# Servlet

## Servlet概述

### 什么是Servlet?

Servlet是由SUN公司提供的动态web资源开发技术, 本质上是一段java程序, 这段java程序无法独立运行, 必须放在Servlet容器(比如: tomcat服务器)中, 由容器调用才可以执行!

### 开发Servlet程序的步骤

1.写一个类, 实现一个Servlet接口, 并添加未实现的方法

2.将编译后类, 放在WEB应用中, 并在WEB应用web.xml文件中配置Servlet对外访问的虚拟路径, 将WEB应用部署到虚拟主机中即可

### 使用记事本开发一个Servlet程序

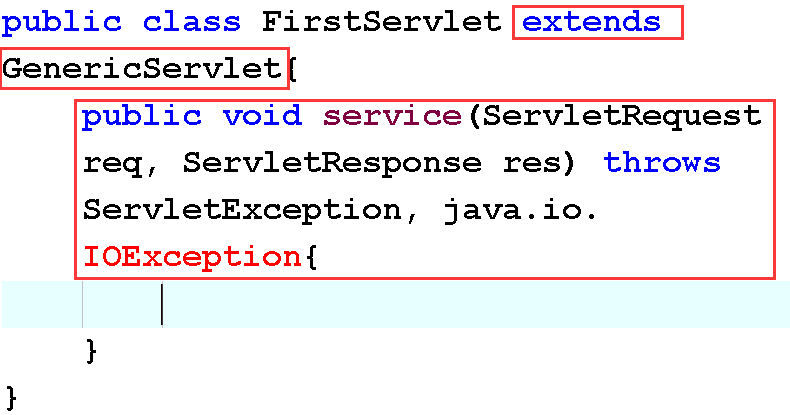
1.需求: 浏览器向服务器发送一个请求, 访问FirstServlet, FirstServlet将表示当前时间的字符串发送浏览器.

2.详细开发步骤

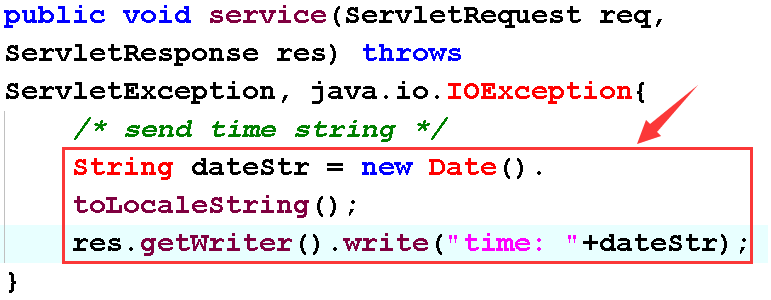
（1）创建一个FirstServlet.java文件



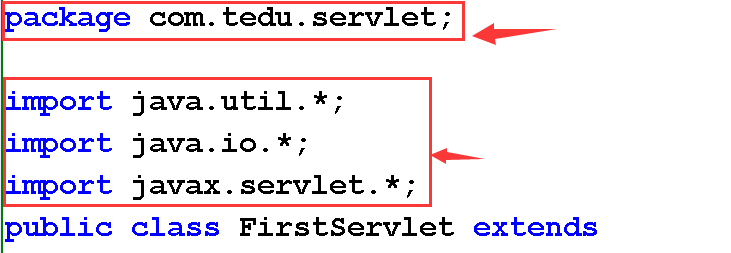
（2）编写FirstServlet类, 实现Servlet接口(或继承Servlet接口的子类), 并添加未实现的方法



（3）实现Servlet中提供的处理请求的方法service方法, 将表示当前时间的字符串响应给浏览器

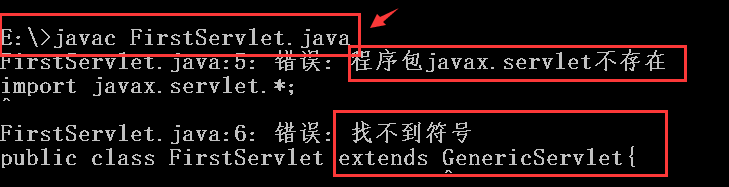


（4）细节处理(添加servlet类的包路径和导入依赖包)



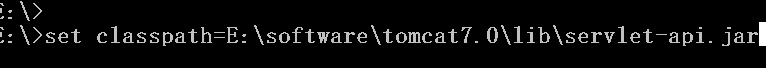
（5）打开CMD窗口, 编译FirstServlet类

通过命令编译会出现如下错误:

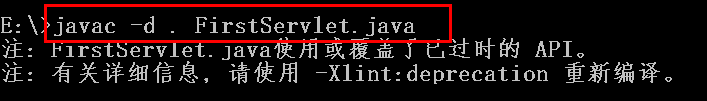


原因是 javax.servlet这个包不存在, 所有下面出现了很多找不到符号错误.

