SSM框架整合

学生 教师 课程 管理系统

**ON DELETE CASCADE 级联删除**

**ON UPDATE CASCADE 级联更新**

CREATE TABLE student (

// s\_id int NOT NULL auto\_increment,

s\_id VARCHAR(50) PRIMARY KEY NOT NULL, //编号主键唯一

s\_name VARCHAR(50) NOT NULL

)

CREATE TABLE course (

c\_id VARCHAR(50) PRIMARY KEY NOT NULL,

c\_name VARCHAR(50) NOT NULL,

**c\_period int NOT NULL,**

**c\_credit double NOT NULL**

**)**

CREATE TABLE teacher (

t\_id VARCHAR(50) PRIMARY KEY NOT NULL,

t\_name VARCHAR(50) NOT NULL

)

s-c 选修

CREATE TABLE selection (

s\_id VARCHAR(50) NOT NULL ,

c\_id VARCHAR(50) NOT NULL,

sel\_year VARCHAR(50) ,

sel\_term int(1) ,

sel\_grade int ,

constraint FK\_sel\_s FOREIGN KEY(s\_id) REFERENCES student(s\_id) **ON DELETE CASCADE ON UPDATE CASCADE**,

constraint FK\_sel\_c FOREIGN KEY(c\_id) REFERENCES course(c\_id) **ON DELETE CASCADE ON UPDATE CASCADE**,

PRIMARY KEY(s\_id,c\_id)

);

t-c 教授

CREATE TABLE instruction (

t\_id VARCHAR(50) NOT NULL ,

c\_id VARCHAR(50) NOT NULL,

constraint FK\_ins\_t FOREIGN KEY(t\_id) REFERENCES teacher(t\_id) **ON DELETE CASCADE ON UPDATE CASCADE**,

constraint FK\_ins\_c FOREIGN KEY(c\_id) REFERENCES course(c\_id) **ON DELETE CASCADE ON UPDATE CASCADE**,

PRIMARY KEY(t\_id,c\_id)

);

ip黑白名单

CREATE TABLE ipfilter(

id INT auto\_increment PRIMARY KEY,

ip VARCHAR(50) NOT NULL,

module VARCHAR(50),

mark INT(1),

ctime TIMESTAMP,

cuser VARCHAR(50),

utime TIMESTAMP,

lable VARCHAR(50)

)

管理员

CREATE TABLE user (

id int auto\_increment PRIMARY KEY,

username VARCHAR(50) NOT NULL,

password VARCHAR(50) NOT NULL

)

//过滤ip,若用户在白名单内，则放行

String ipAddress=IPUtils.getRealIP(request);
log.info("USER IP ADDRESS IS =>"+ipAddress);
if(!StringUtils.isNotBlank(ipAddress))
return false;
ipFilter=new ZwIpFilter();
ipFilter.setModule("sino-bankflow");//模块
ipFilter.setIp(ipAddress);//ip地址
ipFilter.setMark(0);//白名单
List<ZwIpFilter> ips=ipFilterMapper.select(ipFilter);
if(ips.isEmpty()){
response.getWriter().append("<h1 style=\"text-align:center;\">Not allowed!</h1>");
return false;
}
return true;

为空

<trim prefix="(" suffix=")" suffixOverrides="," >

<if test="studentid != null" >

studentid,

</if>

<if test="studentname != null" >

studentname,

</if>

<if test="studentage != null" >

studentage,

</if>

<if test="studentsex != null" >

studentsex,

</if>

<if test="studenttel != null" >

studenttel,

</if>

<if test="classid != null" >

classid,

</if>

</trim>

<trim prefix="values (" suffix=")" suffixOverrides="," >

<if test="studentid != null" >

#{studentid,jdbcType=INTEGER},

</if>

<if test="studentname != null" >

#{studentname,jdbcType=VARCHAR},

</if>

<if test="studentage != null" >

#{studentage,jdbcType=INTEGER},

</if>

<if test="studentsex != null" >

#{studentsex,jdbcType=VARCHAR},

</if>

<if test="studenttel != null" >

#{studenttel,jdbcType=INTEGER},

</if>

<if test="classid != null" >

#{classid,jdbcType=INTEGER},

</if>

</trim>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:aop="http://www.springframework.org/schema/aop"

