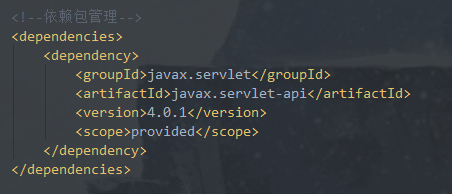
# Servlet实例

## pom.xml文件配置



1. 因为servlet包在tomcat容器的依赖中已经存在，所以scope设为provided，只在编译和测试中生效，不然会在runtime中和tomcat的依赖包产生冲突。

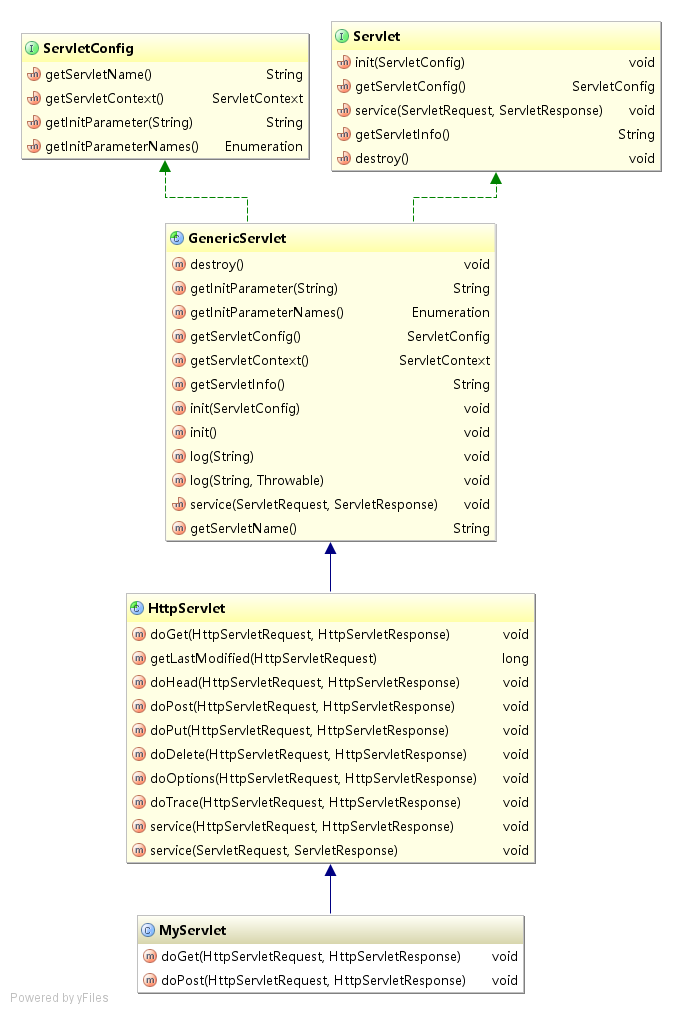
## 继承Servlet接口

### 2.1 自定义servlet继承Servlet接口

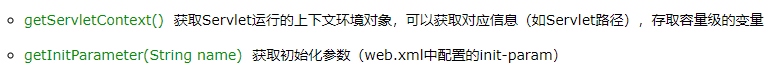


1. 继承HttpServlet，实现了doGet()和doPost()接口，以便来处理Http请求。

### 2.2 Servlet接口继承图



1. **ServletConfig：** Servlet的配置信息，常用来在Servlet初始化时进行信息传递



1. **Servlet：**Servlet的生命周期主接口

* public void init(ServletConfig config) throws ServletException;

初始化servlet配置信息

* public void service(ServletRequest req, ServletResponse res)

throws ServletException, IOException;

service为主要处理方法

* public String getServletInfo();
* public ServletConfig getServletConfig();

获取servlet基本信息以及配置信息

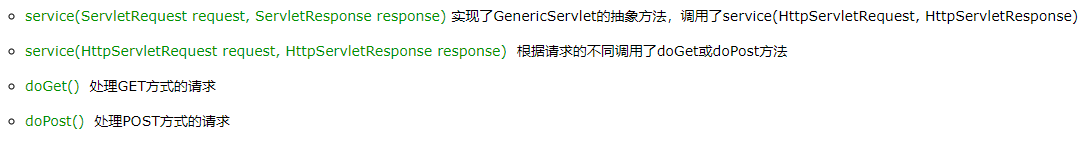
* public void destroy();

servlet的销毁方法

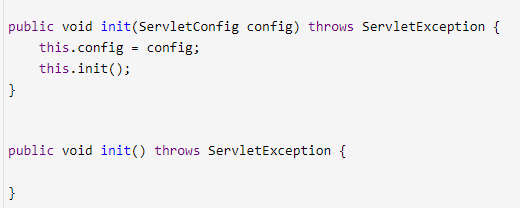
1. **GenericServlet：**一般的Servlet，实现了Servlet和ServletConfig接口