在tomcat服务器的lib目录下有servlet的jar包, 可以设置包路径:

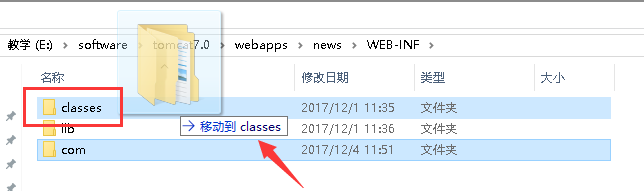


接着再次进行编译:

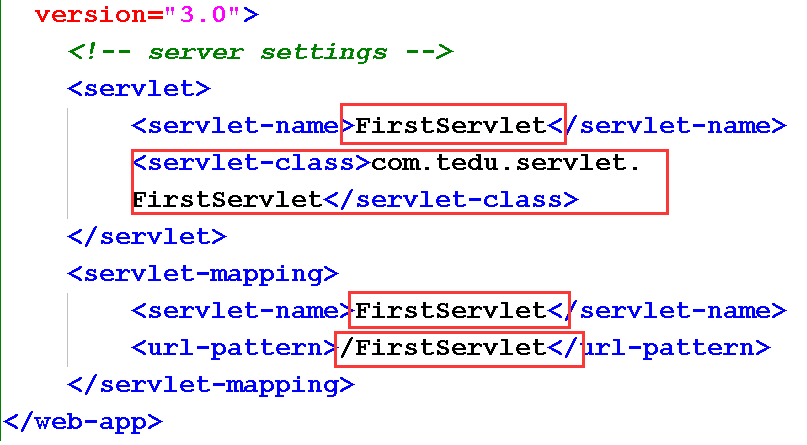


（6）将FirstServlet类放在WEB应用中, 并在web.xml文件中配置Servlet对外访问的虚拟路径

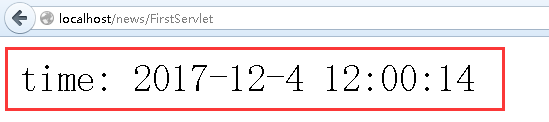
将编译好的class文件(包括包路径)放在web应用的classes目录下.



在WEB应用的web.xml文件中配置servlet对外访问的虚拟路径:



（7）将WEB应用部署到虚拟主机中, 启动服务器, 通过浏览器访问FirstServletServlet.



## Servlet继承结构

Servlet接口 – 通用的Servlet接口, 提供了一个Servlet应该具有的功能

|

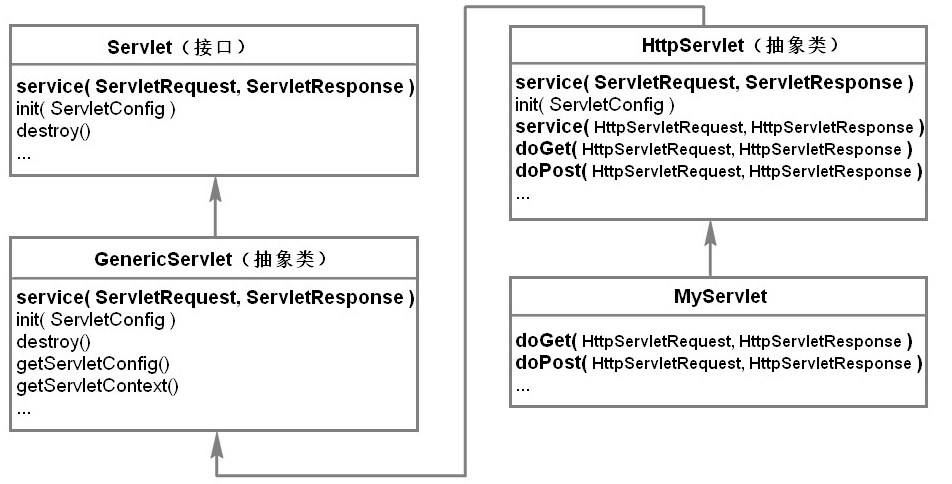
|--- GenericServlet类, 实现了Servlet接口, 并实现了其中的大部分的方法, 但是service方法没有实现, 这个方法需要开发人员自己来实现.

|

|--- HttpServlet类, 继承了GenericServlet, 并实现了service方法, 在service方法中, 判断请求方式, 根据不同的请求方式xxx调用不同doXxx

|

|-- XxxServlet类, 在开发中, 我们只需要写一个类(XxxServlet), 继承HttpServlet, 并覆盖doGet和doPost方法, 来处理GET请求和POST请求即可!!



