* 为什么需要servlet技术？

比如需求：我们希望用户可以发帖，用户还可以回复....这样一些和用户交互的功能，用普通的Java技术就完成不了，sun就开发了servlet供程序员使用

Servlet(java服务器小程序)

* Servlet的介绍

①servlet其实就是java程序

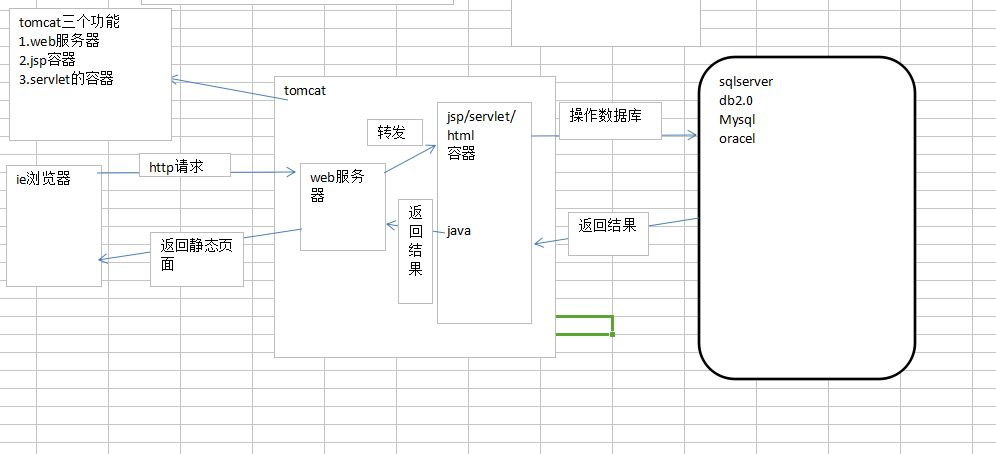
②该java程序（java类）要遵循servlet的开发规范

③servlet是运行在服务器端的

④servlet功能强大，几乎可以完成网站的所有功能

⑤是学习jsp的基础

* Servlet和tomcat在网络中的地位



* Servlet的快速入门案例

①实现servlet接口

②继承Generic Servlet

③通过继承HttpServlet

需求如下：请使用实现接口的方式，来开发一个Servlet，要求该servlet可以显示helloworld同时显示当前时间

步骤：

①建立一个web应用wsxWeb

②在wsxWeb下建立一个WEB-INF/web.xml[web.xml可以从ROOT/WEB-INF/web.xml拷贝]

③在wsxWeb下建立classes目录(我们的Servlet就要在该目录下开发)建立lib文件夹

④开发Myfirst.java

package com.wsx;

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.io.\*;

class Myfirst implements Servlet

{

//该函数用于初始化servlet，就是把该servlet装载到内存中去

//该函数只会被调用一次

public void init (ServletConfig config)

throws ServletException{

}

//得到ServletConfig对象

public ServletConfig getServletConfig(){

return null;

}

//该函数是服务函数，我们的业务逻辑代码就是写在这里的

//该函数每次都会被调用

public void service (ServletRequest req,

ServletResponse res)

throws ServletException,

java.io.IOException{

}

//该函数是得到servlet配置信息

public java.lang.String getServletInfo(){

return null;

}

//该函数是把当前servlet从内存中清除，该函数被调用一次

public void destroy(){

}

}

⑤根据servlet的规范，我们还要部署servlet

<?xml version="1.0" encoding="ISO-8859-1"?>

<!--

Licensed to the Apache Software Foundation (ASF) under one or more

contributor license agreements. See the NOTICE file distributed with

this work for additional information regarding copyright ownership.

The ASF licenses this file to You 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.

