SpringBoot接口到底支持多少种类型的参数

<https://mp.weixin.qq.com/s?__biz=MzA5MTkxMDQ4MQ==&mid=2648942681&idx=1&sn=eeea9d5d97e1cdd46a63cb1c953b5176&chksm=88623067bf15b97132d85741b2241ca776bee1d46ff2dad373323be806488e7bcb5b602071f9&scene=178&cur_album_id=1873497824336658435#rd>

## 1、来看 2 个好问题

大家在使用 SpringMVC 或者 SpringBoot 开发接口的时候，有没有思考过下面这 2 个问题

* 接口的参数到底支持哪些类型？有什么规律可循么？
* 接口参数的值是从哪里来的呢？

说实话，这 2 个问题非常关键，搞懂原理之后，开发接口将得心应手，今天就带大家从原理上来搞懂这俩问题。

## 2、SpringMVC 处理请求大概的过程

step1、接受请求

step2、根据请求信息找到能够处理请求的控制器方法

step3、解析请求，组装控制器方法需要的参数的值

step4、通过反射调用控制器方法

step5、响应结果等

咱们重点来看 step3 参数值组装这个过程。

## 3、解析处理器方法参数的值

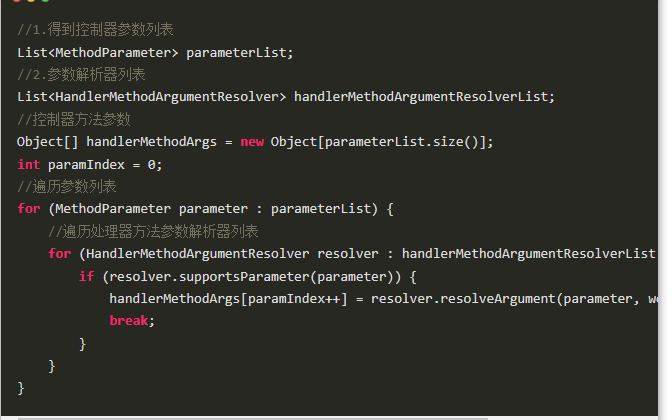
解析参数需要的值，SpringMVC 中专门有个接口来干这个事情，这个接口就是：**HandlerMethodArgumentResolver**，中文称呼：处理器方法参数解析器，说白了就是解析请求得到 Controller 方法的参数的值。

### 3.1、处理器方法参数解析器：HandlerMethodArgumentResolver 接口



### 3.1、解析参数值的过程

SpringMVC 中会配置多个 HandlerMethodArgumentResolver，组成一个 HandlerMethodArgumentResolver 列表，用这个列表来解析参数得到参数需要的值，相当于 2 嵌套 for 循环，简化版的过程如下：

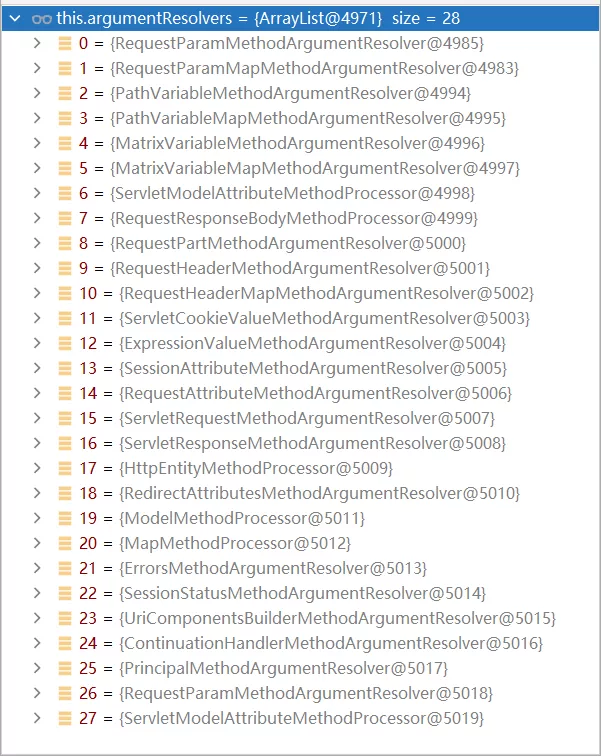


解析参数源码的位置：

org.springframework.web.method.support.InvocableHandlerMethod#getMethodArgumentValues