## Servlet调用过程

### Servlet调用过程

开发一个Servlet, 放在WEB应用中, 并将WEB应用部署在服务器中, 启动服务器, 通过浏览器访问Servlet, 可以访问到Servlet处理的结果, 这个Servlet是如何被调用的, Servlet又是如何执行的?

参考<<Servlet调用过程图>>

### Servlet生命周期

Servlet在第一次被访问时创建Servlet实例, 创建之后服务器会立即调用init方法进行初始化的操作, 创建之后, Servlet实例会一直驻留在服务器的内存中, 为后续的请求服务. 只要有请求来访问这个Servlet, 服务器就会调用service方法来处理请求, 直到服务器关闭, 或者WEB应用被移出容器, 随着WEB应用的销毁, Servlet实例也会跟着销毁, 在销毁之前服务器会调用destroy方法进行善后的处理.

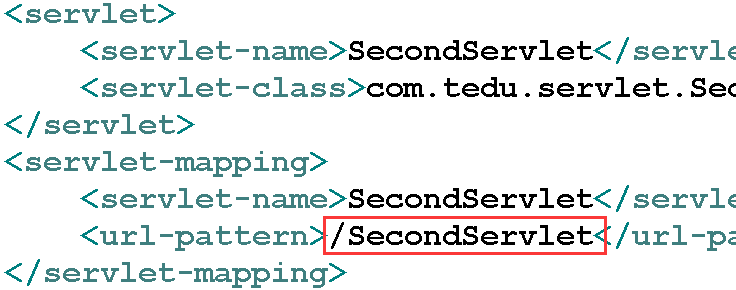
## Servlet开发细节

### Servlet虚拟路径的配置

Servlet在web.xml文件中虚拟路径的配置方式可以有两种方式:

1、直接写一个路径

如: /SeondServlet



后期需要通过 /SecondServlet这个路径来访问该Servlet

2、还可以使用星号(\*)匹配符来配置一个路径, 配置的方式有两中:

（1）以斜杠(/)开头, 以斜杠星(/\*)结尾.

比如: /\*、/a/\*...

（2）以星点后缀(\*.后缀)的形式

比如: \*.html、\*.abc、\*.do

问题：使用星号匹配符可以使得路径的访问变得更加灵活， 但是也可能会导致一个url会被多个mapping所匹配。路径的优先级规则是什么？

（1）哪一个更接近（更精准、更像）哪一个起作用

（2）\*.后缀的优先级永远最低.

**示例：**

Servlet1 映射到 /abc/\*

Servlet2 映射到 /\*

Servlet3 映射到 /abc

Servlet4 映射到 \*.do

当请求URL为“/abc/a.html”，“/abc/\*”和“/\*”都匹配，哪个servlet响应

servlet1将会匹配

当请求URL为“/abc”时，“/abc/\*”和“/abc”都匹配，哪个servlet响应

servlet3将会匹配

当请求URL为“/abc/a.do”时，“/abc/\*”和“\*.do”都匹配，哪个servlet响应

servlet1将会匹配

当请求URL为“/a.do”时，“/\*”和“\*.do”都匹配，哪个servlet响应

servlet2将会匹配

### 修改Servlet模版

通过Eclipse可以直接创建一个Servlet类，这相比通过记事本等文本编辑工具创建Servlet，可以节省配置Servlet的时间，提高了我们的开发效率。但是通过Eclipse生成的Servlet类中包含了许多我们不需要的注释和默认实现代码，这些每次都删除也非常占用时间。接下来可以通过添加模版代码的形式，来生成Servlet的内容，以便于提高我们的开发效率。

**1、先创建一个Servlet，将其中的内容修改为自己期望的模版格式，并复制其中的内容，例如：**

package com.tedu;

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 TestServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

}

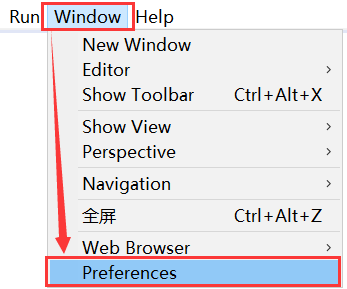
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

doGet(request, response);

