

## 第10章 用户档案PDF报表

- 理解数据填充的两种方式
- 熟练构造分组报表
- 熟练构造Chart图形报表
- 实现个人档案的PDF输出

### 1 数据填充

我们介绍了如何使用JasperReport来生成简单的文本报表，正式企业开发中动态数据展示也是报表中最重要的一环，接下来我们共同研究的就是填充动态数据到PDF报表中。

```
/**
 * 填充数据构造JasperPrint
 *      is: 文件输入流
 * parameters: 参数
 * dataSource: 数据源
 */
public static JasperPrint fillReport(InputStream is, Map<String, Object> parameters,
JRDataSource dataSource) throws JRException {
```

通过这段填充数据的源代码得知，JasperReport对报表模板中的数据填充有很多中方式，最典型的有以下两种：

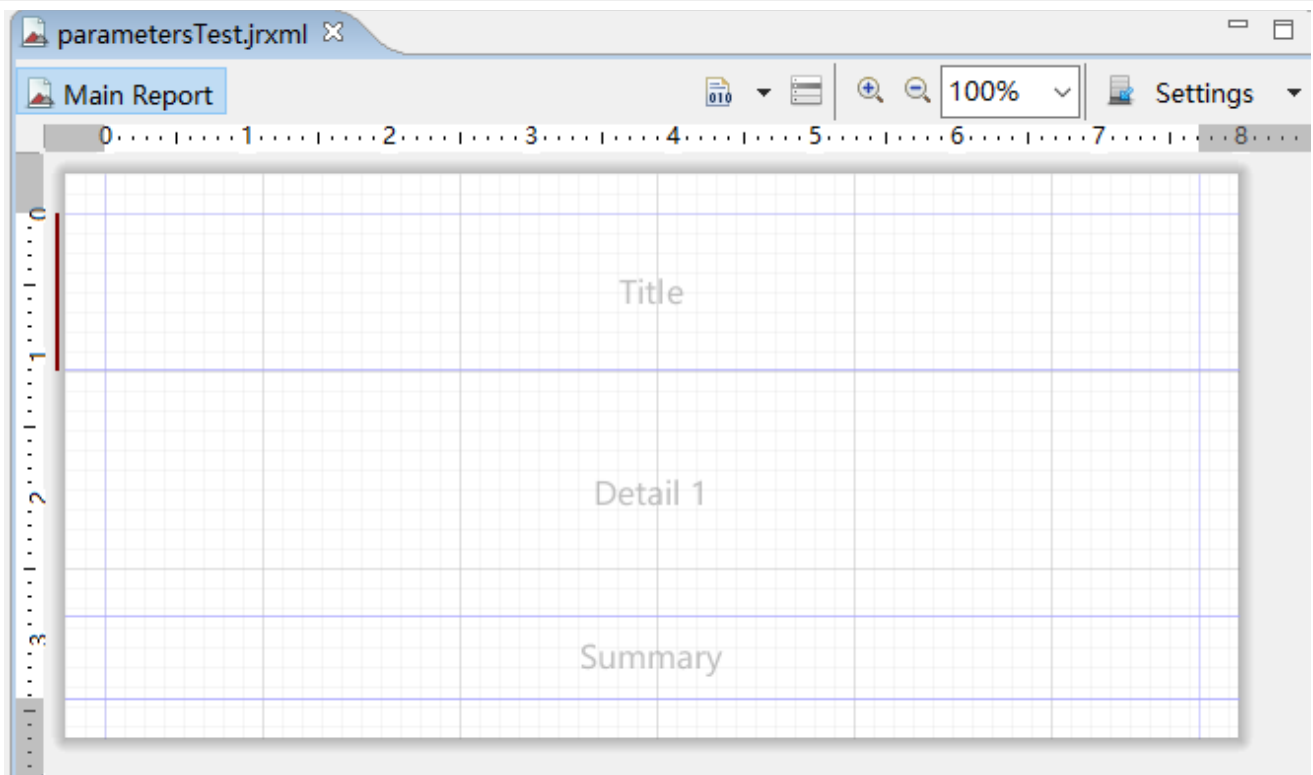
- Parameters（参数）填充
- DataSource（数据源）填充

#### 1.1 参数Map填充数据

Parameters通常是用来在打印的时候从程序里传值到报表里。也就是说parameters通常的是起参数传递的作用。他们可以被用在一些特定的场合(比如应用中SQL 查询的条件),如report中任何一个需要从外部传入的变量等(如一个Image对象所包括的char或报表title的字符串)。parameters也需要在创建的时候定义它的数据类型。parameters的数据类型是标准的java的Object。

