

# Spring与Web

- Spring与Web
  - 1. SpringWeb

JavaWeb有三大组件，

- Javaweb三大组件及环境特点

在Java语言范畴内，web层框架都是基于Javaweb基础组件完成的，所以有必要复习一下Javaweb组件的特点

组件	作用	特点
Servlet	服务端小程序，负责接收客户端请求并作出响应的	单例对象，默认第一次访问创建，可以通过配置指定服务器启动就创建，Servlet创建完毕会执行初始化init方法。每个Servlet有一个service方法，每次访问都会执行service方法，但是缺点是一个业务功能就需要配置一个Servlet
Filter	过滤器，负责对客户端请求进行过滤操作的	单例对象，服务器启动时就创建，对象创建完毕执行init方法，对客户端的请求进行过滤，符合要求的放行，不符合要求的直接响应客户端，执行过滤的核心方法doFilter
Listener	监听器，负责对域对象的创建和属性变化进行监听的	根据类型和作用不同，又可分为监听域对象创建销毁和域对象属性内容变化的，根据监听的域不同，又可以分为监听Request域的，监听Session域的，监听ServletContext域的

使用Spring进行Web开发，需要实现以下三点：

- ApplicationContext创建一次，配置类加载一次
- 最好在Web服务器启动时，就执行第一步操作，后续直接从容器中获取Bean使用
- ApplicationContext的引用需要在Web层任何位置都可以获取到

因此，可以按如下思路解决，

- 在ServletContextListener的contextInitialized方法中，执行ApplicationContext的创建，或者，在Servlet的init方法中执行ApplicationContext的创建，并给Servlet的load-on-startup属性一个数字值，确保服务器启动Servlet就创建
- 将创建好的ApplicationContext存储到ServletContext域中，这样整个Web层任何位置就都可以获取到了

## 1. SpringWeb

- 引入坐标

```
<!-- https://mvnrepository.com/artifact/org.springframework/spring-web -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-web</artifactId>
  <version>5.3.25</version>
</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
```

```

<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>javax.servlet-api</artifactId>
  <version>4.0.1</version>
  <scope>provided</scope>
</dependency>

```

- 配置web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
    Copyright 2002 Sun Microsystems, Inc. All rights reserved.
-->
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web
    Application 2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:applicationContext.xml</param-value>
  </context-param>

  <listener>
    <listener-
class>org.springframework.web.context.ContextLoaderListener</listener-class>
    </listener>

</web-app>

```

- 定义Servlet

```

@WebServlet(urlPatterns = "/UserService")
public class UserServiceServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
        //      System.out.println("doGet...");
        ApplicationContext app =
WebApplicationContextUtils.getWebApplicationContext(req.getServletContext());
        UserService userService = app.getBean("userService", UserService.class);
        userService.show();
    }
}

```

但是，目前仍有许多缺陷，

- MVC框架思想及其设计思路

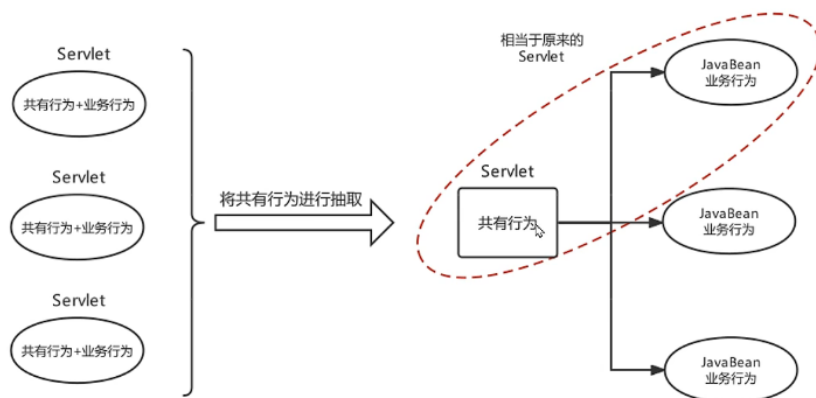
原始Javaweb开发中，Servlet充当Controller的角色，Jsp充当View角色，JavaBean充当模型角色，后期Ajax异步流行后，在加上现在前后端分离开发模式成熟后，View就被原始Html+Vue替代。原始Javaweb开发中，Service充当Controller有很多弊端，显而易见的有如下几个：

Servlet作为Controller的问题	解决思路和方案
每个业务功能请求都对应一个Servlet	根据业务模块去划分Controller
每个Servlet的业务操作太繁琐	将通用的行为，功能进行抽取封装
Servlet获得Spring容器的组件只能通过客户端代码去获取，不能优雅的整合	通过Spring的扩展点，去封装一个框架，从原有的Servlet完全接手过来web层的业务

那么，可以有如下拆解，

- MVC框架思想及其设计思路

负责共有行为的Servlet称之为前端控制器，负责业务行为的JavaBean称之为控制器Controller



分析前端控制器基本功能如下：

- 1、具备可以映射到业务Bean的能力
- 2、具备可以解析请求参数、封装实体等共有功能
- 3、具备响应视图及响应其他数据的功能