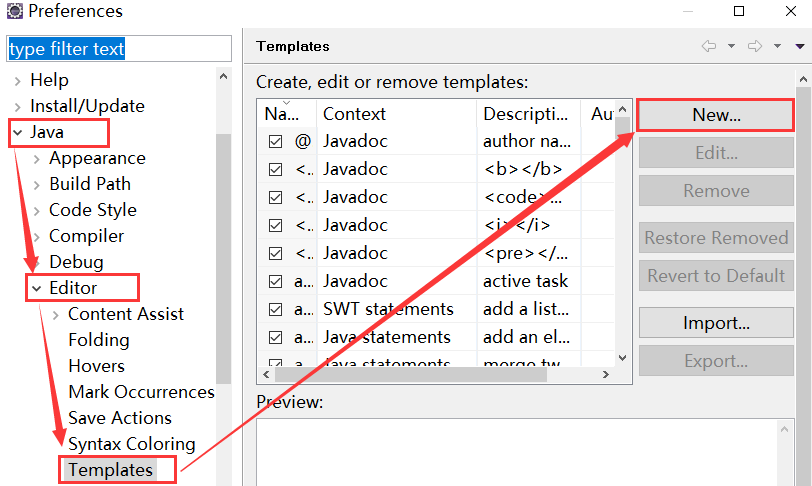
}

}

**2、点击菜单栏中的window --> Preferences**：



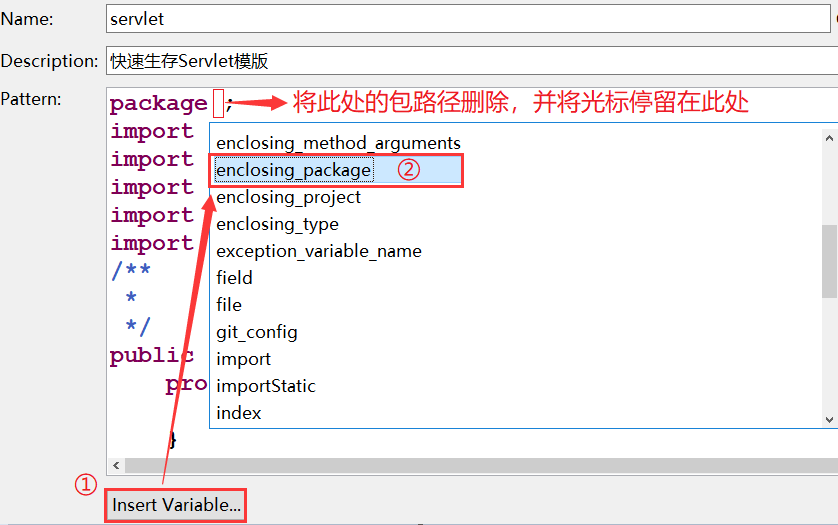
**3、在出现的窗口左侧依次点击：Java --> Editor --> templates -->(在右边的窗口中) 点击New… ：**



**4、在出现的新窗口中填写如下内容：**

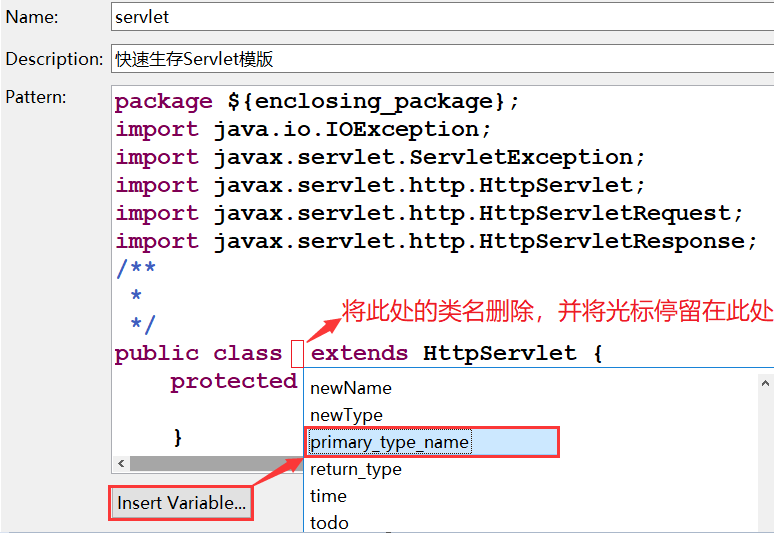


**5、替换包路径和类名（作用是在新建Servlet生成的Servlet模版中使用当前类的包路径和类型）**



效果如下：



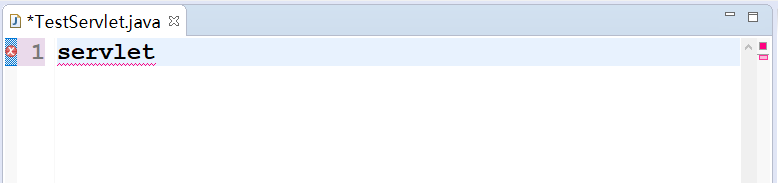


效果如下：



**6、点击OK保存，创建新的Servlet文件，测试**：

将Servlet中的所有内容全选删除，并输入"servlet"，接着按 "Alt+ /" 提示即可生成自己想要的Servlet模版内容!



效果如下:

