HttpMessageConverter<T>

HttpMessageConverter<T> 将请求信息转化并绑定到处理方法入 参中或将响应结果转为对应类型响应信息 Spring 提供了 两种 方法

* 使 @RequestBody / @ResponseBody 对处方法 进 标注
* 使 HttpEntity<T> / ResponseEntity<T> 作为处方法入参或回 值

当控制器处方法使到 @RequestBody/@ResponseBody 或

HttpEntity<T>/ResponseEntity<T> 时, Spring 先 先 根据 请 求 头 或 响应头Accept 属性 选择 匹配HttpMessageConverter, 进而根据参数类型或泛型类型过滤得到匹 HttpMessageConverter, 找不到HttpMessageConverter 将报错

@RequestBody 和 @ResponseBody 不成对 出 现

1. **注解方式**

写一个表单

<form action=*"testFile"* method=*"post"* enctype=*"multipart/form-data"*>

File：<input type=*"file"* name=*"file"*/><br>

Desc：<input type=*"text"* name=*"desc"*><br>

<input type=*"submit"* value=*"Submit"* />

</form>

响应

@ResponseBody

@RequestMapping("/testFile")

**public** String testFile(@RequestBody String bs) {

System.***out***.println(**new** String(bs));

**return** "Hello";

}

此方法会把文件以字符串的形式打印到控制台，在把Hello返回发哦页面上

**该注解可以直接解析JSON**

@ResponseBody

@RequestMapping("/testJson")

**public** Collection<Employee> testJson() {

**return** employeeDao.getAll();

}

**会直接把这个Employee集合转化为JSON字符串返回到前端**

1. **HttpEntity<T> / ResponseEntity<T>**

可以实现下载的功能，写一个超链接

<a href=*"testDownLoad"*>下载</a>

然后写一个响应方法

@RequestMapping("/testDownLoad")

**public** ResponseEntity<**byte**[]> testDownLoad(HttpSession session) {

**byte**[] body = **null**;

InputStream iStream = session.getServletContext().getResourceAsStream("/files/file.txt");

// 数组初始化，长度为字节流的长度

**try** {

body = **new** **byte**[iStream.available()];

iStream.read(body);

} **catch** (IOException e) {

e.printStackTrace();

}

HttpHeaders headers = **new** HttpHeaders();

// 添加一个下载的响应头

headers.add("Content-Disposition", "attachment;filename=abc.txt");

HttpStatus statusCode = HttpStatus.***OK***;

// 三个参数，第一个为字节数组，下载的内容，第二个为响应头，写上下载的响应头，第三个为？？？

ResponseEntity<**byte**[]> responseEntity = **new** ResponseEntity<**byte**[]>(body, headers, statusCode);

**return** responseEntity;

}