

## 关于servlet的doXXX和service方法

- 1.servlet的service方法可以不需要重写, 他会自动根据请求去分配get|post方法
- 2.在get|post和service方法同时存在的时候,请求不会进入get和post方法,只会进入service方法,如果需要进入post|get方法,那么需要调用super.service()方法

滨岫溪 17, 2020 5:21:59 涓嬧嶰 org.apache.catalina.deploy.WebXml setVersion

璑一櫟: Unknown version string [4.0]. Default version will be used.

[2020-02-17 05:21:59,435] Artifact Servlet\_2.17:war exploded: Artifact is deployed successfully

[2020-02-17 05:21:59,435] Artifact Servlet\_2.17:war exploded: Deploy took 411 milliseconds

我是test的构造方法

我被初始化了

我是服务方法

滨岫溪 17, 2020 5:22:08 涓嬧嶰 org.apache.catalina.startup.HostConfig deployDirectory

淇℃佻: Deploying web application directory D:\apache-tomcat-7.0.96\apache-tomcat-7.0.96\webapps\manager

滨岫溪 17, 2020 5:22:08 涓嬧嶰 org.apache.catalina.startup.HostConfig deployDirectory

淇℃佻: Deployment of web application directory D:\apache-tomcat-7.0.96\apache-tomcat-7.0.96\webapps\manager has

```
public class TestServlet extends javax.servlet.http.HttpServlet {  
    public TestServlet(){  
        System.out.println("我是test的构造方法");  
    }  
    protected void doPost(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response)  
    {  
        System.out.println("我是post方法");  
    }  
    protected void doGet(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response)  
    {  
        System.out.println("我是get方法");  
    }  
    @Override  
    protected void service(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
        System.out.println("我是服务方法");  
    }  
    @Override  
    public void init() throws ServletException {  
        System.out.println("我被初始化了");  
    }  
}
```

- 3.如果重写了service方法,并且需要根据请求方式不同做不同的事情,那么你需要写super.service()方法

我是test的构造方法

我被初始化了

我是服务方法

我是post方法

滨岫溪 17, 2020 5:25:34 涓嬧嶰 org.apache.catalina.startup.HostConfig deployDirectory

淇℃佻: Deploying web application directory D:\apache-tomcat-7.0.96\apache-tomcat-7.0.96\webapps\manager

滨岫溪 17, 2020 5:25:34 涓嬧嶰 org.apache.catalina.startup.HostConfig deployDirectory

淇℃佻: Deployment of web application directory D:\apache-tomcat-7.0.96\apache-tomcat-7.0.96\webapps\manager has

```

public TestServlet() {
    System.out.println("我是test的构造方法");
}

protected void doPost(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) throws ServletException, IOException {
    System.out.println("我是post方法");
}

protected void doGet(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) throws ServletException, IOException {
    System.out.println("我是get方法");
}

@Override
protected void service(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    System.out.println("我是服务方法");
    super.service(req, resp);
}

```



#### 4. 服务器重启也会调用destory方法

```

滨岫溪 17, 2020 5:13:20 涓嬧殑嶰 org.apache.catalina.core.StandardServer a
淇℃佻: A valid shutdown command was received via the shutdown port. St
滨岫溪 17, 2020 5:13:20 涓嬧殑嶰 org.apache.coyote.AbstractProtocol pause
淇℃佻: Pausing ProtocolHandler ["http-bio-80"]
滨岫溪 17, 2020 5:13:20 涓嬧殑嶰 org.apache.coyote.AbstractProtocol pause
淇℃佻: Pausing ProtocolHandler ["ajp-bio-8009"]
我被摧毁了
滨岫溪 17, 2020 5:13:20 涓嬧殑嶰 org.apache.catalina.core.StandardService
淇℃佻: Stopping service Catalina
滨岫溪 17, 2020 5:13:20 涓嬧殑嶰 org.apache.coyote.AbstractProtocol stop

```

#### 5. 第一次进入servlet会调用init方法,后续不会调用init但是会调用相对应的get或post方法

```

[2020-02-17 05:13:49,074] Artifact Servlet_2.17:war exploded: Artifact is deployed successfully
[2020-02-17 05:13:49,074] Artifact Servlet_2.17:war exploded: Deploy took 421 milliseconds
滨岫溪 17, 2020 5:13:58 涓嬧殑嶰 org.apache.catalina.startup.HostConfig deployDirectory
淇℃佻: Deploying web application directory D:\apache-tomcat-7.0.96\apache-tomcat-7.0.96\webapps\
滨岫溪 17, 2020 5:13:58 涓嬧殑嶰 org.apache.catalina.startup.HostConfig deployDirectory
淇℃佻: Deployment of web application directory D:\apache-tomcat-7.0.96\apache-tomcat-7.0.96\webap
我被初始化了
我是post方法
我是post方法

```