

[首页 >](#)

## 使用JasperReport+iReport进行WEB开发

--作者:金山

## 一、JasperReport和iReport简介:

## 1、JasperReport简介

JasperReport是一个强大、灵活的报表生成工具，是开源组织sf.net中的一个java 报表打印工程。能够展示丰富的页面内容，并将之转换成PDF, HTML, XML, Excel（通过POI或JExcelAPI实现）和Rtf（通过POI实现）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。它的主要目的是辅助生成面向页面的（page oriented），准备付诸打印的文档。JasperReport借由定义于XML文档中的report design进行数据组织。这些数据可能来自不同的数据源，包括关系型数据库，collections, java对象数组。通过实现简单的接口，用户就可以将report library插入到订制好的数据源中。用JasperReport进行报表开发的过程如图1.1:

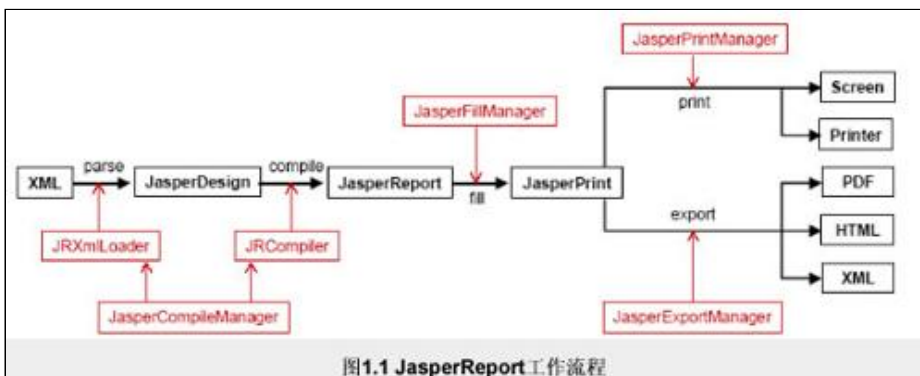


图1.1 JasperReport工作流程

## 2、iReport简介

iReport也是开源组织sf.net中的一款免费软件，是为JasperReport设计的强大的，直观的，易于使用的可视化报表设计器，采用纯Java开发。这个工具允许用户可视化编辑包含charts、图片、子报表等的复杂报表。iReport 还集成了JFreeChart图表制作包，允许用户可视化地编辑XML(JasperDesign)文件。用于打印的数据可以通过多种方式获取包括: JDBC, TableModels, JavaBeans, XML,Hibernate(支持HQL查询语言), CSV等。它支持多种输出格式包括: PDF,RTF,XML,XLS,CSV,HTM。