xmlns:tx="http://www.springframework.org/schema/tx"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop.xsd

http://www.springframework.org/schema/tx

http://www.springframework.org/schema/tx/spring-tx.xsd">

<servlet>

<servlet-mapping>

<beans xmlns="http://www.springframework.org/schema/beans" xmlns:mvc="http://www.springframework.org/schema/mvc" xmlns:context="http://www.springframework.org/schema/context" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=" http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc.xsd http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd">

<?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN" "http://mybatis.org/dtd/mybatis-3-config.dtd">

<dependency>

<groupId>com.github.pagehelper</groupId>

<artifactId>pagehelper</artifactId>

<version>5.1.2</version>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-web</artifactId>

<version>${spring.security.version}</version>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-config</artifactId>

<version>${spring.security.version}</version>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-core</artifactId>

<version>${spring.security.version}</version>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-taglibs</artifactId>

<version>${spring.security.version}</version>

</dependency>

<dependency>

<groupId>javax.annotation</groupId>

<artifactId>jsr250-api</artifactId>

<version>1.0</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context-support</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-beans</artifactId>

<version>${spring.version}</version>

</dependency>

<build>

<pluginManagement>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.2</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

<encoding>UTF-8</encoding>

<showWarnings>true</showWarnings>

</configuration>

</plugin>

</plugins>

</pluginManagement>

</build>

private String id; // 主键

private String productNum; // 编号 唯一

private String productName; // 名称

private String cityName; // 出发城市

private Date departureTime; // 出发时间

private String departureTimeStr;

private double productPrice; // 产品价格

private String productDesc; // 产品描述

private Integer productStatus; // 状态 0 关闭 1 开启

private String productStatusStr;

