# spring-web模块和spring-webmvc模块区别

**spring-web** 和**spring-webmvc** 是 Spring Framework 中两个独立的模块，它们的功能有部分重叠，但也有显著的区别。

## 1. spring-web 模块

**spring-web** 是 Spring Framework 中的一个基础模块，提供了一些通用的 Web 相关功能和工具类。它不依赖于 **MVC** 设计模式，因此不仅仅局限于构建基于 **MVC** 的 Web 应用程序。

**主要功能：**

* **Web 环境支持**：为 Web 应用程序提供通用的功能，如请求、响应处理，[会话管理](https://zhida.zhihu.com/search?content_id=252093124&content_type=Article&match_order=1&q=%E4%BC%9A%E8%AF%9D%E7%AE%A1%E7%90%86&zhida_source=entity" \t "_blank)等。
* **REST 支持**：支持 **RESTful** 风格的 Web 服务（如 @RestController）。
* **HTTP 相关工具类**：提供一些与 HTTP 相关的工具类和接口，帮助[开发者](https://zhida.zhihu.com/search?content_id=252093124&content_type=Article&match_order=1&q=%E5%BC%80%E5%8F%91%E8%80%85&zhida_source=entity" \t "_blank)处理请求和响应。
* **WebSocket 支持**：支持 WebSocket 和 SockJS 协议，适用于现代的实时 Web 应用（例如，聊天、实时推送等）。
* **文件上传**：提供文件上传的支持，尤其是 **Multipart** 文件上传功能。

**主要组件：**

* WebRequest：接口，表示 HTTP 请求，可以用来访问 HTTP 请求的详细信息。
* RequestContextListener 和 RequestContextFilter：用于管理请求和[线程](https://zhida.zhihu.com/search?content_id=252093124&content_type=Article&match_order=1&q=%E7%BA%BF%E7%A8%8B&zhida_source=entity" \t "_blank)的上下文。
* HttpServletRequest 和 HttpServletResponse 的封装和处理工具类。
* **WebSocket** 支持：用于支持实时通信。

**spring-web** 是构建 Web 应用程序的基础设施，但它不包含实现特定架构（如 MVC）的组件。

## 2. spring-webmvc 模块

**spring-webmvc** 是一个 **Spring Web MVC** 模块，专门用于实现 **Model-View-Controller（MVC）设计模式**。它基于 **spring-web**，并在此基础上实现了 Web 应用中的控制器、视图解析等 Web 特定功能。

**主要功能：**

* **MVC 架构支持**：Spring Web MVC 是基于 **MVC** 设计模式，提供了请求-处理-响应的流程，帮助开发者处理 HTTP 请求、[业务逻辑](https://zhida.zhihu.com/search?content_id=252093124&content_type=Article&match_order=1&q=%E4%B8%9A%E5%8A%A1%E9%80%BB%E8%BE%91&zhida_source=entity" \t "_blank)、视图渲染等。
* **DispatcherServlet**：是 Spring Web MVC 的核心，负责请求的分发、控制器的调用和视图的渲染。
* **HandlerMapping 和 HandlerAdapter**：用于处理 URL 到控制器方法的[映射](https://zhida.zhihu.com/search?content_id=252093124&content_type=Article&match_order=1&q=%E6%98%A0%E5%B0%84&zhida_source=entity" \t "_blank)和执行控制器方法。
* **表单处理**：支持表单提交和绑定，帮助将请求参数绑定到 Java 对象中。
* **视图解析**：支持通过不同的视图技术（如 **JSP**、**Thymeleaf**、**FreeMarker** 等）渲染视图。
* **拦截器**：Spring MVC 允许你在请求处理前后执行自定义逻辑，如日志记录、[权限检查](https://zhida.zhihu.com/search?content_id=252093124&content_type=Article&match_order=1&q=%E6%9D%83%E9%99%90%E6%A3%80%E6%9F%A5&zhida_source=entity" \t "_blank)等。
* **数据绑定与验证**：Spring MVC 支持从请求中绑定数据，并支持验证功能。

**主要组件：**

* **DispatcherServlet**：用于接受和分发 HTTP 请求。
* **@Controller** 和 **@RequestMapping**：用于标记控制器和请求处理方法。
* **ModelAndView**：用于封装模型数据和视图信息。
* **ViewResolver**：视图解析器，用于选择和渲染视图。
* **HandlerInterceptor**：拦截器，用于在请求和响应过程中插入额外的处理逻辑。

## 3. 区别总结

|  |  |  |
| --- | --- | --- |
| 特性 | spring-web | spring-webmvc |
| 模块类型 | 基础 Web 功能库 | 专注于 Web 应用的 MVC 组件 |
| 功能范围 | 提供 Web 环境支持（包括 REST、WebSocket、文件上传等） | 提供基于 MVC 的 Web 应用架构支持 |
| MVC 支持 | 不直接支持 MVC | 完全支持 MVC 架构，核心组件如 DispatcherServlet |
| 请求处理 | 提供通用请求和响应处理工具，适用于多种 Web 架构 | 提供请求处理流程（Controller-View） |
| WebSocket 支持 | 支持 WebSocket 和 SockJS | 不支持 WebSocket |
| RESTful 支持 | 支持创建 RESTful Web 服务（如 @RestController） | 支持创建 RESTful 风格的控制器，但重点是 MVC |
| 文件上传支持 | 提供文件上传的通用支持 | 不涉及文件上传的专门功能 |
| 核心组件 | 提供 WebRequest，HttpServletRequest 处理类 | 提供 DispatcherServlet，ModelAndView，HandlerMapping 等 |

## 4. 何时使用哪个模块？

* **使用 spring-web 模块**：
  + 当需要构建一个 **RESTful Web 服务**，而不需要完整的 MVC 架构时，使用 spring-web 是更合适的选择。
  + 当应用程序需要支持 **WebSocket**、**文件上传** 或其他通用 Web 功能时，spring-web 提供了相关的基础设施。
* **使用 spring-webmvc 模块**：
  + 当需要构建一个基于 **Model-View-Controller (MVC)** 模式的传统 Web 应用时，使用 spring-webmvc。
  + 如果应用程序需要完整的 **控制器**、**视图解析**、**表单处理**、[**数据绑定**](https://zhida.zhihu.com/search?content_id=252093124&content_type=Article&match_order=2&q=%E6%95%B0%E6%8D%AE%E7%BB%91%E5%AE%9A&zhida_source=entity) 和 **验证**，Spring Web MVC 是最佳选择。

## 5. 关系

* spring-webmvc 是 **spring-web** 的上层扩展。换句话说，spring-webmvc 依赖于 spring-web 模块。
* **spring-web** 提供了通用的 Web 功能和工具，而 **spring-webmvc** 提供了基于 **MVC** 架构的具体实现。

简而言之，如果你只是需要 Web 相关的一些通用功能，使用 spring-web 足矣；如果你需要构建 Web 应用并且使用 MVC 模式，使用 spring-webmvc。