报表的批量导出

1. 创建一个webbook，对应一个Excel文件
2. 在webbook中添加一个sheet，对应Excel文件中的sheet
3. 在sheet中添加表头第0行,注意老版本poi对Excel的行数列数有限制short
4. 创建单元格，并设置值表头 设置表头居中
5. 写入实体数据 实际应用中这些数据从数据库得到

一．定义了Excel的格式：

|  |
| --- |
| package com.hand.hap.export.excel;  import java.text.SimpleDateFormat;  import java.util.List;  import org.apache.poi.hssf.usermodel.HSSFCell;  import org.apache.poi.hssf.usermodel.HSSFCellStyle;  import org.apache.poi.hssf.usermodel.HSSFRow;  import org.apache.poi.hssf.usermodel.HSSFSheet;  import org.apache.poi.hssf.usermodel.HSSFWorkbook;  import com.hand.hap.forecast.dto.ForecastOrderDO;  /\*\*  \* 销售预测明细报表  \* @author sudong  \*  \*/  public class GenerateForecastOrderExcel {  //从数据库中查到LIST的数据  public HSSFWorkbook exportExcel(List<ForecastOrderDO> list)  throws Exception {  // 第一步，创建一个webbook，对应一个Excel文件  HSSFWorkbook wb = new HSSFWorkbook();  // 第二步，在webbook中添加一个sheet,对应Excel文件中的sheet  HSSFSheet sheet = wb.createSheet("销售预测明细表");  /\*\*  \* 设置默认单元格宽度.  \*/  sheet.setDefaultColumnWidth((short)30);  // 第三步，在sheet中添加表头第0行,注意老版本poi对Excel的行数列数有限制short  HSSFRow row = sheet.createRow((int) 0);  // 第四步，创建单元格，并设置值表头 设置表头居中  HSSFCellStyle style = wb.createCellStyle();  style.setAlignment(HSSFCellStyle.ALIGN\_CENTER); // 创建一个居中格式  HSSFCell cell = row.createCell((short) 0);  cell.setCellValue("序号");  cell.setCellStyle(style);  cell = row.createCell((short) 1);  cell.setCellValue("预测订单编号");  cell.setCellStyle(style);  cell = row.createCell((short) 2);  cell.setCellValue("订单状态");  cell.setCellStyle(style);  cell = row.createCell((short) 3);  cell.setCellValue("包装厂名称");  cell.setCellStyle(style);    cell = row.createCell((short) 4);  cell.setCellValue("终端客户名称");  cell.setCellStyle(style);    cell = row.createCell((short) 5);  cell.setCellValue("行编号");  cell.setCellStyle(style);    cell = row.createCell((short) 6);  cell.setCellValue("预测起始日期");  cell.setCellStyle(style);    cell = row.createCell((short) 7);  cell.setCellValue("预测终止日期");  cell.setCellStyle(style);    cell = row.createCell((short) 8);  cell.setCellValue("成本系数");  cell.setCellStyle(style);    cell = row.createCell((short) 9);  cell.setCellValue("融资金额");  cell.setCellStyle(style);    cell = row.createCell((short) 10);  cell.setCellValue("行状态");  cell.setCellStyle(style);    cell = row.createCell((short) 11);  cell.setCellValue("创建时间");  cell.setCellStyle(style);    cell = row.createCell((short) 12);  cell.setCellValue("创建人");  cell.setCellStyle(style);    cell = row.createCell((short) 13);  cell.setCellValue("备注");  cell.setCellStyle(style);    SimpleDateFormat df=new SimpleDateFormat("yyyy-MM-dd");  // 第五步，写入实体数据 实际应用中这些数据从数据库得到，  for (int i = 0; i < list.size(); i++) {  row = sheet.createRow((int) i + 1);  ForecastOrderDO forecastOrder = (ForecastOrderDO) list.get(i);  // 第四步，创建单元格，并设置值  row.createCell((short) 0).setCellValue(i+1);  row.createCell((short) 1).setCellValue(forecastOrder.getForecastOrderNum());  row.createCell((short) 2).setCellValue(forecastOrder.getForecastOrderState());  row.createCell((short) 3).setCellValue(forecastOrder.getCompanyName());  row.createCell((short) 4).setCellValue(forecastOrder.getCustomerName());  row.createCell((short) 5).setCellValue(forecastOrder.getLineNum());  //时间形式要判空  if(forecastOrder.getForecastStartDate()!=null){  String forecastDate=df.format(forecastOrder.getForecastStartDate());  row.createCell((short) 6).setCellValue(forecastDate);  }else{  row.createCell((short) 6).setCellValue("");  }  //时间形式要判空  if(forecastOrder.getForecastEndDate()!=null){  String forecastDate=df.format(forecastOrder.getForecastEndDate());  row.createCell((short) 7).setCellValue(forecastDate);  }else{  row.createCell((short) 7).setCellValue("");  }  row.createCell((short) 8).setCellValue(forecastOrder.getCostRadio());    if(forecastOrder.getFinancingAmount()!=null){  row.createCell((short) 9).setCellValue(forecastOrder.getFinancingAmount().toString());  }else{  row.createCell((short) 9).setCellValue("");  }  row.createCell((short) 10).setCellValue(forecastOrder.getForecastOrderLineState());    if(forecastOrder.getCreationDate()!=null){  row.createCell((short) 11).setCellValue(forecastOrder.getCreationDate());  }else{  row.createCell((short) 11).setCellValue("");  }  row.createCell((short) 12).setCellValue(forecastOrder.getUserName());  row.createCell((short) 13).setCellValue(forecastOrder.getRemark());  }  return wb;  }  } |

