlet-name>
    <servlet-class>cx.course.jweb.servlet.HelloWorldServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>HelloWorldServlet</servlet-name>
    <url-pattern>/hello</url-pattern>
  </servlet-mapping>

  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

2.2.7 测试应用





5、简答题

如图 20 所示，/hello 指明了一个 URL，请问/代表什么？

答：“/”代表从当前端口向后，即 http://localhost:8080/MyWebApp/hello。

注：实验报告的内容及格式可由学院根据学科专业特点确定；全校各专业必须使用学校统一封面。