实现

注册组件到容器中

行内写法

写法可以不用空格

单引号原样输出 会转义 \\n

双引号会解释特殊字符 不会转义 \n

输出换行

<!--标红加版本号-->

*#spring:*

*# mvc:*

*# static-path-pattern: /res/\*\* 这个会导致welcome page功能、*Favicon 功能*失效*

*# resources:*

*# static-locations: [classpath:/haha/]*

*路径变量*

*获取请求头*

*获取请求参数*

*重复的key会被覆盖 skill集合有多个值，存入Map*

*MutiValueMap可以接收重复key*

*import java.io.BufferedInputStream;*

*import java.io.BufferedOutputStream;*

*import java.io.ByteArrayInputStream;*

*import java.io.ByteArrayOutputStream;*

*import java.io.IOException;*

*import java.io.InputStream;*

*import java.io.OutputStream;*

*import java.math.BigDecimal;*

*import java.text.DecimalFormat;*

*import java.text.SimpleDateFormat;*

*import java.util.ArrayList;*

*import java.util.Date;*

*import java.util.Iterator;*

*import java.util.List;*

*import java.util.Map;*

*import javax.servlet.ServletOutputStream;*

*import javax.servlet.http.HttpServletResponse;*

*import org.apache.poi.hssf.usermodel.HSSFCell;*

*import org.apache.poi.hssf.usermodel.HSSFCellStyle;*

*import org.apache.poi.hssf.usermodel.HSSFClientAnchor;*

*import org.apache.poi.hssf.usermodel.HSSFComment;*

*import org.apache.poi.hssf.usermodel.HSSFFont;*

*import org.apache.poi.hssf.usermodel.HSSFPatriarch;*

*import org.apache.poi.hssf.usermodel.HSSFRichTextString;*

*import org.apache.poi.hssf.usermodel.HSSFRow;*

*import org.apache.poi.hssf.usermodel.HSSFSheet;*

*import org.apache.poi.hssf.usermodel.HSSFWorkbook;*

*import org.apache.poi.ss.usermodel.Cell;*

*import org.apache.poi.ss.usermodel.CellStyle;*

*import org.apache.poi.ss.usermodel.Font;*

*import org.apache.poi.ss.usermodel.Row;*

*import org.apache.poi.ss.usermodel.Sheet;*

*import org.apache.poi.ss.usermodel.Workbook;*

*import org.apache.poi.xssf.streaming.SXSSFCell;*

*import org.apache.poi.xssf.streaming.SXSSFRow;*

*import org.apache.poi.xssf.streaming.SXSSFSheet;*

*import org.apache.poi.xssf.streaming.SXSSFWorkbook;*

*import org.apache.poi.xssf.usermodel.XSSFWorkbook;*

*import com.alibaba.fastjson.JSONArray;*

*import com.alibaba.fastjson.JSONObject;*

*public static String NO\_DEFINE = "no\_define";//未定义的字段*

*public static String DEFAULT\_DATE\_PATTERN="yyyy-MM-dd";//默认日期格式*

*public static int DEFAULT\_COLOUMN\_WIDTH = 17;*

*private final static String excel2003L =".xls";    //2003- 版本的excel*

*private final static String excel2007U =".xlsx";   //2007+ 版本的excel*

*/\*\**

*\* Excel导入*

*\*/*

*public static  List<List<Object>> getBankListByExcel(InputStream in, String fileName) throws Exception{*

*List<List<Object>> list = null;*

*//创建Excel工作薄*

*Workbook work = getWorkbook(in,fileName);*

*if(null == work){*

*throw new Exception("创建Excel工作薄为空！");*

*}*

*Sheet sheet = null;*

*Row row = null;*

*Cell cell = null;*

*list = new ArrayList<List<Object>>();*

*//遍历Excel中所有的sheet*

*for (int i = 0; i < work.getNumberOfSheets(); i++) {*

*sheet = work.getSheetAt(i);*

*if(sheet==null){continue;}*

*//遍历当前sheet中的所有行*

*//包涵头部，所以要小于等于最后一列数,这里也可以在初始值加上头部行数，以便跳过头部*

*for (int j = sheet.getFirstRowNum(); j <= sheet.getLastRowNum(); j++) {*

*//读取一行*

*row = sheet.getRow(j);*

*//去掉空行和表头*

*if(row==null||row.getFirstCellNum()==j){continue;}*

*//遍历所有的列*

*List<Object> li = new ArrayList<Object>();*

*for (int y = row.getFirstCellNum(); y < row.getLastCellNum(); y++) {*

*cell = row.getCell(y);*

*li.add(getCellValue(cell));*

*}*

*list.add(li);*

*}*

*}*

*return list;*

*}*

*/\*\**

*\* 描述：根据文件后缀，自适应上传文件的版本*

*\*/*

*public static  Workbook getWorkbook(InputStream inStr,String fileName) throws Exception{*