注意下载的iReport一定要与你所使用的JasperReport 版本相匹配，当然您也可以更新iReport 中所使用的JasperReport 组件。

## 二、JasperReport+iReport开发

## 1、报表设计一般过程

## • 定制报表格式

使用iReport制定报表模板或者直接写jrxml文件，其实就是xml文件，只不过是后缀名不一样罢了。将jrxml文件编译后就生成了jasper模板文件。

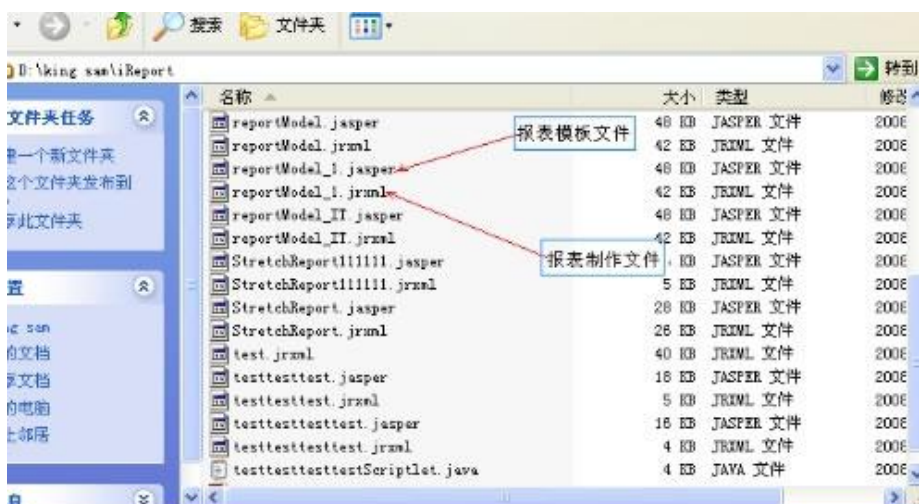


图2.1: 报表模板文件

联系我们:

☎ 电话: 800-830-9336

✉ Email: [点击联系我们](#)

## 本期相关文章

- » 中国地质大学“软酷·卓越实验室”挂牌 致力GIS领域项目研发
- » 卓越实验室无锡区域中心投入运营
- » “软酷·卓越实验室”项目实训纳入高校学分 正式进入课程体系
- » 使用JasperReport+iReport进行WEB开发
- » 企业资产管理系统

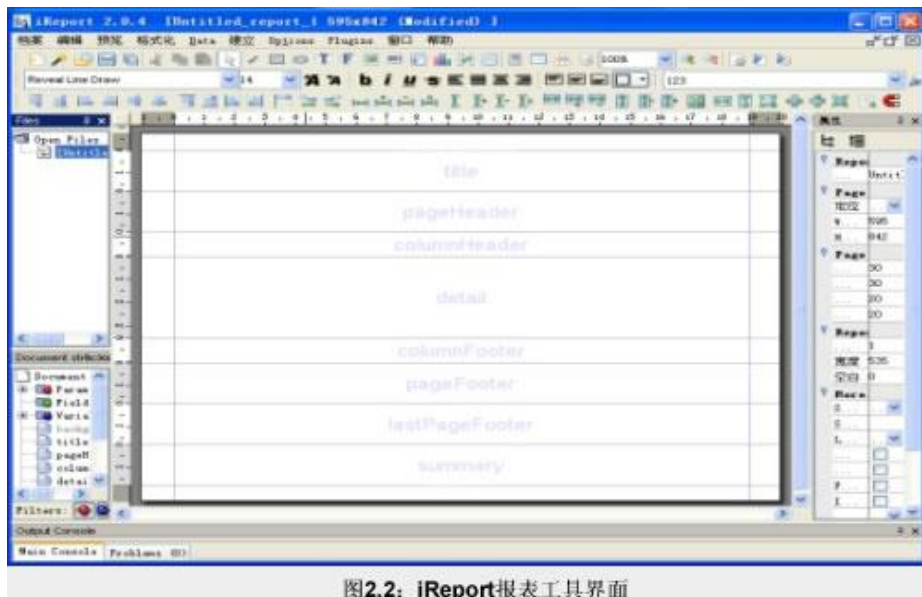


图2.2: iReport报表工具界面

- 填充数据  
填充数据一般使用二种方式，一种方式是通过JDBC连接提供数据源，一种就是通过 JavaBean 的集合提供数据源。当然还有web Service的xml文件提供的。我的建议是，如果你的程序中的统计直接使用JDBC就可以完成，那么就使用jdbc数据源的方法，反之，使用javaBean的集合是不错的选择，因为这样不会在意你的数据的来源，你也可以任意处理，比如说，要通过权限检查的数据才在报表中生成的话，就可以过滤到不符合权限的数据。
- 显示  
将JasperReport生成的文件直接显示出来。点击执行报表按钮。