1. 在service中调用上面定义好的类：

|  |
| --- |
| /\*\*  \* 导出报表  \*/  @Override  **public** HSSFWorkbook export(List<ForecastOrderDO> list) **throws** Exception {  GenerateForecastOrderExcel ge=**new** GenerateForecastOrderExcel();  **return** ge.exportExcel(list);  } |

1. 在controller中调用service：

|  |
| --- |
| /\*\*  \* 导出excel报表  \* **@param** request  \* **@param** response  \* **@param** json  \* **@throws** Exception  \*/  @RequestMapping(value = "bathExport", produces = "application/json; charset=utf-8")  @ResponseBody  **public** **void** exportExcel(HttpServletRequest request,  HttpServletResponse response,String json,Integer userId)  **throws** Exception {    @SuppressWarnings("static-access")  JSONObject obj=**new** JSONObject().*fromObject*(json);  Object startDate=**null**,endDate=**null**;  Date forecastStartDate=**null**,forecastEndDate=**null**;  **if**(obj!=**null**){  startDate=obj.get("forecastStartDate");  endDate=obj.get("forecastEndDate");  **if**(startDate!=**null**){  forecastStartDate=DateUtil.*strToDate*(startDate.toString());  }    **if**(endDate!=**null**){  forecastEndDate=DateUtil.*strToDate*(endDate.toString());  }  }  //将json转换成bean  ForecastOrderDO forecastOrder = (ForecastOrderDO) JSONObject.*toBean*(obj,ForecastOrderDO.**class**);  **if**(userId!=**null**){  forecastOrder.setUserId(userId);  }  forecastOrder.setForecastEndDate(forecastEndDate);  forecastOrder.setForecastStartDate(forecastStartDate);  forecastOrder.setPaging("N");  String fileName="销售预测明细表.xls";  List<ForecastOrderDO> list=iForecastOrderService.generateForecastReport(forecastOrder);  HSSFWorkbook wb = iForecastOrderService.export(list);  response.setContentType("application/vnd.ms-excel");  response.setHeader("Content-disposition",  "attachment;filename="+**new** String(fileName.getBytes("gb2312"), "ISO8859-1"));  OutputStream ouputStream = response.getOutputStream();  wb.write(ouputStream);  ouputStream.flush();  ouputStream.close();  } |