*Workbook wb = null;*

*String fileType = fileName.substring(fileName.lastIndexOf("."));*

*if(excel2003L.equals(fileType)){*

*wb = new HSSFWorkbook(inStr);  //2003-*

*}else if(excel2007U.equals(fileType)){*

*wb = new XSSFWorkbook(inStr);  //2007+*

*}else{*

*throw new Exception("解析的文件格式有误！");*

*}*

*return wb;*

*}*

*/\*\**

*\* 描述：对表格中数值进行格式化*

*\*/*

*public static  Object getCellValue(Cell cell){*

*Object value = null;*

*DecimalFormat df = new DecimalFormat("0");  //格式化字符类型的数字*

*SimpleDateFormat sdf = new SimpleDateFormat("yyy-MM-dd");  //日期格式化*

*DecimalFormat df2 = new DecimalFormat("0.00");  //格式化数字*

*switch (cell.getCellType()) {*

*case Cell.CELL\_TYPE\_STRING:*

*value = cell.getRichStringCellValue().getString();*

*break;*

*case Cell.CELL\_TYPE\_NUMERIC:*

*if("General".equals(cell.getCellStyle().getDataFormatString())){*

*value = df.format(cell.getNumericCellValue());*

*}else if("m/d/yy".equals(cell.getCellStyle().getDataFormatString())){*

*value = sdf.format(cell.getDateCellValue());*

*}else{*

*value = df2.format(cell.getNumericCellValue());*

*}*

*break;*

*case Cell.CELL\_TYPE\_BOOLEAN:*

*value = cell.getBooleanCellValue();*

*break;*

*case Cell.CELL\_TYPE\_BLANK:*

*value = "";*

*break;*

*default:*

*break;*

*}*

*return value;*

*}*

*/\*\**

*\* 导出Excel 97(.xls)格式 ，少量数据*

*\* @param title 标题行*

*\* @param headMap 属性-列名*

*\* @param jsonArray 数据集*

*\* @param datePattern 日期格式，null则用默认日期格式*

*\* @param colWidth 列宽 默认 至少17个字节*

*\* @param out 输出流*

*\*/*

*public static void exportExcel(Map<String, String> headMap,JSONArray jsonArray,String datePattern,int colWidth, OutputStream out) {*

