链滴

Spring 参数绑定注解

作者：guobing

原文链接：https://ld246.com/article/1475122776726

来源网站：链滴

许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

handler method 参数绑定常用的注解,我们根据他们处理的Request的不同内容部分分为四类：

- 处理requet uri 部分（这里指uri template中variable，不含queryString部分）的注解： @PathVariable;
- 处理request header部分的注解： @RequestHeader, @CookieValue;
- 处理request body部分的注解：@RequestParam，@RequestBody;
- 处理attribute类型是注解： @SessionAttributes, @ModelAttribute;

1、 @PathVariable

当使用@RequestMapping URI template 样式映射时， 即 someUrl/{paramId}, 这时的paramId可过 @Pathvariable注解绑定它传过来的值到方法的参数上。

```
Controller
@RequestMapping("/owners/{ownerId}")
public class RelativePathUriTemplateController {

  @RequestMapping("/pets/{petId}")
  public void findPet(@PathVariable String ownerId, @PathVariable String petId, Model model
 {
    // implementation omitted
  }
}
```

变量名称不一致，需要在@PathVariable("name")指定uri template中的名称。

2. @RequestParam

- 因为使用request.getParameter()方式获取参数，所以可以处理get 方式中queryString的值，可以处理post方式中 body data的值；

- 用来处理Content-Type: 为 application/x-www-form-urlencoded编码的内容，提交方式GET POST

- 该注解有两个属性： value、required； value用来指定要传入值的id名称，required用来指示数是否必须绑定；

```
@RequestMapping(value = "/{xx}/xxx", method = RequestMethod.POST)
    public String msgCallback(@RequestParam(value = "timestamp", required = false) String t
mestamp,
                              @RequestParam(value = "nonce", required = false) String nonce,
                              @RequestParam(value = "msg_signature", required = false) String msgSig
ature,
                              @PathVariable String revAppId,
                              HttpServletRequest request, HttpServletResponse response) {

        return wxMessageHandlerService.messageHandler(request, timestamp, nonce, msgSigna
ure, revAppId);
    }
```

3. @RequestBody

该注解常用来处理Content-Type: 不是application/x-www-form-urlencoded编码的内容，例如appl

cation/json, application/xml等;

```
@RequestMapping(value = "/something", method = RequestMethod.PUT)
public void handle(@RequestBody String body, Writer writer) throws IOException {
  writer.write(body);
}
```