## 2、报表安装与配置

- JDK安装  
安装jdk并配置系统环境变量JAVA\_HOME；在classpath,path中均要配置。  
例如：JAVA\_HOME: C:/java/jdk1.5.0\_06;  
classpath: c:/java/jdk1.5.0\_06/lib/tools.jar;c:/java/jdk1.5.0\_06/lib/dt.jar;  
path: c:/java/jdk1.5.0\_06/bin;
- iReport的安装：  
iReport的安装直接下一步就可以了，如果下载的是解压版的，解压后就可以用了。

## 3、iReport入门

- 新建报表文档

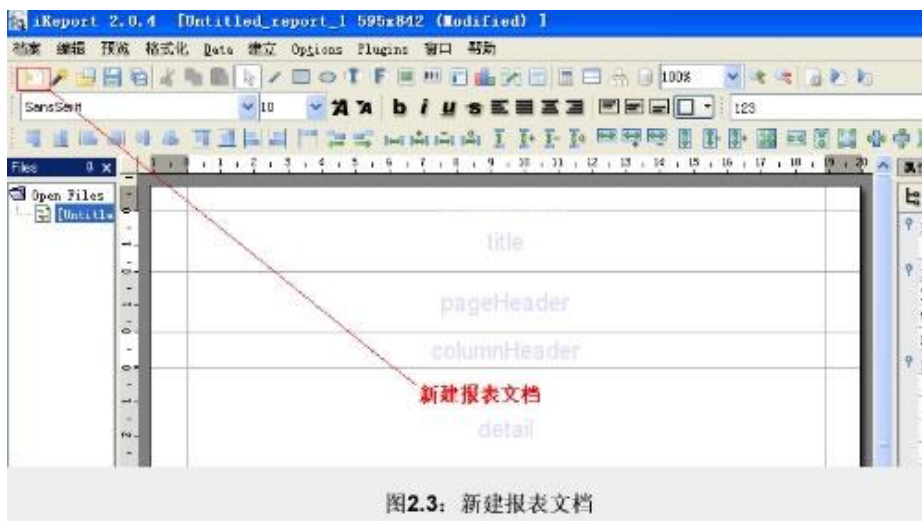
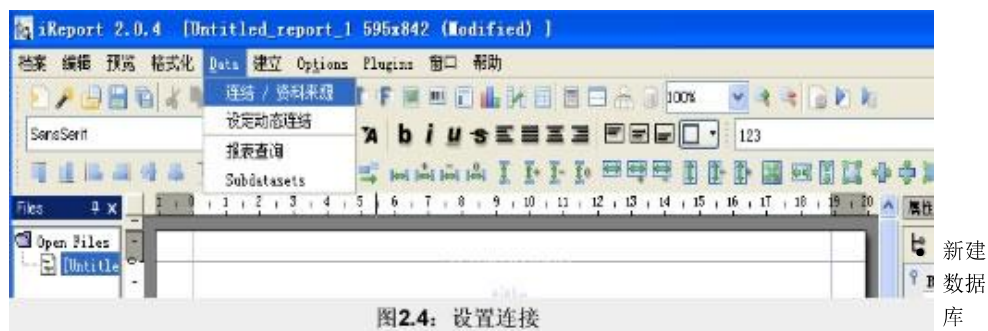