*if(datePattern==null) datePattern = DEFAULT\_DATE\_PATTERN;*

*// 声明一个工作薄*

*HSSFWorkbook workbook = new HSSFWorkbook();*

*workbook.createInformationProperties();*

*workbook.getDocumentSummaryInformation().setCompany("\*\*\*\*\*公司");*

*//表头样式*

*HSSFCellStyle titleStyle = workbook.createCellStyle();*

*titleStyle.setAlignment(HSSFCellStyle.ALIGN\_CENTER);*

*HSSFFont titleFont = workbook.createFont();*

*titleFont.setFontHeightInPoints((short) 20);*

*titleFont.setBoldweight((short) 700);*

*titleStyle.setFont(titleFont);*

*// 列头样式*

*HSSFCellStyle headerStyle = workbook.createCellStyle();*

*headerStyle.setFillPattern(HSSFCellStyle.SOLID\_FOREGROUND);*

*headerStyle.setFillPattern(HSSFCellStyle.SOLID\_FOREGROUND);*

*headerStyle.setBorderBottom(HSSFCellStyle.BORDER\_THIN);*

*headerStyle.setBorderLeft(HSSFCellStyle.BORDER\_THIN);*

*headerStyle.setBorderRight(HSSFCellStyle.BORDER\_THIN);*

*headerStyle.setBorderTop(HSSFCellStyle.BORDER\_THIN);*

*headerStyle.setAlignment(HSSFCellStyle.ALIGN\_CENTER);*

*HSSFFont headerFont = workbook.createFont();*

*headerFont.setFontHeightInPoints((short) 12);*

*headerFont.setBoldweight(HSSFFont.BOLDWEIGHT\_BOLD);*

*headerStyle.setFont(headerFont);*

*// 单元格样式*

*HSSFCellStyle cellStyle = workbook.createCellStyle();*

*cellStyle.setFillPattern(HSSFCellStyle.SOLID\_FOREGROUND);*

*cellStyle.setBorderBottom(HSSFCellStyle.BORDER\_THIN);*

*cellStyle.setBorderLeft(HSSFCellStyle.BORDER\_THIN);*

*cellStyle.setBorderRight(HSSFCellStyle.BORDER\_THIN);*

*cellStyle.setBorderTop(HSSFCellStyle.BORDER\_THIN);*

*cellStyle.setAlignment(HSSFCellStyle.ALIGN\_CENTER);*

*cellStyle.setVerticalAlignment(HSSFCellStyle.VERTICAL\_CENTER);*

*HSSFFont cellFont = workbook.createFont();*

*cellFont.setBoldweight(HSSFFont.BOLDWEIGHT\_NORMAL);*

*cellStyle.setFont(cellFont);*

*// 生成一个(带标题)表格*

*HSSFSheet sheet = workbook.createSheet();*

*// 声明一个画图的顶级管理器*

*HSSFPatriarch patriarch = sheet.createDrawingPatriarch();*

*// 定义注释的大小和位置,详见文档*

*HSSFComment comment = patriarch.createComment(new HSSFClientAnchor(0,*

*0, 0, 0, (short) 4, 2, (short) 6, 5));*

*// 设置注释内容*

*comment.setString(new HSSFRichTextString("可以在POI中添加注释！"));*

*// 设置注释作者，当鼠标移动到单元格上是可以在状态栏中看到该内容.*

*comment.setAuthor("JACK");*

*//设置列宽*

*int minBytes = colWidth<DEFAULT\_COLOUMN\_WIDTH?DEFAULT\_COLOUMN\_WIDTH:colWidth;//至少字节数*

*int[] arrColWidth = new int[headMap.size()];*

*// 产生表格标题行,以及设置列宽*

*String[] properties = new String[headMap.size()];*

*String[] headers = new String[headMap.size()];*

*int ii = 0;*

*for (Iterator<String> iter = headMap.keySet().iterator(); iter*

*.hasNext();) {*

*String fieldName = iter.next();*

*properties[ii] = fieldName;*

*headers[ii] = fieldName;*

*int bytes = fieldName.getBytes().length;*

*arrColWidth[ii] =  bytes < minBytes ? minBytes : bytes;*

*sheet.setColumnWidth(ii,arrColWidth[ii]\*256);*

*ii++;*

*}*

*// 遍历集合数据，产生数据行*

*int rowIndex = 0;*

*for (Object obj : jsonArray) {*

*if(rowIndex == 65535 || rowIndex == 0){*

*if ( rowIndex != 0 ) sheet = workbook.createSheet();//如果数据超过了，则在第二页显示*

*HSSFRow headerRow = sheet.createRow(0); //列头 rowIndex =0*

*for(int i=0;i<headers.length;i++)*

*{*

*headerRow.createCell(i).setCellValue(headers[i]);*

*headerRow.getCell(i).setCellStyle(headerStyle);*

*}*

*rowIndex = 1;//数据内容从 rowIndex=1开始*

*}*

*JSONObject jo = (JSONObject) JSONObject.toJSON(obj);*

*HSSFRow dataRow = sheet.createRow(rowIndex);*

*for (int i = 0; i < properties.length; i++)*

*{*

*HSSFCell newCell = dataRow.createCell(i);*

*Object o =  jo.get(properties[i]);*

*String cellValue = "";*

*if(o==null) cellValue = "";*

*else if(o instanceof Date) cellValue = new SimpleDateFormat(datePattern).format(o);*

*else cellValue = o.toString();*

*newCell.setCellValue(cellValue);*

*newCell.setCellStyle(cellStyle);*

*}*

*rowIndex++;*

*}*

*// 自动调整宽度*

*/\*for (int i = 0; i < headers.length; i++) {*

*sheet.autoSizeColumn(i);*

*}\*/*

*try {*

*workbook.write(out);*

*workbook.close();*

*} catch (IOException e) {*

*e.printStackTrace();*

*}*

*}*

*/\*\**

*\* 导出Excel 2007 OOXML (.xlsx)格式*

*\* @param title 标题行*

*\* @param headMap 属性-列头*

*\* @param jsonArray 数据集*

*\* @param datePattern 日期格式，传null值则默认 年月日*

*\* @param colWidth 列宽 默认 至少17个字节*

*\* @param out 输出流*

*\*/*

*public static void exportExcelX(Map<String, String> headMap,JSONArray jsonArray,String datePattern,int colWidth, OutputStream out) {*

*if(datePattern==null) datePattern = DEFAULT\_DATE\_PATTERN;*

*// 声明一个工作薄*

*SXSSFWorkbook workbook = new SXSSFWorkbook(1000);//缓存*

*workbook.setCompressTempFiles(true);*

*//表头样式*

*CellStyle titleStyle = workbook.createCellStyle();*

*titleStyle.setAlignment(HSSFCellStyle.ALIGN\_CENTER);*

*Font titleFont = workbook.createFont();*

*titleFont.setFontHeightInPoints((short) 20);*

*titleFont.setBoldweight((short) 700);*

*titleStyle.setFont(titleFont);*

*// 列头样式*

*CellStyle headerStyle = workbook.createCellStyle();*

*headerStyle.setFillPattern(HSSFCellStyle.SOLID\_FOREGROUND);*

*headerStyle.setBorderBottom(HSSFCellStyle.BORDER\_THIN);*

*headerStyle.setBorderLeft(HSSFCellStyle.BORDER\_THIN);*

*headerStyle.setBorderRight(HSSFCellStyle.BORDER\_THIN);*

*headerStyle.setBorderTop(HSSFCellStyle.BORDER\_THIN);*

*headerStyle.setAlignment(HSSFCellStyle.ALIGN\_CENTER);*

*Font headerFont = workbook.createFont();*

*headerFont.setFontHeightInPoints((short) 12);*

*headerFont.setBoldweight(HSSFFont.BOLDWEIGHT\_BOLD);*

*headerStyle.setFont(headerFont);*

*// 单元格样式*

*CellStyle cellStyle = workbook.createCellStyle();*

*cellStyle.setFillPattern(HSSFCellStyle.SOLID\_FOREGROUND);*

*cellStyle.setBorderBottom(HSSFCellStyle.BORDER\_THIN);*

*cellStyle.setBorderLeft(HSSFCellStyle.BORDER\_THIN);*

*cellStyle.setBorderRight(HSSFCellStyle.BORDER\_THIN);*

*cellStyle.setBorderTop(HSSFCellStyle.BORDER\_THIN);*

*cellStyle.setAlignment(HSSFCellStyle.ALIGN\_CENTER);*

*cellStyle.setVerticalAlignment(HSSFCellStyle.VERTICAL\_CENTER);*

*Font cellFont = workbook.createFont();*

*cellFont.setBoldweight(HSSFFont.BOLDWEIGHT\_NORMAL);*

*cellStyle.setFont(cellFont);*

*// 生成一个(带标题)表格*

*SXSSFSheet sheet = workbook.createSheet();*

*//设置列宽*

*int minBytes = colWidth<DEFAULT\_COLOUMN\_WIDTH?DEFAULT\_COLOUMN\_WIDTH:colWidth;//至少字节数*