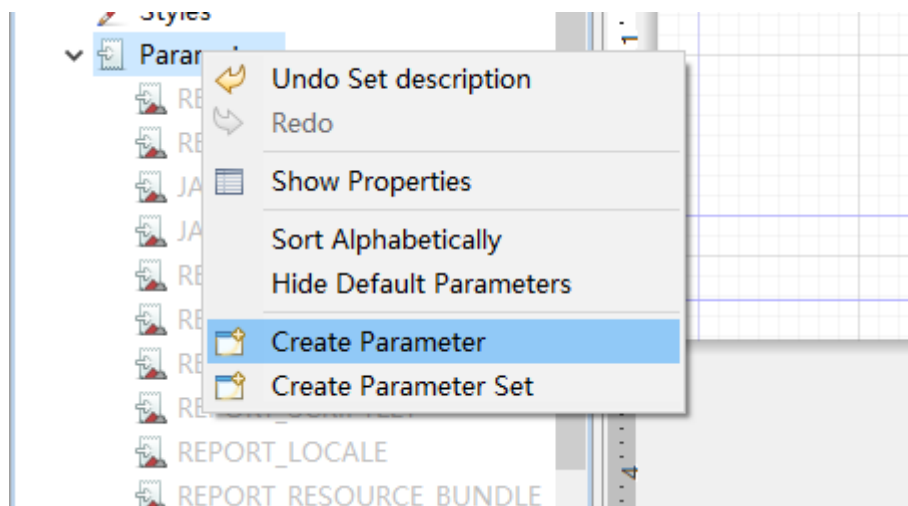
##### 1.1.1 模板制作

(1) 创建新模板，删除不需要的Band

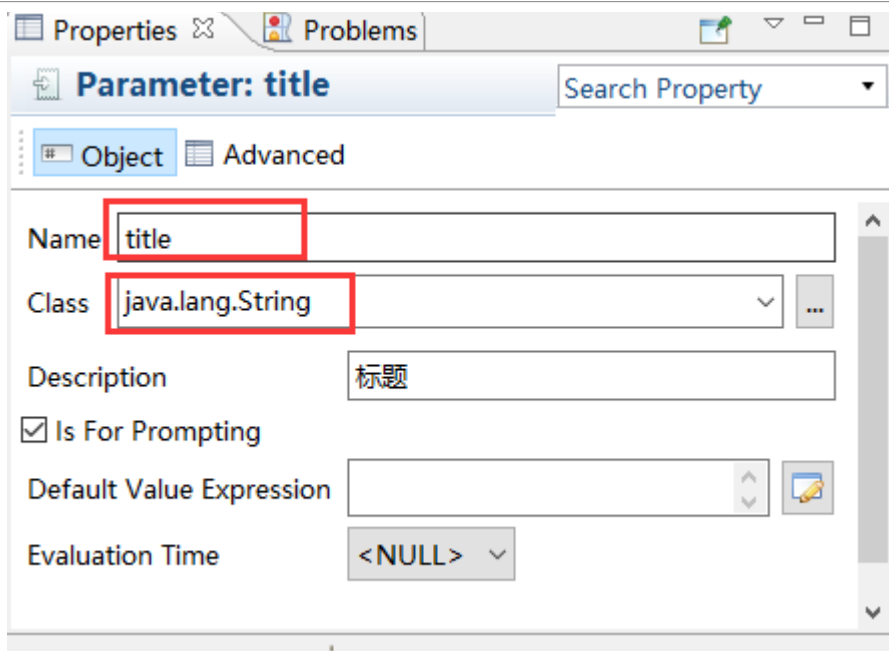


## (2) 创建Parameter

在outline面板中找到Parameters,右键 -> Create Parameter,新建一个Parameter(生成一个Parameter1)



右键 Parameter1 -> Show Properties. 设置Name为title、Class为java.lang.String.这里要注意名字要认真取不能重复，因为传入的参数key就是这个参数名，以此来进行一一对应



