# 第一个servlet程序

**servlet规范是一个什么规范？**

遵循servlet规范的webapp可以放在不同服务器中，因为他遵循了servlet规范。

**servlet规范包括哪些内容？**

包括：规范了哪些接口、规范了哪些类、规范了一个webapp中应该有哪些配置文件、规范了一个webapp中配置文件的名字、配置文件存放路径、配置文件的内容等，以及一个合法的webapp的结构是怎样的。

## （重点）开发一个带有servlet的webapp

**开发步骤是什么样的？**

1. 在webapps目录下新建一个目录，比如crm（就是webapp的名字）。

注意，接下来crm文件夹就是这个webapp的根。

1. 在webapp的根目录下新建一个目录WEB-INF。

注意，这个名字必须放在webapp根目录并且全部大写。

1. 在WEB-INF下面新建一个目录classes

这个目录必须全部小写，也是servlet规定的，这个目录下存放的一定是编译后的java class文件

1. 在WEB-INF目录下建立一个lib目录

注意：这个目录不是必须的，但是如果webapp需要第三方jar包的话，都放在这里。

这个目录必须全部小写。

例如，连接数据库需要的jar包，也在这个目录下面。

1. 在WEB-INF下面新建一个web.xml。

注意：这个文件是必须的，并且必须叫web.xml。

这个配置文件中描述了请求路径和servlet类之间的对照关系。

这个文件最好从其他webapp中复制。

1. 编写一个java程序，这个小java程序也不能随意开发，而是必须实现servlet接口。

这个servlet接口不在jdk中，而是Oracle提供的，servlet接口是JavaEE规范中的一个，tomcat实现了servlet规范，所以也需要使用servlet接口。

1. 编译Servlet程序。

重点：

1. 首先要确保实现了Servlet接口。
2. 如何让HelloServlet.java编译通过？ 答：配置CLASSPATH（配置新的变量）：

.;D:\code\_space\environments\apache-tomcat-10.0.27\lib\servlet-api.jar

思考：以上配置的CLASSPATH和tomcat服务器的运行有无关系？

没有，这个只是为了让HelloServlet.java可以正常编译，tomcat本身是可以找到这个jar包的。

1. 将以上编译的.class文件复制到WEB-INF/classes目录下即可。
2. 编写配置文件web.xml，让请求路径和servlet类名关联在一起（注册servlet类）。

[演示代码](#_添加配置信息后的web.xml)

1. 启动tomcat服务器。
2. 浏览器输入url：

浏览器的请求路径必须是http://IP地址:端口号/项目名/web.xml中url-pattern标签下的路径。

浏览器上的请求路径和web.xml中url-pattern中的唯一区别就是浏览器上的请求路径带有项目名

另外：浏览器上写的路径太复杂，可以使用超链接，具体方法就是在WEB-INF外部创建html文件，通过a标签链接到浏览器请求路径。

*JavaEE最高版本是8，后来被Oracle捐赠给了Apache，改名JakartaEE，所以本来应该叫做JavaEE9的（现在应该叫JakartaEE9）servlet不再是·javax.servlet.Servlet·而是被改成了·jakarta.servlet.Servlet· 。*

注意：

1. .java的源文件写在哪里都可以，但是.class文件必须在WEB-INF的classes文件夹下面。
2. 新的Servlet包的路径是：**jakarta.servlet.Servlet**。

# 相关内容

## web.xml基本内容

下面是web.xml文件中的最基本内容（从默认webapp ROOT中复制来的）：

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="https://jakarta.ee/xml/ns/jakartaee"

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

xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee

https://jakarta.ee/xml/ns/jakartaee/web-app\_5\_0.xsd"

version="5.0"

metadata-complete="true">

</web-app>

## 添加配置信息后的web.xml

添加配置信息后的xml文件：

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="https://jakarta.ee/xml/ns/jakartaee"

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

xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee

https://jakarta.ee/xml/ns/jakartaee/web-app\_5\_0.xsd"

version="5.0"

metadata-complete="true">

<!--

servlet描述信息

任何一个servlet都对应一个servlet-mapping

-->

<servlet>

<servlet-name>helloServlet</servlet-name>

<!-- 这个位置必须是带有包名的全限定类名 -->

<servlet-class>com.tian.servlet.HelloServlet</servlet-class>

</servlet>

<!-- servlet映射信息 -->

<servlet-mapping>

<!-- 要确保和前面的servlet-name是一样的 -->

<servlet-name>helloServlet</servlet-name>

<!--

这里需要一个路径

要求是：必须以 / 开始

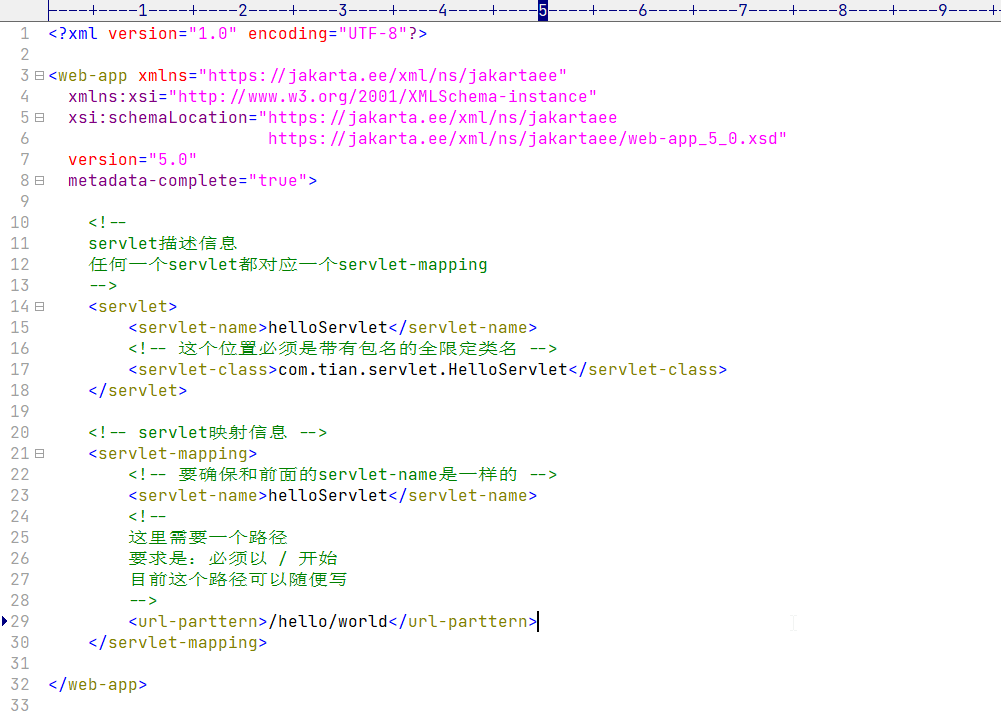
目前这个路径可以随便写

-->

<url-pattern>/asd/qwe</url-pattern>

</servlet-mapping>

</web-app>



## 解决tomcat服务器显示乱码的问题

找到conf目录下的logging,properties文件。

找到java.util.logging.ConsoleHandler.encoding = UTF-8

改为BGK即可。

## 代码附件（整个根文件夹）