*int[] arrColWidth = new int[headMap.size()];*

*// 产生表格标题行,以及设置列宽*

*String[] properties = new String[headMap.size()];*

*String[] headers = new String[headMap.size()];*

*int ii = 0;*

*for (Iterator<String> iter = headMap.keySet().iterator(); iter*

*.hasNext();) {*

*String fieldName = iter.next();*

*properties[ii] = fieldName;*

*headers[ii] = headMap.get(fieldName);*

*int bytes = fieldName.getBytes().length;*

*arrColWidth[ii] =  bytes < minBytes ? minBytes : bytes;*

*sheet.setColumnWidth(ii,arrColWidth[ii]\*256);*

*ii++;*

*}*

*// 遍历集合数据，产生数据行*

*int rowIndex = 0;*

*for (Object obj : jsonArray) {*

*if(rowIndex == 65535 || rowIndex == 0){*

*if ( rowIndex != 0 ) sheet = workbook.createSheet();//如果数据超过了，则在第二页显示*

*SXSSFRow headerRow = sheet.createRow(0); //列头 rowIndex =0*

*for(int i=0;i<headers.length;i++)*

*{*

*headerRow.createCell(i).setCellValue(headers[i]);*

*headerRow.getCell(i).setCellStyle(headerStyle);*

*}*

*rowIndex = 1;//数据内容从 rowIndex=1开始*

*}*

*JSONObject jo = (JSONObject) JSONObject.toJSON(obj);*

*SXSSFRow dataRow = sheet.createRow(rowIndex);*

*for (int i = 0; i < properties.length; i++)*

*{*

*SXSSFCell newCell = dataRow.createCell(i);*

*Object o =  jo.get(properties[i]);*

*String cellValue = "";*

*if(o==null) cellValue = "";*

*else if(o instanceof Date) cellValue = new SimpleDateFormat(datePattern).format(o);*

*else if(o instanceof Float || o instanceof Double)*

*cellValue= new BigDecimal(o.toString()).setScale(2,BigDecimal.ROUND\_HALF\_UP).toString();*

*else cellValue = o.toString();*

*newCell.setCellValue(cellValue);*

*newCell.setCellStyle(cellStyle);*

*}*

*rowIndex++;*

*}*

*// 自动调整宽度*

*for (int i = 0; i < headers.length; i++) {*

*sheet.autoSizeColumn(i);*

*}*

*try {*

*workbook.write(out);*

*workbook.close();*

*workbook.dispose();*

*} catch (IOException e) {*

*e.printStackTrace();*

*}*

*}*

*//Web 导出excel*

*public static void downloadExcelFile(String title,Map<String,String> headMap,JSONArray ja,HttpServletResponse response){*

*try {*

*ByteArrayOutputStream os = new ByteArrayOutputStream();*

*ExcellUtil.exportExcelX(headMap,ja,null,0,os);*

*byte[] content = os.toByteArray();*

*InputStream is = new ByteArrayInputStream(content);*

*// 设置response参数，可以打开下载页面*

*response.reset();*

*response.setContentType("application/vnd.openxmlformats-officedocument.spreadsheetml.sheet;charset=utf-8");*

*response.setHeader("Content-Disposition", "attachment;filename="+ new String((title + ".xlsx").getBytes(), "iso-8859-1"));*

*response.setContentLength(content.length);*

*ServletOutputStream outputStream = response.getOutputStream();*

*BufferedInputStream bis = new BufferedInputStream(is);*

*BufferedOutputStream bos = new BufferedOutputStream(outputStream);*

*byte[] buff = new byte[8192];*

*int bytesRead;*

*while (-1 != (bytesRead = bis.read(buff, 0, buff.length))) {*

*bos.write(buff, 0, bytesRead);*

*}*

*bis.close();*

*bos.close();*

*outputStream.flush();*

*outputStream.close();*

*}catch (Exception e) {*

*e.printStackTrace();*

*}*

*}*

*}*

*import java.io.InputStream;*

*import java.text.SimpleDateFormat;*

*import java.util.ArrayList;*

*import java.util.Date;*

*import java.util.Iterator;*

*import java.util.LinkedHashMap;*

*import java.util.List;*

*import java.util.Map;*

*import org.apache.commons.lang3.StringUtils;*

*import org.springframework.beans.factory.annotation.Autowired;*

*import org.springframework.stereotype.Service;*

*import com.city.pms.common.utils.ImportUtil2;*

*import com.city.pms.entity.project.design.designA.DesignA;*

*import com.city.pms.repository.project.design.designA.DesignADao;*

*@Service*

*public class DesignAServiceImpl implements DesignAService {*

*@Autowired*

*private DesignADao designADao;*

*//*

*@Override*

*public DesignA save(String id) {*

*// TODO Auto-generated method stub*

*return designADao.save(id);*

*}*

*//导入*

*@Override*

*public String insertDatasFromExcel(InputStream in, String fileName,Map<String, String> headMap) {*