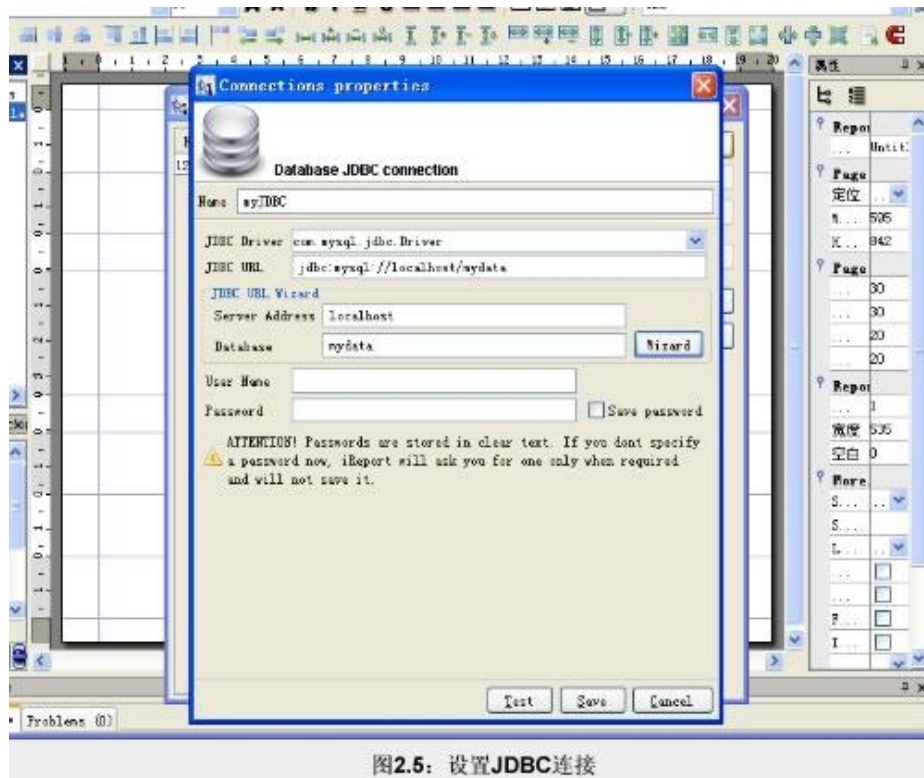
图2.3: 新建报表文档

- 选择"Data---->连接/资料来源"菜单



JDBC连接

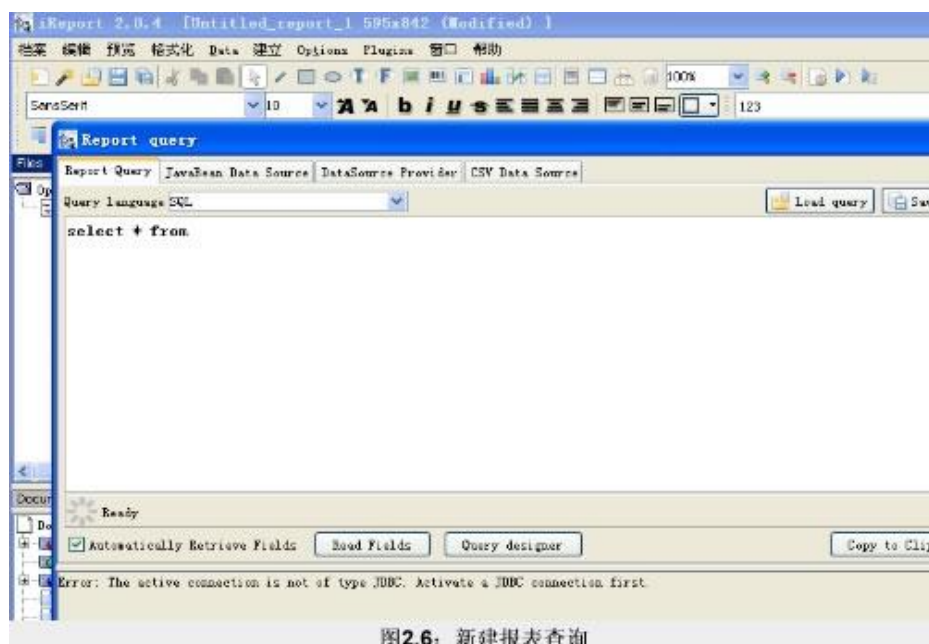
单击“new”按钮,弹出如下窗口,设置JDBC连接参数



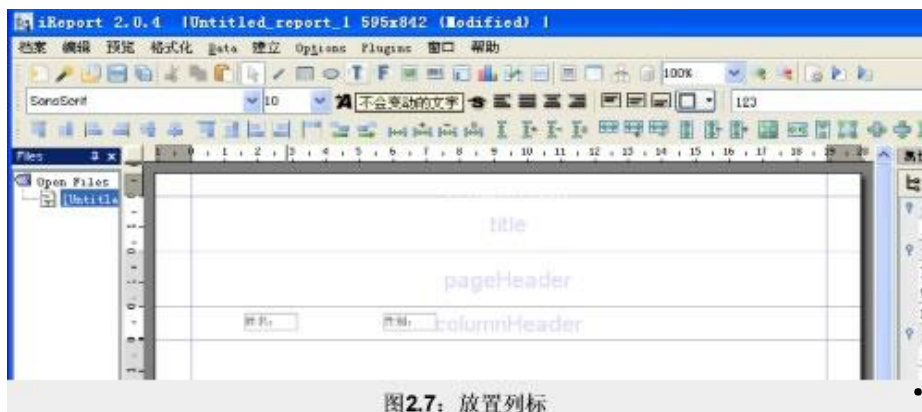
- 新建报表查询

选择“Data---->报表查询”菜单,出现如下窗口,输入SQL语句:

select \* from employee

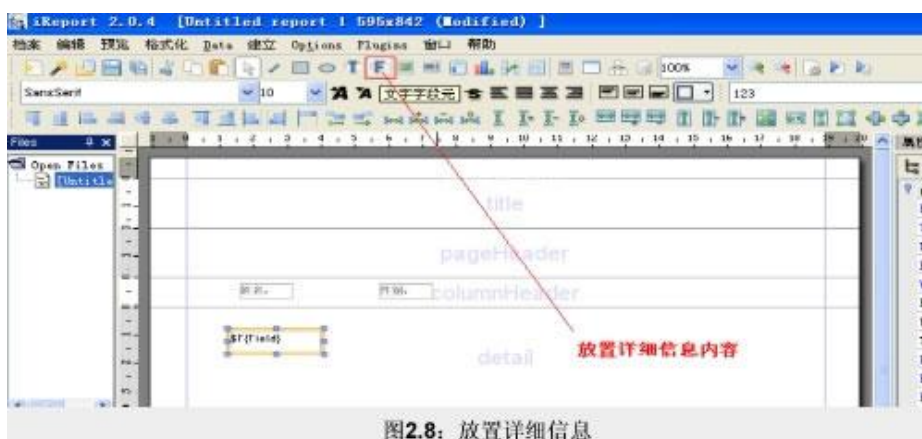


- 放置列标题



- 放置详细

信息



- 查看报表结果



报表设计注意事项

(1) 报表模板中每一个text或者file文本域属性common中的Position Type是很有用的一个属性，在使用中如果遇到文本域需要浮动则选择float。其他相应一些技巧均在属性中可以找到，在模板制作过程中应多注意，这里不再多说。





使用JavaBean做为数据源的话，制作一个javaBean将其打成jar文件，在iReport工具中的Options选项下的Classpath中使用添加jar将文件引入。在Data□Report query中选择JavaBean Data Source将jar文件中的类名字输入到Class name位置点击右边按钮就可以了， javaBean中定义的字段就显示出来了选择自己需要用的字段点击add按钮。

