|  |
| --- |
| package com.atguigu.springmvc.test;  import java.io.IOException; import java.io.InputStream; import java.util.Collection; import java.util.Date; import java.util.Locale; import java.util.Map;  import javax.servlet.ServletContext; import javax.servlet.http.HttpSession;  import com.atguigu.springmvc.crud.test.UserNameNotMatchPasswordException; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.context.support.ResourceBundleMessageSource; import org.springframework.http.HttpHeaders; import org.springframework.http.HttpStatus; import org.springframework.http.ResponseEntity; import org.springframework.stereotype.Controller; import org.springframework.web.bind.annotation.ControllerAdvice; import org.springframework.web.bind.annotation.ExceptionHandler; import org.springframework.web.bind.annotation.RequestBody; import org.springframework.web.bind.annotation.RequestMapping; import org.springframework.web.bind.annotation.RequestMethod; import org.springframework.web.bind.annotation.RequestParam; import org.springframework.web.bind.annotation.ResponseBody; import org.springframework.web.bind.annotation.ResponseStatus; import org.springframework.web.multipart.MultipartFile; import org.springframework.web.servlet.ModelAndView;  import com.atguigu.springmvc.crud.dao.EmployeeDao; import com.atguigu.springmvc.crud.entities.Employee;  @Controller public class SpringMVCTest {   @Autowired  private EmployeeDao employeeDao;   @Autowired  private ResourceBundleMessageSource messageSource;           } |

# Springmvc概述

|  |
| --- |
|  |

# SpringMVC的HelloWorld

|  |
| --- |
|  |

# 使用@RequestMapping 映射请求

|  |
| --- |
|  |

# 映射请求参数 & 请求头

|  |
| --- |
|  |

# 处理模型数据

|  |
| --- |
|  |

# 视图和视图解析器

|  |
| --- |
|  |

# RESTTful CRUD

|  |
| --- |
|  |

# SpringMVC表单标签 &处理静态资源

|  |
| --- |
|  |

# 数据转换 & 数据格式化 & 数据校验

## 数据转换

|  |
| --- |
|  |

## 数据格式化

|  |
| --- |
|  |

## 数据校验

|  |
| --- |
|  |

## 错误消息的显示

|  |
| --- |
|  |

# 处理JSON使用HttpMessageConverter

|  |
| --- |
| 1584081185(1) |
|  |

# 国际化

|  |
| --- |
| package com.atguigu.springmvc.test;  import java.io.IOException; import java.io.InputStream; import java.util.Collection; import java.util.Date; import java.util.Locale; import java.util.Map;  import javax.servlet.ServletContext; import javax.servlet.http.HttpSession;  import com.atguigu.springmvc.crud.test.UserNameNotMatchPasswordException; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.context.support.ResourceBundleMessageSource; import org.springframework.http.HttpHeaders; import org.springframework.http.HttpStatus; import org.springframework.http.ResponseEntity; import org.springframework.stereotype.Controller; import org.springframework.web.bind.annotation.ControllerAdvice; import org.springframework.web.bind.annotation.ExceptionHandler; import org.springframework.web.bind.annotation.RequestBody; import org.springframework.web.bind.annotation.RequestMapping; import org.springframework.web.bind.annotation.RequestMethod; import org.springframework.web.bind.annotation.RequestParam; import org.springframework.web.bind.annotation.ResponseBody; import org.springframework.web.bind.annotation.ResponseStatus; import org.springframework.web.multipart.MultipartFile; import org.springframework.web.servlet.ModelAndView;  import com.atguigu.springmvc.crud.dao.EmployeeDao; import com.atguigu.springmvc.crud.entities.Employee;  @Controller public class SpringMVCTest {   @Autowired  private EmployeeDao employeeDao;   @Autowired  private ResourceBundleMessageSource messageSource;   @RequestMapping("/testSimpleMappingExceptionResolver")  public String testSimpleMappingExceptionResolver(@RequestParam("i") int i){  String [] vals = new String[10];  System.***out***.println(vals[i]);  return "success";  }   @RequestMapping(value="/testDefaultHandlerExceptionResolver",method=RequestMethod.***POST***)  public String testDefaultHandlerExceptionResolver(){  System.***out***.println("testDefaultHandlerExceptionResolver...");  return "success";  }   @ResponseStatus(reason="测试",value=HttpStatus.***NOT\_FOUND***)  @RequestMapping("/testResponseStatusExceptionResolver")  public String testResponseStatusExceptionResolver(@RequestParam("i") int i){  if(i == 13){  throw new UserNameNotMatchPasswordException();  }  System.***out***.println("testResponseStatusExceptionResolver...");   return "success";  }  // @ExceptionHandler({RuntimeException.class}) // public ModelAndView handleArithmeticException2(Exception ex){ // System.out.println("[出异常了]: " + ex); // ModelAndView mv = new ModelAndView("error"); // mv.addObject("exception", ex); // return mv; // }   /\*\*  \* 1. 在 @ExceptionHandler 方法的入参中可以加入 Exception 类型的参数, 该参数即对应发生的异常对象  \* 2. @ExceptionHandler 方法的入参中不能传入 Map. 若希望把异常信息传导页面上, 需要使用 ModelAndView 作为返回值  \* 3. @ExceptionHandler 方法标记的异常有优先级的问题.  \* 4. @ControllerAdvice: 如果在当前 Handler 中找不到 @ExceptionHandler 方法来出来当前方法出现的异常,  \* 则将去 @ControllerAdvice 标记的类中查找 @ExceptionHandler 标记的方法来处理异常.  \*/ // @ExceptionHandler({ArithmeticException.class}) // public ModelAndView handleArithmeticException(Exception ex){ // System.out.println("出异常了: " + ex); // ModelAndView mv = new ModelAndView("error"); // mv.addObject("exception", ex); // return mv; // }   @RequestMapping("/testExceptionHandlerExceptionResolver")  public String testExceptionHandlerExceptionResolver(@RequestParam("i") int i){  System.***out***.println("result: " + (10 / i));  return "success";  }    } |

# 文件的上传

|  |
| --- |
|  |

# 使用拦截器

|  |
| --- |
|  |

# 异常处理

|  |
| --- |
|  |

# SpringMVC 运行流程

|  |
| --- |
|  |

# 在 Spring的环境下使用SpringMVC

|  |
| --- |
|  |

# SpringMVC 对比 Struts2

|  |
| --- |
|  |