*try {*

*//首先调用工具类，读取excel文件，封装称实体集合*

*List<List<Object>> listob = ImportUtil2.getListByExcel(in, fileName);*

*//获取动态数组  动态的增加和减少元素  实现了ICollection和IList接口  灵活的设置数组的大小*

*List<DesignA> designA = new ArrayList<DesignA>();*

*//读取第一列，列头信息，第一行*

*List<Object> list = listob.get(0);*

*Iterator<String> keys = headMap.keySet().iterator();*

*//LinkedHashMap保留插入的顺序（key,value）*

*Map<String,Integer> newMap=new LinkedHashMap<String,Integer>();*

*while(keys.hasNext()){*

*String key = keys.next();*

*String name = headMap.get(key);*

*boolean flag=true;*

*for (int i = 0; i < list.size(); i++) {*

*if(StringUtils.contains(name, list.get(i).toString())){*

*newMap.put(key,i);*

*flag=false;*

*break;*

*}*

*}*

*if(flag==true){*

*throw new Exception("Excel表格参数错误，缺少"+name);*

*}*

*}*

*//遍历listob数据，把数据放到List中  从第二行开始*

*for (int i = 1; i < listob.size(); i++) {*

*List<Object> ob = listob.get(i);*

*DesignA design = new DesignA();*

*design.setEid(Integer.parseInt(ob.get(newMap.get("eid")).toString()));*

*design.setSafe(Integer.parseInt(ob.get(newMap.get("safe")).toString()));*

*//object类型转Double类型*

*design.setPrewarning(Double.parseDouble(ob.get(newMap.get("prewarning")).toString()));*

*design.setAlarm(Double.parseDouble(ob.get(newMap.get("alarm")).toString()));*

*design.setVoerflow(Double.parseDouble(ob.get(newMap.get("voerflow")).toString()));*

*Date parse = new SimpleDateFormat("yyyy-MM-dd").parse(ob.get(newMap.get("createDate")).toString());*

*design.setCreateDate(parse);*

*//Add方法用于添加一个元素到当前列表的末尾*

*designA.add(design);*

*}*

*//批量插入*

*designADao.insertDatasFromExcel(designA);*

*} catch (Exception e) {*

*e.printStackTrace();*

*}*

*return null;*

*}*

*}*

*public String admin\_importmembers(HttpServletResponse response,HttpServletRequest request) throws Exception {*

*MultipartHttpServletRequest multipartRequest = (MultipartHttpServletRequest) request;*

*List<List<Object>> listob = null;*

*MultipartFile file = multipartRequest.getFile("file");*

*// employeeMapper.impExcel(file);*

*System.out.println((POIservice==null)+"--------");*

*POIservice.impExcel(file);*

*System.out.println("............");*

*return "redirect:/emps";*

*}*

*for (int i = 1; i < listob.size(); i++) {*

*List<Object> lo = listob.get(i);*

*Employee employee = new Employee();*

*String id = String.valueOf(lo.get(0));*

*employee.setId(Integer.parseInt(id));*

*employee.setLastName(String.valueOf(lo.get(1)));*

*employee.setGender(String.valueOf(lo.get(2)));*

*employee.setEmail(String.valueOf(lo.get(3)));*

*impusers.add(employee);*

*}*

*System.out.println("impusersSize:"+impusers.size());*

*System.out.println(impusers.toString());*

*employeeMapper.impusers(impusers);*

*<insert id="impusers" parameterType="java.util.List">*

*insert into tbl\_employee(id,last\_name,gender,email)*

*values*

*<foreach collection="list" item="item" index="index" separator=",">*

*(#{item.id},#{item.lastName},#{item.gender},#{item.email})*

*</foreach>*

*</insert>*