

在RestController中获取各种相关信息的方法

内容	获取方法
URL中路径的一部分	<p>首先需要在RequestMapping做映射，之后在方法中可以通过注解使用映射的变量</p> <pre>@GetMapping("/{id}") public TvSeries getOne(@PathVariable("id") Integer id){}</pre> <p>可以写多个</p> <pre>@PutMapping("/{id}/characters/{cId}")</pre> <p>还可以使用正则表达式限制类型（不符合要求会返回4xx的错误信息，表示请求参数有问题）此例子表示id必须是数字：</p> <pre>@PutMapping("/{id:\\d+}")</pre>
POST方法传递过来的JSON	<p>给参数前增加@RequestBody注解，Spring会自动把POST的Request Body部分的JSON转成方法声明的类。如果转化失败会返回4xx错误，表示请求参数有问题。</p> <pre>public Object updateOne(@RequestBody TvSeries tvSeries)</pre>
POST方法传递的是表单数据	<p>首先需要声明传入的是application/x-www-form-urlencoded的格式，可在RequestMapping增加consumes参数：</p> <pre>@PostMapping(value="/tvseries", consumes=MediaType.APPLICATION_FORM_URLENCODED_VALUE)</pre> <p>在方法上增加参数，参数使用@RequestParam注解即可</p> <pre>public Object insert(@RequestParam(value="name", required=false) String name)</pre> <p>凡是可以通过HttpServletRequest.getParameter(String)方法取到的值，包含表单提交的、QueryString附带的，都可以使用@RequestParam注解得到默认是request=true。</p>
QueryString的参数	<p>使用@RequestParam注解，通过参数获得，例如：</p> <pre>public Object query(@RequestParam(value="page", required=false) Integer page)</pre>
Request Header	<p>可以使用@Requestheader注解获取Request的头信息，例如：</p> <pre>public ResultJSON editCompany(@RequestHeader("user-agent") String userAgent)</pre> <p>注意：@RequestHeader后面的头名字不区分大小写，但RequestParam，PathVariable等是区分的。</p> <p>如果RequestHeader后面不写参数，会用后面的变量名替代</p>
获取cookie值	<p>使用@CookieValue注解，和其他类似，除非为了兼容老现有客户端，新API里不建议用cookie。</p>
获取当前的Request Response	<p>直接写参数，例如：</p> <pre>public Object doSomething(HttpServletRequest request, HttpServletResponse response)</pre>
获取当前用户	<p>直接在方法上增加参数，类型为：</p> <p>org.springframework.security.core.Authentication，例如：</p> <pre>public TvSeries deleteOne(Authentication auth)</pre> <p>参数auth内会存储有当前的用户信息。</p>

内容	获取方法
文件上传	<p>首先要设置consumes为multipart/form-data</p> <pre>@PostMapping(value="/files", consumes=MediaType.MULTIPART_FORM_DATA_VALUE)</pre> <p>在方法中写参数：</p> <pre>public Map<String, Object> uploadFile(@RequestParam("file") MultipartFile file)</pre> <p>在方法中可以直接使用MultipartFile中的流保存文件了。</p>