

第9章 PDF报表

目标

- 掌握iText实现导出预约订单信息
- 掌握JasperReport的模板设计器Jaspersoft Studio
- 掌握JasperReport导出PDF报表

1. 常见的PDF报表生成方式

【目标】

- 1: PDF介绍
- 2: 了解iText
- 3: 了解JasperReport

【路径】

- 1: PDF介绍
- 2: iText报表的使用（需要的坐标）
- 3: JasperReports报表的使用（需要的坐标）

【讲解】

1.1 PDF介绍

PDF（Portable Document Format的简称，意为“可携带文档格式”），是由Adobe Systems用于与应用程序、操作系统、硬件无关的方式进行文件交换所发展出的文件格式。PDF文件以PostScript语言图象模型为基础，无论在何种打印机上都可保证精确的颜色和准确的打印效果，即PDF会忠实地再现原稿的每一个字符、颜色以及图象。

PDF种文件格式与操作系统平台无关，也就是说，PDF文件不管是在Windows，Unix还是在苹果公司的Mac OS操作系统中都是通用的。这一特点使它成为在Internet上进行电子文档发行和数字化信息传播的理想文档格式。越来越多的电子图书、产品说明、公司文告、网络资料、电子邮件在开始使用PDF格式文件。

1.2. iText

iText是著名的开源码的站点sourceforge一个项目，是用于生成PDF文档的一个java类库。通过iText不仅可以生成PDF或rtf的文档，而且可以将XML、Html文件转化为PDF文件。iText的安装非常方便，只要引入它的依赖坐标，在程序中就可以使用iText类库了。

maven坐标：

```
1 <!--导入Itext报表-->
2 <dependency>
3   <groupId>com.lowagie</groupId>
4   <artifactId>itext</artifactId>
5   <version>2.1.7</version>
6 </dependency>
```

1.3. JasperReports

JasperReports是一个强大、灵活的报表生成工具，通过预先设计好模板再往模板填充数据的方式来生成报表。通过模板设计，使得可以在上面展示丰富的页面内容，并将之转换成PDF，HTML，或者XML格式。该库完全由Java写成，可以用于在各种Java应用程序，包括J2EE，Web应用程序中生成动态内容。一般情况下，JasperReports会结合Jaspersoft Studio(模板设计器)使用导出PDF报表。

在maven工程中，需引入它的坐标：

```
1 <dependency>
2   <groupId>net.sf.jasperreports</groupId>
3   <artifactId>jasperreports</artifactId>
4   <version>6.8.0</version>
5 </dependency>
```

【小结】

1. PDF,是一种文档的存储格式，一般用归档，公司企业比较流行
2. 2种生成方式
 - iText, 原来的api方式来生成pdf, 引入依赖
 - jaspersReports, 设计模板，再给模板填充数据，引入依赖

2. IText报表使用

【目标】

- 1: iText报表入门
- 2: iText报表整合项目，导出订单详情

【路径】

- 1: iText报表入门案例
 - 在项目 health_common 引入 Itext jar支持
 - 在health_common中创建测试类，使用Itext api生成 PDF报表 (参考资料中的文档)
 - iText报表中解决中文问题
- 2: IText报表整合项目
 - 在页面 orderSuccess.html提供 pdf导出按钮
 - pdf导出按钮 绑定事件提交请求，传递订单ID
 - 在OrderMobileController中添加exportSetmealInfo的方法，接收订单id
 - 通过id查询订单详情，修改OrderDao映射文件，添加查询套餐id列
 - 获取套餐的id, 再查询套餐详情
 - 创建文档，设置保存输出流，response设置内容体格式，头信息下载文件。

- 打开文档，添加内容（预约信息(段落)、套餐详情(Table))
- 根据iText的API，添加表格的样式
- 关闭文档

【讲解】

2.1. iText报表入门案例

1: 在项目 health_parent 引入 itext.jar支持 版本管理

```

1  <dependencyManagement>
2  <dependencies>....
3  <!--导入iText报表-->
4  <dependency>
5      <groupId>com.lowagie</groupId>
6      <artifactId>itext</artifactId>
7      <version>2.1.7</version>
8  </dependency>
9      ...
10 </dependencies>
11 </dependencyManagement>

```

在项目health_common引入itext.jar支持

```

1  <!-- 导入iText报表 -->
2  <dependency>
3      <groupId>com.lowagie</groupId>
4      <artifactId>itext</artifactId>
5  </dependency>

```

生成的jar包



2: 参考资料\iText目录下的书籍：《[iText实战(第2版)].(iText.in.Action).Bruno.Lowagie.文字版.pdf》

生成PDF五步



测试：在health_common中创建测试类TestItext.java，使用itext 生成 PDF报表

示例代码:

```

1  package com.itheima.test;
2
3  import com.lowagie.text.Document;
4  import com.lowagie.text.Paragraph;
5  import com.lowagie.text.pdf.PdfWriter;
6  import org.junit.Test;
7
8  import java.io.File;
9  import java.io.FileOutputStream;
10
11 /**
12  * Description: No Description
13  * User: Eric

```

```

14  */
15  public class TestItext {
16
17      @Test
18      public void testItext() throws Exception {
19          // 创建文件对象
20          Document doc = new Document();
21          // 设置文件存储
22          PdfWriter.getInstance(doc, new FileOutputStream(new
File("d:\\iText.pdf")));
23          // 打开文档
24          doc.open();
25          doc.add(new Paragraph("Hello world"));
26          // 关闭文档
27          doc.close();
28      }
29  }
30

```

这里注意：中文是无法生成到pdf的。

3: iText报表解决中文问题

需要设置字体（设置可以支持中文的字库【操作系统】，【导入itext-asian的jar包】）

```

1  ## 把 资料\iText\中文乱码\中的iTextAsian.jar 安装到仓库中
2  mvn install:install-file -Dfile=E:\iTextAsian.jar -DgroupId=com.alpha -
DartifactId=itextasian -Dversion=1.0.0 -Dpackaging=jar
3
4  ## -Dfile=E:\iTextAsian.jar iTextAsian 不要有中文空格
5  ## idea 中的maven仓库索引更新一下

```

坐标:

```

1  health_common pom中
2  <!-- 导入iText报表，支持中文 -->
3  <dependency>
4      <groupId>com.alpha</groupId>
5      <artifactId>itextasian</artifactId>
6      <version>1.0.0</version>
7  </dependency>

```

```

1      @Test
2      public void testItext() throws Exception {
3          // 创建文件对象
4          Document doc = new Document();
5          // 设置文件存储
6          PdfWriter.getInstance(doc, new FileOutputStream(new
File("d:\\iText.pdf")));
7          // 打开文档
8          doc.open();
9          // 添加段落
10         BaseFont bfChinese = BaseFont.createFont("STSong-Light", "UniGB-
UCS2-H", BaseFont.NOT_EMBEDDED);

```

```

11         doc.add(new Paragraph("你好，传智播客", new Font(bfChinese)));
12         //doc.add(new Paragraph("Hello 传智播客"));
13         // 关闭文档
14         doc.close();
15     }
16

```

其中：STSong-Light和UniGB-UCS2-H对应iTextAsian.jar包的内容



其中，“STSong-Light”定义了使用的中文字体，iTextAsian.jar类库中提供了几个可供使用的字体，都是以properties结尾的文件。“UniGB-UCS2-H”定义文字的编码标准和样式，GB代表编码方式为gb2312，H代表横排字，V代表竖排字，iTextAsian.jar类库中以cmap结尾的几个文件都是关于编码和样式的。

2.2. IText报表整合项目

【需求】：在移动端health_mobile系统中。预约成功页面，使用导出功能，导出预约的套餐信息，包括检查组-检查项。

1：在页面 orderSuccess.html提供 pdf导出按钮

```

1 <div class="info-title">
2     <span class="name">体检预约成功</span>
3     <input style="font-size: x-small;" id="exportSetmealInfoButton"
4     @click="exportSetmealInfo()" type="button" value="导出预约订单信息">
5 </div>

```



2：添加vue提交的按钮

```

1 methods:{
2     exportSetmealInfo(){
3         window.location.href = "/order/exportSetmealInfo.do?id="+id;
4     }
5 },

```

3：在OrderController中添加exportSetmealInfo的方法，传递订单ID

```

1 /**
2  * 导出预约成功信息
3  * @param id
4  * @param res
5  * @return
6  */
7 @GetMapping("/exportSetmealInfo")
8 public Result exportSetmealInfo(int id, HttpServletResponse res){
9     // 订单信息
10    Map<String,Object> orderInfo = orderService.findOrderDetailById(id);
11    // 套餐详情
12    Integer setmeal_id = (Integer)orderInfo.get("setmeal_id");
13    Setmeal setmeal = setmealService.findDetailById(setmeal_id);
14    // 写到pdf里
15    Document doc = new Document();
16    // 保存到输出流

```

```

17     try {
18         res.setContentType("application/pdf");
19         res.setHeader("Content-
Disposition", "attachment; filename=setmealInfo.pdf");
20         PdfWriter.getInstance(doc, res.getOutputStream());
21         // 打开文档
22         doc.open();
23         // 写内容
24         // 设置表格字体
25         BaseFont cn = BaseFont.createFont("STSongStd-Light", "UniGB-UCS2-
H", false);
26         Font font = new Font(cn, 10, Font.NORMAL, Color.BLUE);
27
28         doc.add(new Paragraph("体检人: " +
(String)orderInfo.get("member"), font));
29         doc.add(new Paragraph("体检套餐: " +
(String)orderInfo.get("setmeal"), font));
30         Date orderDate = (Date) orderInfo.get("orderDate");
31         doc.add(new Paragraph("体检日期: " +
DateUtils.parseDate2String(orderDate, "yyyy-MM-dd"), font));
32         doc.add(new Paragraph("预约类型: " +
(String)orderInfo.get("orderType"), font));
33
34         // 套餐详情
35         Table table = new Table(3); // 3列 表头
36
37         //===== 表格样式 =====
38         // 向document 生成pdf表格
39         table.setWidth(80); // 宽度
40         table.setBorder(1); // 边框
41         table.getDefaultCell().setHorizontalAlignment(Element.ALIGN_CENTER);
//水平对齐方式
42         table.getDefaultCell().setVerticalAlignment(Element.ALIGN_TOP); //
垂直对齐方式
43         /*设置表格属性*/
44         table.setBorderColor(new Color(0, 0, 255)); //将边框的颜色设置为蓝色
45         table.setPadding(5); //设置表格与字体间的间距
46         //table.setSpacing(5); //设置表格上下的间距
47         table.setAlignment(Element.ALIGN_CENTER); //设置字体显示居中样式
48
49         table.addCell(buildCell("项目名称", font));
50         table.addCell(buildCell("项目内容", font));
51         table.addCell(buildCell("项目解读", font));
52
53         // 检查组
54         List<CheckGroup> checkGroups = setmeal.getCheckGroups();
55         if(null != checkGroups){
56             for (CheckGroup checkGroup : checkGroups) {
57                 // 项目名称列
58                 table.addCell(buildCell(checkGroup.getName(), font));
59                 // 项目内容, 把所有的检查项拼接
60                 List<CheckItem> checkItems = checkGroup.getCheckItems();
61                 String checkItemStr = "";
62                 if(null != checkItems){
63                     StringBuilder sb = new StringBuilder();
64                     for (CheckItem checkItem : checkItems) {
65                         sb.append(checkItem.getName()).append(" ");
66                     }

```

```

67         sb.setLength(sb.length()-1); // 去最后一个空格
68         // 检查项的拼接完成
69         checkItemStr = sb.toString();
70     }
71     table.addCell(buildCell(checkItemStr,font));
72     // 项目解读
73     table.addCell(buildCell(checkGroup.getRemark(),font));
74 }
75 }
76 // 添加表格
77 doc.add(table);
78 // 关闭文档
79 doc.close();
80 return null;
81 } catch (Exception e) {
82     e.printStackTrace();
83 }
84 return new Result(false, "导出预约信息失败");
85 }
86
87 // 传递内容和字体样式，生成单元格
88 private Cell buildCell(String content, Font font)
89     throws BadElementException {
90     Phrase phrase = new Phrase(content, font);
91     return new Cell(phrase);
92 }

```

【生成pdf报表的效果】



4: 可根据iText的API，添加表格的样式：

```

1 // 向document 生成pdf表格
2 Table table = new Table(3); //创建3列的表格
3 // 不要复制上行代码
4 table.setWidth(80); // 宽度
5 table.setBorder(1); // 边框
6 table.getDefaultCell().setHorizontalAlignment(Element.ALIGN_CENTER); //水平对
   齐方式
7 table.getDefaultCell().setVerticalAlignment(Element.ALIGN_TOP); // 垂直对齐方
   式
8 /*设置表格属性*/
9 table.setBorderColor(new Color(0, 0, 255)); //将边框的颜色设置为蓝色
10 table.setPadding(5); //设置表格与字体间的间距
11 //table.setSpacing(5); //设置表格上下的间距
12 table.setAlignment(Element.ALIGN_CENTER); //设置字体显示居中样式

```

生成效果：



【小结】

1: iText报表入门案例

- 1: 在项目 health_common 引入 ltext jar支持

- 2: 在health_common中创建测试类TestItext.java, 使用Itext 生成 PDF报表
 - 5个步骤
 - 创建文档
 - 创建pdfWriter(doc,保存的输出流)
 - 打开文档
 - 添加内容
 - 关闭文档
- 3: iText报表解决中文问题 安装jar到仓库(update一下maven本地仓库的索引), 再引用, idea要刷新health_common工程的依赖

2: IText报表整合项目 (订单信息, 套餐详情)

- 1: 在页面 orderSuccess.html提供 pdf导出按钮
- 2: 添加vue提交的按钮方法
- 3: 在OrderController中添加exportSetmealInfo的方法, 传递订单ID, 通过订单ID查询订单信息时, 要额外的查出套餐id(OrderDao.xml 查询语句添加套餐id的列)
- 4: 可根据Itext的API, 添加表格的样式(样式操作起来很烦琐, 且不看好)
- 5: 设置头信息, 内容体类型

2. JasperReports概述

【目标】

JasperReports的使用

【路径】

- 1: JasperReport快速体验
- 2: JasperReport原理
- 3: 开发流程

【讲解】

3.1. JasperReports快速体验

本小节我们先通过一个快速体验来感受一下JasperReports的开发过程。

第一步: 创建maven工程, 导入JasperReports的maven坐标

```
1 <dependency>
2   <groupId>net.sf.jasperreports</groupId>
3   <artifactId>jasperreports</artifactId>
4   <version>6.8.0</version>
5 </dependency>
6 <dependency>
7   <groupId>junit</groupId>
8   <artifactId>junit</artifactId>
9   <version>4.12</version>
10</dependency>
```

第二步: 将提前准备好的jrxml文件(资料\pdf模板文件\demo.jrxml)复制到maven工程中(后面会详细讲解如何创建jrxml文件)

第三步：编写单元测试，输出PDF报表

```
1 package com.itheima.test;
2
3 import net.sf.jasperreports.engine.*;
4 import net.sf.jasperreports.engine.data.JRBeanCollectionDataSource;
5 import org.junit.Test;
6
7 import java.sql.Connection;
8 import java.sql.DriverManager;
9 import java.util.ArrayList;
10 import java.util.HashMap;
11 import java.util.List;
12 import java.util.Map;
13
14 /**
15  * Description: No Description
16  * User: Eric
17  */
18 public class JasperTest {
19
20     @Test
21     public void testJasperReports() throws Exception {
22         // 定义模板jrxml文件路径
23         String jrxml =
24             "C:\\sz89\\jasperdemo\\src\\main\\resources\\demo.jrxml";
25         // 设置编译后的路径
26         String jasper =
27             "C:\\sz89\\jasperdemo\\src\\main\\resources\\demo.jasper";
28         // 编译模板文件
29         // 第一个参数：jrxml，第二个参数为编译后的文件 jasper
30         JasperCompileManager.compileReportToFile(jrxml, jasper);
31         // 构建数据
32         //构造数据
33         Map paramters = new HashMap();
34         paramters.put("reportDate", "2019-10-10");
35         paramters.put("company", "itcast");
36
37         List<Map> list = new ArrayList();
38         Map map1 = new HashMap();
39         map1.put("name", "xiaoming");
40         map1.put("address", "beijing");
41         map1.put("email", "xiaoming@itcast.cn");
42         Map map2 = new HashMap();
43         map2.put("name", "xiaoli");
44         map2.put("address", "nanjing");
45         map2.put("email", "xiaoli@itcast.cn");
46         list.add(map1);
47         list.add(map2);
48
49         // 填充数据 JRBeanCollectionDataSource自定义数据
50         JasperPrint jasperPrint = JasperFillManager.fillReport(jasper,
51             paramters, new JRBeanCollectionDataSource(list));
52         // 导出保存
53         String pdfPath = "D:\\test.pdf";
```

```
51 JasperExportManager.exportReportToPdfFile(jasperPrint, pdfPath);  
52 }  
53 }  
54
```

3.2. JasperReports流程



- JRXML: 报表填充模板, 本质是一个xml文件
- Jasper: 由JRXML模板编译成的二进制文件, 用于代码填充数据
- Jrprint: 当用数据填充完Jasper后生成的对象, 用于输出报表
- Exporter: 报表输出的管理类, 可以指定要输出的报表为何种格式
- PDF/HTML/XML: 报表形式

3.3. JasperReport开发流程【重点】

使用JasperReports导出pdf报表, 开发流程如下:

1. 制作报表模板, jrxml文件, 存放的目录
2. 模板编译 jasper 二进制的文件
3. 构造数据, 调用服务数据
4. 填充数据 可以使用response.getOutputStream()
5. 输出文件 导出时, 下载PDF

【小结】

1: JasperReport快速体验

- 创建工程导入依赖
- 测试方法里:
 - 复制资料中的模板文件demo.jrxml 放到resources
 - 编译后的文件.jasper
 - 编译
 - 创建数据
 - 填充数据到编译后的模板中
 - 导出成pdf

2: JasperReport原理

3: JasperReport的开发流程

3. 模板设计器Jaspersoft Studio

【目标】

1: 学习JasperReport的模板设计器Jaspersoft Studio

【路径】

1: JasperReport面板介绍

2: 创建工程和模板文件

3: 设计模板文件

- (1) 增减Band
- (2) 将元素应用到模板中
- (3) 动态数据填充

4: 结合JasperReport的API输出报表

- (1) JDBC数据源方式填充数据
- (2) JavaBean数据源方式填充数据（推荐）

【讲解】

Jaspersoft Studio是一个图形化的报表设计工具，可以非常方便的设计出PDF报表模板文件(其实就是一个xml文件)，再结合JasperReports使用，就可以渲染出PDF文件。

下载地址:<https://community.jaspersoft.com/community-download>



下载完成后会得到如下安装文件：



直接双击安装即可。

4.1. Jaspersoft Studio面板介绍



4.2. 创建工程和模板文件

打开Jaspersoft Studio工具，首先需要创建一个工程，创建过程如下：



创建完工程后，可以在工程上点击右键，创建模板文件：





可以看到创建处理的模板文件后缀为jrxml，从设计区面板可以看到如下效果：



可以看到整个文件是可视化的，分为几大区域（Title、Page Header、Column Header等），如果某些区域不需要也可以删除。

在面板左下角可以看到有三种视图方式：Design（设计模式）、Source（源码模式）、Preview（预览模式）：

- 通过Design视图可以看到模板的直观结构和样式
- 通过Source视图可以看到文件xml源码
- 通过Preview视图可以预览PDF文件输出后的效果

通过右侧Palette窗口可以看到常用的元素：



4.3. 设计模板文件

4.3.1. 增减Band

可以根据情况删除或者增加模板文件中的区域（称为Band），例如在Page Header区域上点击右键，选择删除菜单：



其中Detail区域可以添加多个，其他区域只能有一个。

4.3.2. 将元素应用到模板中

4.3.2.1. Image元素

从右侧Palette面板中选择Image元素（图片元素），拖动到Title区域：



弹出如下对话框，有多种创建模式，选择URL模式，并在下面输入框中输入一个网络图片的连接地址：





可以选中图片元素，鼠标拖动调整位置，也可以通过鼠标调整图片的大小。

调整完成后，可以点击Preview进入预览视图，查看PDF输出效果：



点击Source进入源码视图，查看xml文件内容：



其实我们上面创建的demo1.jrxml模板文件，本质上就是一个xml文件，只不过我们不需要自己编写xml文件的内容，而是通过Jaspersoft Studio这个设计器软件进行可视化设计即可。

4.3.2.2. Static Text元素

Static Text元素就是静态文本元素，用于在PDF文件上展示静态文本信息：



双击Title面板中的Static Text元素，可以修改文本内容：



选中元素，也可以调整文本的字体和字号：



点击Preview进入预览视图，查看效果：



4.3.2.3. Current Date元素

Current Date元素用于在报表中输出当前系统日期，将改元素拖动到Title区域：



预览输出效果：



默认日期输出格式如上图所示，可以回到设计视图并选中元素，在Properties面板中的Text Field子标签中修改日期输出格式：



修改日期格式：



保存文件后重新预览：



4.3.3. 动态数据填充

上面我们在PDF文件中展示的都是些静态数据，那么如果需要动态展示一些数据应该如何实现呢？我们可以使用Outline面板中的Parameters和Fields来实现。



Parameters通常用来展示单个数据，Fields通常用来展示需要循环的列表数据。

4.3.3.1. Parameters

在Parameters上点击右键，创建一个Parameter参数：



可以在右侧的Properties面板中修改刚才创建的参数名称：



将刚才创建的Parameter参数拖动到面板中：



进入预览视图，查看效果：



由于模板中我们使用了Parameter动态元素，所以在预览之前需要为其动态赋值：



注意：由于我们是在Jaspersoft Studio软件中进行预览，所以需要通过上面的输入框动态为Parameter赋值，在后期项目使用时，需要我们在Java程序中动态为Parameter赋值进行数据填充。

4.3.3.2. Fields

使用Fields方式进行数据填充，既可以使用jdbc数据源方式也可以使用JavaBean数据源方式。

- jdbc数据源数据填充

第一步：在Repository Explorer面板中，在Data Adapters点击右键，创建一个数据适配器



第二步：选择Database JDBC Connection



第三步：选择mysql数据库，并完善jdbc连接信息



为了能够在Jaspersoft Studio中预览到数据库中的数据，需要加入MySQL的驱动包



第四步：回到Design模式下，在Outline视图中，右键点击工程名，选择Database and Query菜单



第五步：在弹出的对话框中选择刚刚创建的JDBC数据库连接选项



第六步：在弹出对话框中Language选择sql，在右侧区域输入SQL语句并点击Read Fields按钮



可以看到通过点击上面的Read Fields按钮，已经读取到了t_setmeal表中的所有字段信息并展示在了下面，这些字段可以根据需要进行删除或者调整位置

第七步：在Outline视图中的Fields下可以看到t_setmeal表中相关字段信息，拖动某个字段到设计区的Detail区域并调整位置



可以看到，在拖动Fields到设计区时，同时会产生两个元素，一个是静态文本，一个是动态元素。静态文本相当于表格的表头，可以根据需要修改文本内容。最终设计完的效果如下：



第八步：使用Preview预览视图进行预览



通过上图可以看到，虽然列表数据展示出来了，但是展示的还有问题。在每条数据遍历时表头也跟着遍历了一遍。这是怎么回事呢？这是由于我们设计的表头和动态Fields都在Detail区域。为了能够解决上面的问题，需要将表头放在Column Header区域，将动态Fields放在Detail区域。具体操作如下：

1、在Outline视图的Column Header点击右键创建出一个区域



2、将Detail下的静态文本拖动到Column Header下



拖动完成后如下：



3、调整静态文本在Column Header区域的位置，最终效果如下



4、预览查看效果



- JavaBean数据源数据填充

第一步：复制上面的demo1.jrxml文件，名称改为demo2.jrxml



修改Report Name：



第二步：打开demo2.jrxml文件，将detail区域中的动态Fields元素删除



第三步：将Outline面板中Fields下的字段全部删除



第四步：清除JDBC数据源和相关SQL语句





第五步：在Fields处点击右键创建新的Field



创建完成后在Properties属性面板中修改Field的名称



第六步：将创建的Fields拖动到Detail区域并调整好位置



注意：使用此种JavaBean数据源数据填充方式，无法正常进行预览，因为这些动态Fields需要在Java程序中动态进行数据填充。

4.3.3.3. 边框

选中表格框，在Borders下添加边框



4.4. 结合JasperReports输出报表

前面我们已经使用Jaspersoft Studio设计了两个模板文件：demo1.jrxml和demo2.jrxml。其中demo1.jrxml的动态列表数据是基于JDBC数据源方式进行数据填充，demo2.jrxml的动态列表数据是基于JavaBean数据源方式进行数据填充。本小节我们就结合JasperReports的Java API来完成pdf报表输出。

4.4.1. JDBC数据源方式填充数据

第一步：创建maven工程，导入相关maven坐标

```

1  <dependency>
2      <groupId>net.sf.jasperreports</groupId>
3      <artifactId>jasperreports</artifactId>
4      <version>6.8.0</version>
5  </dependency>
6  <dependency>
7      <groupId>junit</groupId>
8      <artifactId>junit</artifactId>
9      <version>4.12</version>
10 </dependency>
11 <dependency>
12     <groupId>mysql</groupId>
13     <artifactId>mysql-connector-java</artifactId>
14     <version>5.1.47</version>
15 </dependency>

```

第二步：将设计好的demo1.jrxml文件复制到当前工程的resources目录下



第三步：编写单元测试


```

1  @Test
2  public void testJDBCDataSource() throws Exception{
3      // 定义模板jrxml文件路径
4      String jrxml =
5          "C:\\sz89\\jasperdemo\\src\\main\\resources\\demo1.jrxml";
6      // 设置编译后的路径
7      String jasper =
8          "C:\\sz89\\jasperdemo\\src\\main\\resources\\demo1.jasper";
9      // 编译模板文件
10     // 第一个参数: jrxml, 第二个参为编译后的文件 jasper
11     JasperCompileManager.compileReportToFile(jrxml, jasper);
12
13     // jdbc connection
14     Class.forName("com.mysql.jdbc.Driver");
15     Connection connection =
16         DriverManager.getConnection("jdbc:mysql:///health89", "root", "root");
17
18     Map paramters = new HashMap();
19     paramters.put("company", "itcast");
20
21     // 填充数据 JRBeanCollectionDataSource自定数据
22     JasperPrint jasperPrint = JasperFillManager.fillReport(jasper,
23         paramters, connection);
24     // 导出保存
25     String pdfPath = "D:\\testJDBC.pdf";
26     JasperExportManager.exportReportToPdfFile(jasperPrint, pdfPath);
27 }

```

通过上面的操作步骤可以输出pdf文件，但是中文的地方无法正常显示。这是因为JasperReports默认情况下对中文支持并不友好，需要我们自己进行修复。具体操作步骤如下：

1、在Jaspersoft Studio中打开demo1.jrxml文件，选中中文相关元素，统一将字体设置为“华文宋体”并将修改后的demo1.jrxml重新复制到maven工程中

解决中文显示

2、将本章资源/解决中文无法显示问题目录下的文件复制到maven工程的resources目录中



按照上面步骤操作后重新执行单元测试导出PDF文件：



4.4.2. JavaBean数据源方式填充数据【重点】

第一步：为了避免中文无法显示问题，首先需要将demo2.jrxml文件相关元素字体改为“华文宋体”并将demo2.jrxml文件复制到maven工程的resources目录下



第二步：编写单元测试方法输出PDF文件

```

1  @Test
2  public void testJavaBean() throws Exception {
3      // 定义模板jrxml文件路径

```

```

4      String jrxml =
      "C:\\sz89\\jasperdemo\\src\\main\\resources\\demo2.jrxml";
5      // 设置编译后的路径
6      String jasper =
      "C:\\sz89\\jasperdemo\\src\\main\\resources\\demo2.jasper";
7      // 编译模板文件
8      // 第一个参数: jrxml, 第二个参为编译后的文件 jasper
9      JasperCompileManager.compileReportToFile(jrxml, jasper);
10     // 构建数据
11     //构造数据
12     Map paramters = new HashMap();
13     paramters.put("company", "传智播客");
14
15     List<Map> list = new ArrayList();
16     Map map1 = new HashMap();
17     map1.put("tName", "入职体检套餐");
18     map1.put("tCode", "RZTJ");
19     map1.put("tAge", "18-60");
20     map1.put("tPrice", 500d);
21
22     Map map2 = new HashMap();
23     map2.put("tName", "阳光爸妈老年健康体检");
24     map2.put("tCode", "YGBM");
25     map2.put("tAge", "55-60");
26     map2.put("tPrice", 500d);
27     list.add(map1);
28     list.add(map2);
29
30     // 填充数据 JRBeanCollectionDataSource自定数据
31     JasperPrint jasperPrint = JasperFillManager.fillReport(jasper,
paramters, new JRBeanCollectionDataSource(list));
32     // 导出保存
33     String pdfPath = "D:\\testJavaBean.pdf";
34     JasperExportManager.exportReportToPdfFile(jasperPrint, pdfPath);
35 }

```

查看输出效果:



【小结】

1: JasperReport面板介绍

2: 创建工程和模板文件

3: 设计模板文件

(1) 增减Band

(2) 将元素应用到模板中 拖拽组件 设置属性

(3) 动态数据填充

- 数据库连接Connection (create database adapter mysql -> 指定驱动jar) Detail
- 自定义JavaBean的方式 one record empty row Detail

4: 结合JasperReport的API输出报表

(1) JDBC数据源方式填充数据 connection

(2) JavaBean数据源方式填充数据（推荐）

【注】【注】【注】：设计的模板中，所有的中文字符全部使用【华文宋体】

复制资料中的 \解决中文无法显示问题\ 目录下的文件到resources目录下

4.在项目中输出运营数据PDF报表【重点】

【目标】

1: 项目应用，项目中输出运营数据统计的PDF报表

【路径】

1: 设计PDF模板文件

2: 搭建环境health_web

- 引入依赖
- 模板文件
- 中文显示的资源

3: 修改页面

- 添加导出按钮，提交下载pdf请求

4: 代码实现

- 查询运营统计数据
- 定义模板所在
- 定义编译后的文件
- 会员统计、套餐预约统计 parameters Map
- 热门套餐 list Field JRCollectionDataSource
- 设置响应头信息，内容体格式
- 导出pdf