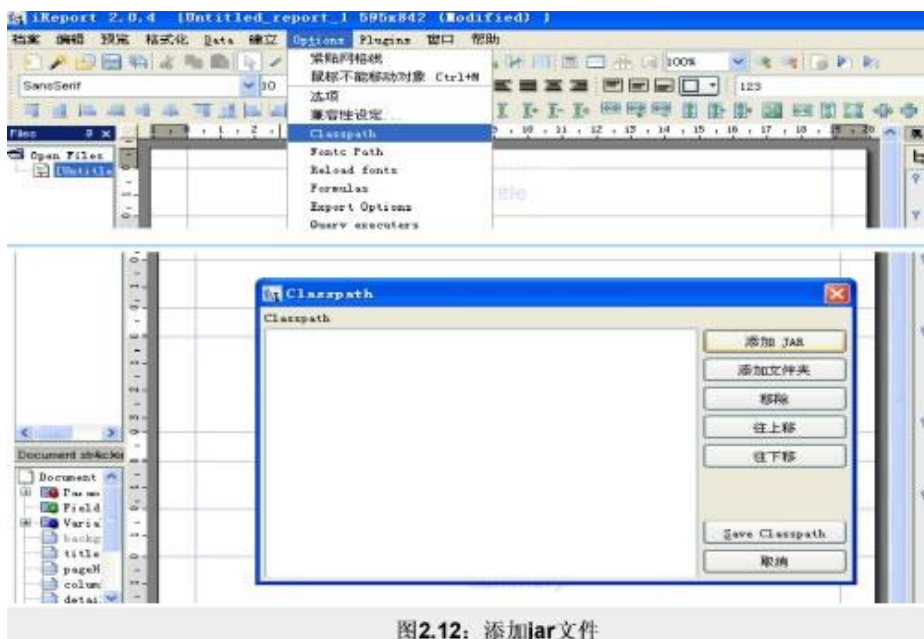


图2.12: 添加jar文件

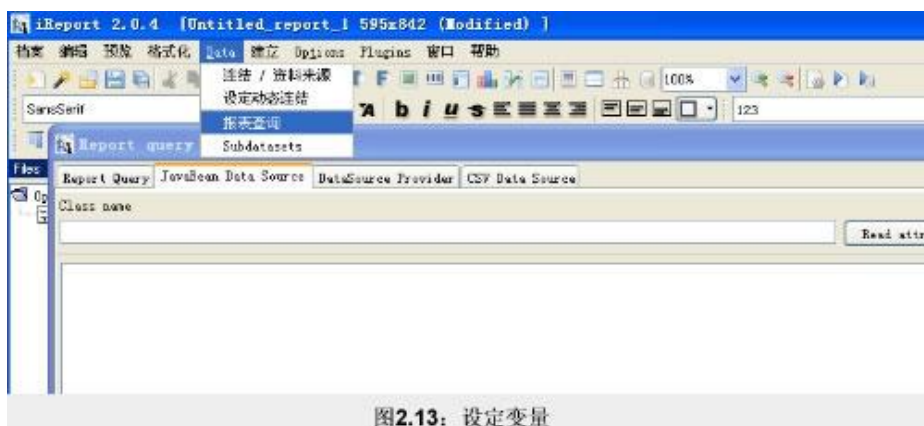


图2.13: 设定变量

写一个DataSourceFactory类打成jar文件，并在classpath中将文件引入，然后在Data□Connections/Datasources选择new再弹出的对话框中选择JavaBeans set data source 然后将刚刚jar文件中的factory类对应的名字写下来就可以了。

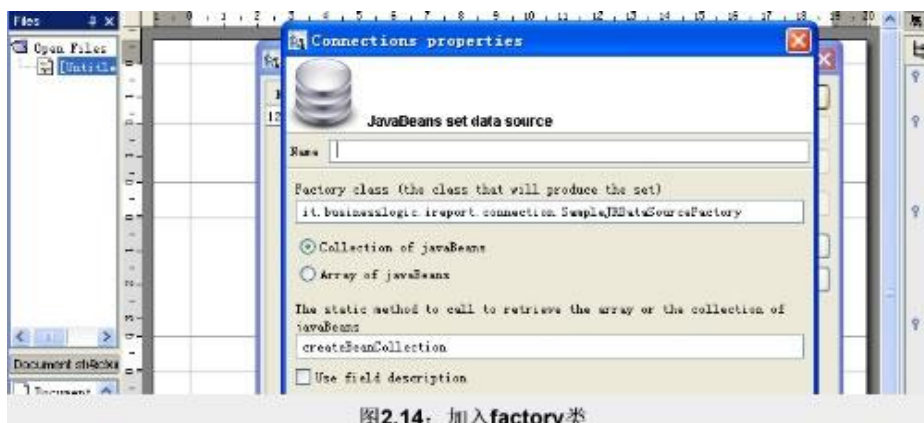


图2.14: 加入factory类

最后在iReport中动态执行报表就可以了。

## 5、iReport生成文件中的字体问题

iReport在生成文件时有可能会遇到中间乱码的问题，那么在制作模板时就要注意中文字段属性要正确的选择,而且3个针对于pdf中文乱码问题的jar文件(iTextAsian.jar, itext-1.3.1.jar, iTextAsianCmaps.jar(该文件我似乎

没有))不能少。还有一点就是关于中文字符问题可以将操作系统自带的字体文件copy到iReport对应的文件夹中就可以使用，在web工程上要放的class文件夹下。例如中文宋体字体文件(C:\WINDOWS\Fonts 文件夹下的文件)copy到web工程下的class文件夹下。

## 6、java代码实例

```
1.package report;
2.
3.import java.sql.Connection;
4.import java.sql.DriverManager;
5. /**
6.*数据库的连接类
7.* /
8.public class JDBCConnection {
9.public static Connection getConnection() {
10.try {
11.String url = "jdbc:oracle:thin:@127.0.0.1:1521:ruanko";
12.Class.forName("oracle.jdbc.driver.OracleDriver");
13.Connection con = DriverManager.getConnection(url, "little", "little");
14.return con;
15. }
16. } catch (Exception e) {
17.     e.printStackTrace();
18. }
19. return null;
20. }
```

java 代码

```
1.package report.datasource;
2.
3.import java.util.HashMap;
4.import java.util.Iterator;
5.import java.util.List;
6.import java.util.Map;
7.
8.import net.sf.jasperreports.engine.JRDataSource;
9.import net.sf.jasperreports.engine.JRException;
10.import net.sf.jasperreports.engine.JRField;
11. /**
12.* dataSource类(也就是数据填充类)，实现JRDataSource接口
13.* 通过放在list里面的Map对象迭代实现数据对应
14.* /
15.public class ReportDataSource implements JRDataSource {
16.
17.private List datas = DataSourceBaseFactory.createBeanCollection(String id);
18.
19.private Iterator iter = datas.iterator();
20.
21.Map map = new HashMap();
22.
23.public ReportDataSource() {
24.    }
25.
26.public ReportDataSource(String id) {
27.    }
28.
29.public boolean next() throws JRException {
30.    if(iter.hasNext()) {
31.        map = (Map) iter.next();
32.        return true;
33.    }
34.    return false;
35.    }
36. }
```

```

37. public Object getFieldValue(JRField arg0) throws JRException {
38. return map.get(arg0.getName());
39. }
40.
41. }

```

java代码

```

1. package report.factory;
2.
3. import report.JDBCConnection;
4. import java.sql.Connection;
5. import java.sql.ResultSet;
6. import java.sql.SQLException;
7. import java.sql.Statement;
8. import java.sql.Timestamp;
9. import java.util.ArrayList;
10. import java.util.HashMap;
11. import java.util.List;
12. import java.util.Map;
13. /**
14. * Map中的键值要与模板文件的file值对应。
15. */
16. public class DataSourceBaseFactory {
17.
18. public static List createBeanCollection(String id) {
19.
20. ResultSet rs = null;
21. Statement st = null;
22. Connection con = null;
23. List datas = new ArrayList();
24.
25. try {
26. con = JDBCConnection .getConnection();
27. st = con.createStatement();
28. rs = st. executeQuery ( "select name,password,sex from people where id = " id);
29. while(rs.next()){
30. Map attris = new HashMap();
31. attris.put("name", rs.getString( "name" ));
32. attris.put("password", rs.getString( "password" ));
33. attris.put("sex", rs.getString( "sex" ));
34. datas.add(attris);
35. }
36. } catch (Exception e) {
37. e.printStackTrace();
38. } finally {
39. try {
40. if(rs != null) rs.close();
41. if(st != null) st.close();
42. if(con != null) con.close();
43. } catch (SQLException e) {
44. e.printStackTrace();
45. }
46. }
47. return datas;
48. }
49. }
50.