1. **HttpServlet：**基于HTTP协议的实现类

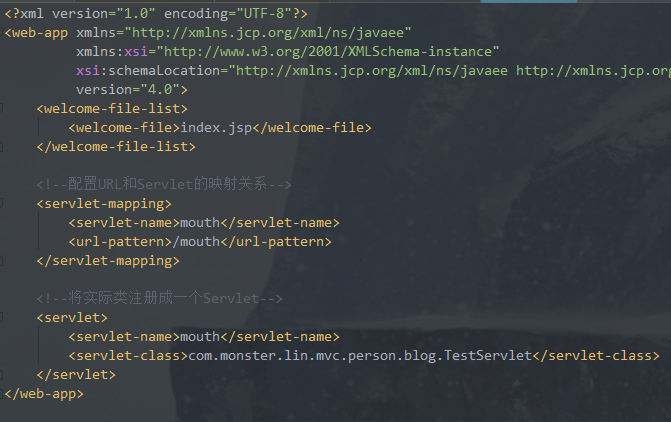


### 2.3 GenericServlet中init方法



1. 为了方便能够在其他地方也能直接使用ServletConfig对象，而不仅仅局限在init(ServletConfig config)方法中，所以创建一个私有的成员变量config，在init(ServletConfig config)方法中就将其赋值给config，这样一来，GenericServlet和其子类都可以调用其getServletConfig()方法来获取ServletConfig对象了。
2. 空的init()，实际上就是为了后续的扩展和重写，有需要的情况下去覆盖init()而不是去覆盖init(ServletConfig config)，因为后者一旦覆盖，就无法通过上述的方法在其他地方便捷地调用getServletConfig方法获取ServletConfig对象了。

## 注册Servlet



1. 详细见web.xml文件

# Servlet的生命周期

## Servlet接口

Servlet从加载到使用发生的一系列事件，称之为生命周期事件，通过生命周期方法来体现。生命周期核心方法有：init(),service(),destroy()。每个Servlet都会实现这些方法，并且在特定的运行时间调用它们。

### 1.1 初始化方法init()

public void init(ServletConfig config) throws ServletException;

1. 在Servlet生命周期的初始化阶段，web容器通过调用init()方法来初始化Servlet实例，并且可以传递一个实现 javax.servlet.ServletConfig 接口的对象给它。
2. 配置对象（configuration object）使Servlet能够读取在web应用的web.xml文件里定义的名值（name-value）初始参数。
3. 这个方法在Servlet实例的生命周期里只调用一次。

### 1.2 处理方法service()

public void service(ServletRequest req, ServletResponse res)

throws ServletException, IOException;

1. 初始化后，Servlet实例就可以处理客户端请求了。web容器调用Servlet的service()方法来处理每一个请求。service()方法定义了能够处理的请求类型并且调用适当方法来处理这些请求。
2. 如果发出一个Servlet没实现的请求，那么父类的方法就会被调用并且通常会给请求方（requester）返回一个错误信息。
3. 处理HTTP请求流程

* 自定义Servlet继承HttpServlet。
* 请求ServletRequest进入到HttpServlet中，判断是否为HttpServletRequest。
* 判断HttpServletRequest中的方法method，转发到对应的doGet(),doPost()方法中。

### 1.3 销毁方法destory()

public void destroy();

1. 最后，web容器调用destroy()方法来终结Servlet。如果你想在Servlet的生命周期内关闭或者销毁一些文件系统或者网络资源，你可以调用这个方法来实现。
2. destroy() 方法和init()方法一样，在Servlet的生命周期里只能调用一次。

# 注解注册Servlet

## 注解@WebServlet



1. 利用注解代替xml文件注册Servlet，建立Servlet和Url的映射。

# 处理带有参数的Http请求

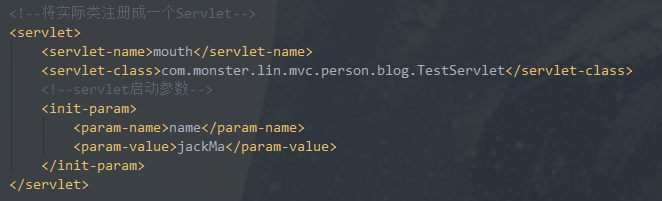
## 方法getParameter()



1. request利用getParameter方法可以获取请求中的参数
2. respose利用writer将返回内容写入outstream

# XML配置servlet启动参数

## 配置



## 在Servlet中使用

方法getServletConfig().getInitParameter("name");

C:\Users\MONSTE~1\AppData\Local\Temp\1548841040(1).png