### (3) 模板参数设置

将设置好的参数直接拖入表格中对应的位置，并设置好大小与对齐方式。

<div> <div>\$P{title}</div> <div>Page Header</div> </div>			
用户名:	\$P{username}	手机号:	\$P{mobile}
所属企业	\$P{tail 1}	所属部门:	\$P

## 1.1.2 PDF输出

```

@GetMapping("/testJasper02")
public void createPdf(HttpServletRequest request, HttpServletResponse response)
throws IOException {
    //1.引入jasper文件
    Resource resource = new ClassPathResource("templates/parametersTest.jasper");
    FileInputStream fis = new FileInputStream(resource.getFile());
    //2.创建JasperPrint,向jasper文件中填充数据

    ServletOutputStream os = response.getOutputStream();
    try {
        /**
         * parameters集合中传递的key需要和设计模板中使用的name一致
         */
        HashMap parameters = new HashMap();
        parameters.put("title", "用户详情");
        parameters.put("username", "李四");
        parameters.put("companyName", "传智播客");
        parameters.put("mobile", "120");
        parameters.put("departmentName", "讲师");
    }
}
    
```

```
JasperPrint print = JasperFillManager.fillReport(fis, parameters,new  
JREmptyDataSource());  
//3. 将JasperPrint已PDF的形式输出  
JasperExportManager.exportReportToPdfStream(print,os);  
response.setContentType("application/pdf");  
  
} catch (JRException e) {  
    e.printStackTrace();  
}finally {  
    os.flush();  
}  
}
```