```

java 代码

```

1. package report.bean
2.
3. import java.io.Serializable;
4. /**

```



```

5.* dataSource的javaBean类。用于创建模板
6.*/
7.public class DataSourceBean implements Serializable {
8.
9.    private static final long serialVersionUID = -4038978834455400007L;
10.
11.    private String name;
12.    private String password;
13.    private String sex;
14.
15.    public String getName() {
16.        return name;
17.    }
18.
19.    public void setName(String name) {
20.        this.name = name;
21.    }
22.
23.    public String getPassword () {
24.        return password;
25.    }
26.
27.    public void setPassword(String password) {
28.        this.password = password
29.    }
30.
31.    public String getSex() {
32.        return sex;
33.    }
34.
35.    public void setSex(String sex) {
36.        this.sex = sex;
37.    }
38.
39.}

```

java 代码

```

1.package report;
2.
3.import java.io.ByteArrayOutputStream;
4.import java.io.File;
5.import java.io.FileOutputStream;
6.import java.io.IOException;
7.import java.util.Date;
8.import java.util.HashMap;
9.import java.util.Map;
10.
11.import net.sf.jasperreports.engine.JRAbstractExporter;
12.import net.sf.jasperreports.engine.JRException;
13.import net.sf.jasperreports.engine.JRExporterParameter;
14.import net.sf.jasperreports.engine.JasperFillManager;
15.import net.sf.jasperreports.engine.JasperPrint;
16.import net.sf.jasperreports.engine.export.JRPdfExporter;
17.import net.sf.jasperreports.engine.export.JRPdfExporterParameter;
18./**
19.* 测试入口类，生成pdf文件
20.* JasperFillManager中有多个生成文件的方法
21.* 除了可以生成pdf文件外还可以生成office文档文件。
22.*/
23.public class TestReportHere {
24.
25.public static void main(String[] args) {

```

```

26. Map parameters = new HashMap();
27.
28. ByteArrayOutputStream outPut = new ByteArrayOutputStream();
29. FileOutputStream outputStream = null;
30. File file = new File("E:/workspace/report.pdf");
31. String reportModelFile = "E:/workspace/reportModel.jasper";
32.
33. try {
34. JasperPrint jasperPrint = JasperFillManager.fillReport(reportModelFile,
    parameters, new ReportDataSource("123"));
35. JRAbstractExporter exporter = new JRPdfExporter();
36. /**
37. * 创建jasperPrint
38. */
39. exporter.setParameter(JRExporterParameter.JASPER_PRINT, jasperPrint);
40. /**
41. * 生成输出流
42. */
43. exporter.setParameter(JRExporterParameter.OUTPUT_STREAM, outPut);
44. /**
45. * 屏蔽copy功能
46. */
47. exporter.setParameter(JRPdfExporterParameter.IS_ENCRYPTED, Boolean.TRUE);
48. /**
49. * 加密
50. */
51. exporter.setParameter(JRPdfExporterParameter.IS_128_BIT_KEY, Boolean.TRUE);
52. exporter.exportReport();
53. outputStream = new FileOutputStream(file);
54. outputStream.write(outPut.toByteArray());
55.         } catch (JRException e) {
56.             e.printStackTrace();
57.         } catch (Exception e) {
58.             e.printStackTrace();
59.         } finally{
60.             try {
61.                 outPut.flush();
62.                 outPut.close();
63.             } catch (IOException e) {
64.                 e.printStackTrace();
65.             }
66.         }
67.     }
68.
69. }
70.

```

 [TOP](#)  [关闭](#)