-->

<web-app xmlns="http://java.sun.com/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/javaee

http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"

version="3.0"

metadata-complete="true">

<!--根据servlet规范，需要将Servlet部署到web.xml文件，该部署配置可以从examples去拷贝-->

<servlet>

<!--servlet-name 给该servlet取名字，该名字可以自己定义：默认就是用该servlet的名字-->

<servlet-name>Myfirst</servlet-name>

<!--servlet-class 要指明该servlet 放在哪个包下的，形式是 包/包..../类-->

<servlet-class>com.wsx.Myfirst</servlet-class>

</servlet>

<servlet-mapping>

<!--这个servlet-name要和上面的servlet-name名字一样-->

<servlet-name>Myfirst</servlet-name>

<!--url-pattern 这里就是将来访问该servlet的资源名部分，默认命名规范：就是该servlet的名字-->

<url-pattern>/ABC</url-pattern>

</servlet-mapping>

</web-app>

Tomcat寻找servlet的顺序：先去<servlet-mapping>里找url-pattern，看他的servlet-name，然后去<servlet>里找同名的servlet，调用它的servlet-class

⑥测试

在浏览器输入

http://localhost：8080/wsxWeb/ABC

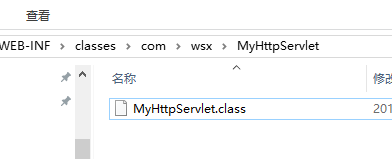
⑦分析一下自己写可能出现的错误

1. <servlet-name>Myfirst</servlet-name> 名字不一样（启动tomcat就出错）
2. <servlet-class>com.wsx.Myfirst</servlet-class> 写成Myfirst.java，会报告500错误

Servlet-class中的地址要一直写到类名，一定要写全

例如：package com.wsx.MyHttpServlet;

编译后如图



一定要写到类名

servlet-class中写

<servlet-class>com.wsx.MyHttpServlet.MyHttpServlet</servlet-class>

1. 资源名自己写错了 http://localhost：8080/wsxWeb/ABCdasdasd

错误的资源url-pattern 404 错误

补充：

如果使用javac去编一个带打包的java文件，则需要带命令参数

javac-d . Java文件

补充：如何不重启tomcat，就去制定一个web应用.方法：

进入到tomcat的manager，点击reload

* Servlet的生命周期/servlet究竟是怎么工作的

UML时序图

参看Excel图

* 简述servlet生命周期：

1. 当servlet第一次被调用时，会触发init函数，该函数会把servlet实例装载到内存，init函数智慧被调用一次，调用后无论哪个浏览器或者用户调用都不会再重新创建
2. 然后去调用servlet的service函数
3. 当第二次以后访问该servlet就直接调用service函数
4. 当web应用 reload的时候或者关闭tomcat再或者关机都回去调用destroy函数，这个函数会去销毁servlet

* 简述servlet运行过程：

1. web服务器搜西安检查是否已经装载并创建了该servlet的实例对象，。如果是，则直接执行第四步，否则执行第二步。
2. 装载并创建该servlet的一个实例对象。
3. 调用该servlet实例对象的init()方法。
4. 创建一个用于封装http请求的消息的HttpServletRequest对象和一个代表Http响应消息的HttpServletResponse对象，然后调用Servlet的service()方法并将请求和响应对象作为参数传递过去。
5. Web应用程序被停止或者重新启动前，servlet引擎将卸载servlet，并在卸载之前调用servlet的destroy()方法。

以下情况会导致destroy()方法调用：

1. tomcat重新启动
2. Reload该webAPP
3. 重新启动电脑

* 使用GenericServlet开发servlet

package com.wsx.MyGenericServlet;

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.io.\*;

public class MyGenericServlet extends GenericServlet {

public void service(ServletRequest req,

ServletResponse res)

throws ServletException,

java.io.IOException{

res.getWriter().println("heelo,world i am Generic Servlet.");

}

}

将该servlet部署到web.xml文件中

<?xml version="1.0" encoding="ISO-8859-1"?>

<!--

Licensed to the Apache Software Foundation (ASF) under one or more

contributor license agreements. See the NOTICE file distributed with

this work for additional information regarding copyright ownership.

The ASF licenses this file to You 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.