## 4、常见的 HandlerMethodArgumentResolver

大家可以在InvocableHandlerMethod#getMethodArgumentValues这个位置设置断点，可以详细了解参数解析的过程，debug 中我们可以在这看到 SpringMVC 中默认情况下注册了这么多解析器，如下图：



如下表，列出了一些常见的，以及这些参数解析器能够解析的参数的特点及类型：

| **实现类** | **支持的参数类型** | **参数值** |
| --- | --- | --- |
| RequestParamMethodArgumentResolver | 参数需使用@RequestParam 标注，且 name 属性有值，参数通常为普通类型、Map 类型；或 MultipartFile、Part 类型，或 MultipartFile、Part 这两种类型的集合、数组 | 请求参数 |
| RequestParamMapMethodArgumentResolver | 参数需使用@RequestParam 标注，且 name 属性没有值，参数为 Map 类型；参数的值从 request 的参数中取值，Map 中的 key 对应参数名称，value 对应参数的值 | 请求参数 |
| PathVariableMapMethodArgumentResolver | 参数需使用@PathVariable 标注，参数通常为普通类型 | 从 url 中取值 |
| RequestHeaderMethodArgumentResolver | 参数需使用@RequestHeader 标注，参数通常为 Map、MultiValueMap、HttpHeaders 类型 | 请求头 |
| ServletCookieValueMethodArgumentResolver | 参数需使用@CookieValue 标注，参数为普通类型或者 Cookie 类型 | cookie |
| ModelMethodProcessor | 参数为 Model 类型，控制器中可以调用 model.addAttribute 想模型中放数据，最终这些数据都会通过 request.setAttribute 复制到 request 中 | 来源于 SpringMVC 容器 |
| MapMethodProcessor | 参数为 Map 类型，值同 ModelMethodProcessor | 来源于 SpringMVC 容器 |
| ModelAttributeMethodProcessor | 参数需要使用@ModelAttribute 标注 | Model.getAttribute |
| ServletRequestMethodArgumentResolver | 参数类型为 WebRequest、ServletRequest、MultipartRequest、HttpSession、Principal、InputStream、Reader、HttpMethod、Locale、TimeZone、ZoneId | Servlet 容器中的 request |
| ServletResponseMethodArgumentResolver | 参数类型是 ServletResponse、OutputStream、Writer | Servlet 容器中的 response |
| ModelMethodProcessor | 参数为 org.springframework.ui.Model 类型 | 来源于 SpringMVC 容器 |
| RequestAttributeMethodArgumentResolver | 参数需使用@RequestAttribute | request.getAttribute |
| SessionAttributeMethodArgumentResolver | 参数需使用@SessionAttribute | session.getAttribute |
| ExpressionValueMethodArgumentResolver | 参数需使用@Value 标注 | 从 Spring 配置中取值 |
| ServletModelAttributeMethodProcessor | 支持为我们自定义的 javabean 赋值 | - |
| RequestResponseBodyMethodProcessor | 参数需使用@RequestBody 标注 | http 请求中的 body |
| HttpEntityMethodProcessor | 参数类型为 HttpEntity 或 RequestEntity 类型，这两种类型的参数基本上包含了请求的所有参数信息 | http 请求中的完整信息 |

实现类比较多，就不一一说了，这里教大家一招，让大家学会如何看每种参数解析器的源码，掌握看源码之后，大家把每个实现类的源码过一下，基本上就知道如何使用了，这里以RequestParamMethodArgumentResolver源码为例来做解读。

## 5、RequestParamMethodArgumentResolver 源码解读

### 5.1、supportsParameter 方法：判断支持参数类型

源码如下，挺简单的，大家注意看注释，秒懂

### 5.2、resolveArgument 方法

resolveArgument 方法最终会调用RequestParamMethodArgumentResolver#resolveName方法，代码如下，如果是文件上传的，就获取的是 MultipartFile 对象，否则就是调用request.getParameterValues从参数中取值