<http://www.cnblogs.com/may-25/p/5485948.html>

在传统的开发过程中，我们的控制CONTROLL层通常需要转向一个JSP视图；但随着WEB2.0相关技术的崛起，我们很多时候只需要返回数据即可，而不是一个JSP页面。

SPRING MVC3的@ResponseBody使Controller直接返回数据，而不是直接指向具体的视图；同时通过

### MessageConverter和produces（如produces="text/plain;charset=UTF-8"）可以返回各种格式的数据（XML，json，RSS，TEXT，字节流等),本章只介绍最简单的使用；

见代码：

@ResponseBody可以直接返回结果，

### 而ResponseEntity 可以定义返回的HttpHeaders和HttpStatus，见文章最后的两张图

**[java]** [view plain](http://blog.csdn.net/json20080301/article/details/12620547) [copy](http://blog.csdn.net/json20080301/article/details/12620547)

1. @RequestMapping(value="/response", method=RequestMethod.GET)
2. **public class ResponseController {**
4. //http://127.0.0.1:8010/response/annotation
5. @RequestMapping("/annotation")
6. **public @ResponseBody String responseBody() {**
7. **return "The String ResponseBody";**
8. }

11. @RequestMapping("/charset/accept")
12. **public @ResponseBody String responseAcceptHeaderCharset() {**
13. **return "\u3053\u3093\u306b\u3061\u306f\u4e16\u754c\uff01 (\"Hello world!\" in Japanese)";**
14. }
16. //http://127.0.0.1:8010/response/charset/produce
17. @RequestMapping(value="/charset/produce", produces="text/plain;charset=UTF-8")
18. **public @ResponseBody String responseProducesConditionCharset() {**
19. **return "\u3053\u3093\u306b\u3061\u306f\u4e16\u754c\uff01 (\"Hello world!\" in Japanese)";**
20. }
22. //http://127.0.0.1:8010/response/entity/status
23. @RequestMapping("/entity/status")
24. **public ResponseEntity<String> responseEntityStatusCode() {**
25. **return new ResponseEntity<String>("The String ResponseBody with custom status code (403 Forbidden)",**
26. HttpStatus.FORBIDDEN);
27. }
29. //http://127.0.0.1:8010/response/entity/headers
30. @RequestMapping("/entity/headers")
31. **public ResponseEntity<String> responseEntityCustomHeaders() {**
32. HttpHeaders headers = **new HttpHeaders();**
33. headers.setContentType(MediaType.TEXT\_PLAIN);
34. **return new ResponseEntity<String>("The String ResponseBody with custom header Content-Type=text/plain",**
35. headers, HttpStatus.OK);
36. }
38. }