【讲解】

本小节我们将在项目中实现运营数据的PDF报表导出功能。

5.1. 设计PDF模板文件【次重点】

使用Jaspersoft Studio设计运营数据PDF报表模板文件health_business3.jrxml，设计后的效果如下：



在资源中已经提供好了此文件，直接使用即可，但是让大家更好的掌握Jaspersoft Studio，我们重新画。

注意1：需要使用Title、Column Header、Detail，剩下的不需要



其中：

注意事项一：

1: Title

用于存放图片，并显示图片



2: Column Header

用于存放静态文本（static Text）和变量（parameters），表示显示文本数据，并使用参数的形式传递



3: Detail

用于存放Fields，表示循环遍历热门套餐。



注意事项二：

类型要一致

```

1  1: Paramters:
2
3  reportDate --> java.lang.String
4  todayNewMember -- > java.lang.Integer
5  totalMember -- > java.lang.Integer
6  thisweekNewMember -- > java.lang.Integer
7  thisMonthNewMember -- > java.lang.Integer
8  todayOrderNumber -- > java.lang.Integer
9  todayVisitsNumber -- > java.lang.Integer
10 thisweekOrderNumber -- > java.lang.Integer
11 thisweekVisitsNumber -- > java.lang.Integer
12 thisMonthOrderNumber -- > java.lang.Integer
13 thisMonthVisitsNumber -- > java.lang.Integer
14
15 2: Fields:
16
17 name --> java.lang.String
18 setmeal_count --> java.lang.Long
19 proportion --> java.math.BigDecimal

```

【注意】字体一定要是 华文宋体

5.2. 搭建环境

第一步：在health_parent工程的pom.xml中导入JasperReports的maven坐标

```

1  <jasperreports.version>6.8.0</jasperreports.version>
2  <dependency>
3      <groupId>net.sf.jasperreports</groupId>
4      <artifactId>jasperreports</artifactId>
5      <version>${jasperreports}</version>
6  </dependency>

```

在health_common工程的pom.xml中导入JasperReports的maven坐标

```
1 <dependency>
2     <groupId>net.sf.jasperreports</groupId>
3     <artifactId>jasperreports</artifactId>
4 </dependency>
```

第二步：将资源中提供的模板文件health_business3.jrxml复制到health_web工程的template目录下

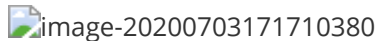


第三步：将解决中问题的相关资源文件复制到项目中



5.3. 修改页面

修改health_web工程的report_business.html页面，添加导出PDF的按钮并绑定事件



5.4. Java代码实现

在health_web工程的ReportController中提供exportBusinessReport4PDF方法

```
1 /**
2  * 导出运营统计数据报表
3  */
4 @GetMapping("/exportBusinessReportPdf")
5 public Result exportBusinessReportPdf(HttpServletRequest req,
6 HttpServletResponse res){
7     // 获取模板的路径，getRealPath("/") 相当于到webapp目录下
8     String basePath =
9 req.getSession().getServletContext().getRealPath("/template");
10    // jrxml路径
11    String jrxml = basePath + File.separator + "report_business.jrxml";
12    // jasper路径
13    String jasper = basePath + File.separator + "report_business.jasper";
14
15    try {
16        // 编译模板
17        JasperCompileManager.compileReportToFile(jrxml, jasper);
18        Map<String, Object> businessReport =
19 reportService.getBusinessReport();
20        // 热门套餐(list -> Detail)
21        List<Map<String, Object>> hotSetmeals =
22 (List<Map<String, Object>>)businessReport.get("hotSetmeal");
23        // 填充数据 JRBeanCollectionDataSource自定义数据
24        JasperPrint jasperPrint = JasperFillManager.fillReport(jasper,
25 businessReport, new JRBeanCollectionDataSource(hotSetmeals));
26        res.setContentType("application/pdf");
27        res.setHeader("Content-
28 Disposition", "attachment;filename=businessReport.pdf");
29        JasperExportManager.exportReportToPdfStream(jasperPrint,
30 res.getOutputStream());
31        return null;
32    } catch (Exception e) {
```

```
26         e.printStackTrace();
27     }
28     return new Result(false, "导出运营数据统计pdf失败");
29 }
```

【小结】

1: 设计PDF模板文件

注意:

1. 如果是单个变量（不需要循环）不要放到Detail，配以静态文本
2. 如果是一个集合的数据需要用到遍历，放到Detail 循环
3. 中文显示问题

- 模板中，所有的中文一定要设置它的字体为 "华文宋体"
- 在项目Resources目录，添加宋体的支持
- 注意看target目录，有没加入中文显示支持授及模板文件

4. 参数对应的数据类型

会员数量，预约数量 类型都为integer

热门套餐 name String, setmeal_count long, proportion: BigDecimal

2: 搭建环境

引入依赖

3: 修改页面

添加导出按钮，添加导出按钮的事件方法

4: 代码实现

获取模板，编译模板，填充数据，导出 设置头信息，内容体