-->

<web-app xmlns="http://java.sun.com/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/javaee

http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"

version="3.0"

metadata-complete="true">

<!--根据servlet规范，需要将Servlet部署到web.xml文件，该部署配置可以从examples去拷贝-->

<servlet>

<!--servlet-name 给该servlet取名字，该名字可以自己定义：默认就是用该servlet的名字-->

<servlet-name>MyGenericServlet</servlet-name>

<!--servlet-class 要指明该servlet 放在哪个包下的，形式是 包/包..../类-->

<servlet-class>com.wsx.MyGenericServlet</servlet-class>

</servlet>

<servlet-mapping>

<!--这个servlet-name要和上面的servlet-name名字一样-->

<servlet-name>MyGenericServlet</servlet-name>

<!--url-pattern 这里就是将来访问该servlet的资源名部分-->

<url-pattern>/MyGenericServlet</url-pattern>

</servlet-mapping>

</web-app>

* 使用继承HttpServlet的方法来开发servlet

1. 在软件公司 90%都是通过该方法开发
2. 举例说明：

package com.wsx.MyHttpServlet;

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.io.\*;

public class MyHttpServlet extends HttpServlet

{

//在httpservlet中，设计者对post提交和get提交分别处理了

//回忆<form action="提交给谁"，method=“post/get”>,默认是get提交

//其实doGet和doPost最终也去调用service方法

protected void doGet(HttpServletRequest req,

HttpServletResponse resp)

throws ServletException,

java.io.IOException{

resp.getWriter().println("i am httpservlet doGet()");

}

protected void doPost(HttpServletRequest req,

HttpServletResponse resp)

throws ServletException,

java.io.IOException{

resp.getWriter().println("i am httpservlet doPost() username="+req.getParameter("username"));

}

}

还有一个login.html

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="Generator" content="EditPlus®">

<meta name="Author" content="">

<meta name="Keywords" content="">

<meta name="Description" content="">

<title>Document</title>

</head>

<body>

<form action="/wsxWeb/MyHttpServlet" method="post">

username：<input type="text" name="username" />

<input type="submit" value="login" />

</form>

</body>

</html>

* Get提交和Post提交的区别：

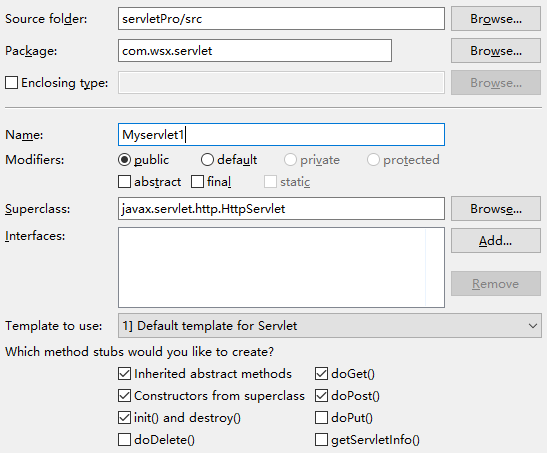
1. 从安全性看get<post，get提交的数据会显示在地址栏里
2. 从提交内容的大小来看，get提交的数据不能大于2K，post提交的数据理论上不限制大小，但实际编程中最好不要超过64K
3. 从响应速度来看:get>post，get要求服务器立刻响应，post有可能会形成一个访问队列

* 使用IDE（eclipse【java se】+myeclipse【插件，可以支持jsp/servlet/hibernate/spring】）开发servlet

需要使用IDE开发一个servlet，该servlet显示hello，world和当前日期

步骤：

1. 建立web工程
2. 在src目录下建立了一个包com.wsx.servlet
3. 开发一个servlet



Myservlet1的代码：

package com.wsx.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class Myservlet1 extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

out.println("hello,world"+new java.util.Date().toString());

}

public void doPost(HttpServletRequest request, HttpServletResponse response)

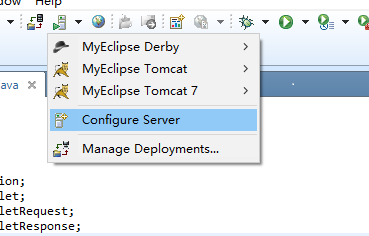
throws ServletException, IOException {

this.doGet(request, response);

}

}

1. 配置tomcat



5.启动tomcat

老方法；

从eclipse启动tomcat；