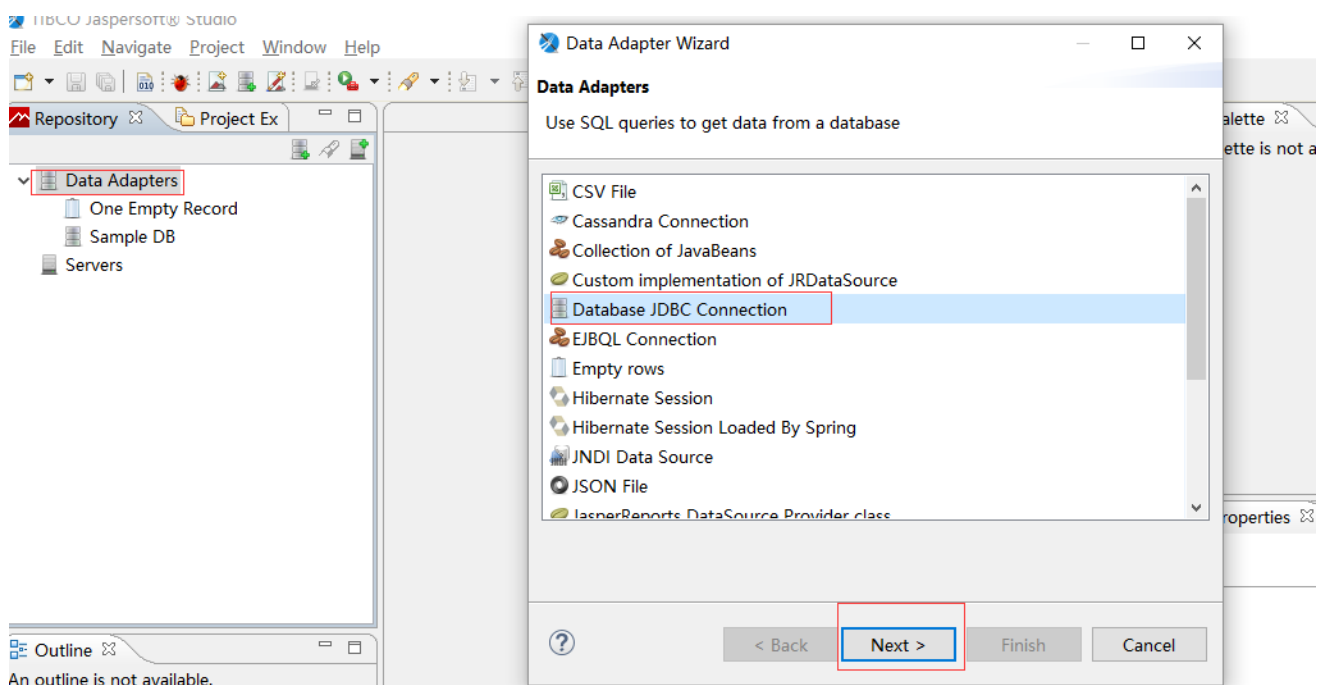
## 1.2 数据源填充数据

### 1.2.1 JDBC数据源

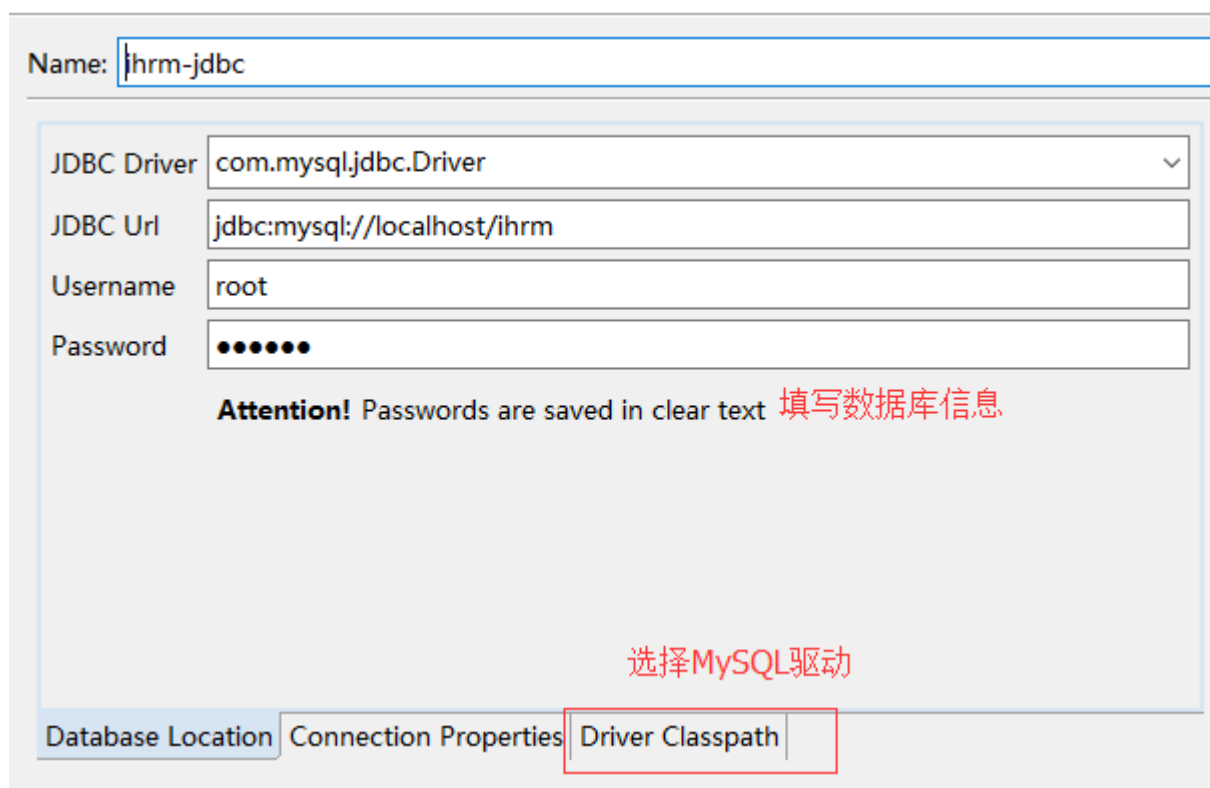
#### 1.2.1.1 配置数据连接

使用JDBC数据源填充数据：使用Jaspersoft Studio 先要配置一个数据库连接

填写数据源的类型，选择“Database JDBC Connection”



配置数据库信息



Name:

JDBC Driver:

JDBC Url:

Username:

Password:

**Attention!** Passwords are saved in clear text 填写数据库信息

选择MySQL驱动

Database Location | Connection Properties | **Driver Classpath**

这一步，需要：（1）给创建的这个数据连接起个名字；（2）根据数据库选择驱动类型；Jaspersoft Studio 已经内置了很多常用数据库的驱动，使用的时候直接选就可以了。当然，如果这还满足不了你的话，你还可以添加你指定的 JDBC 驱动 jar 包。

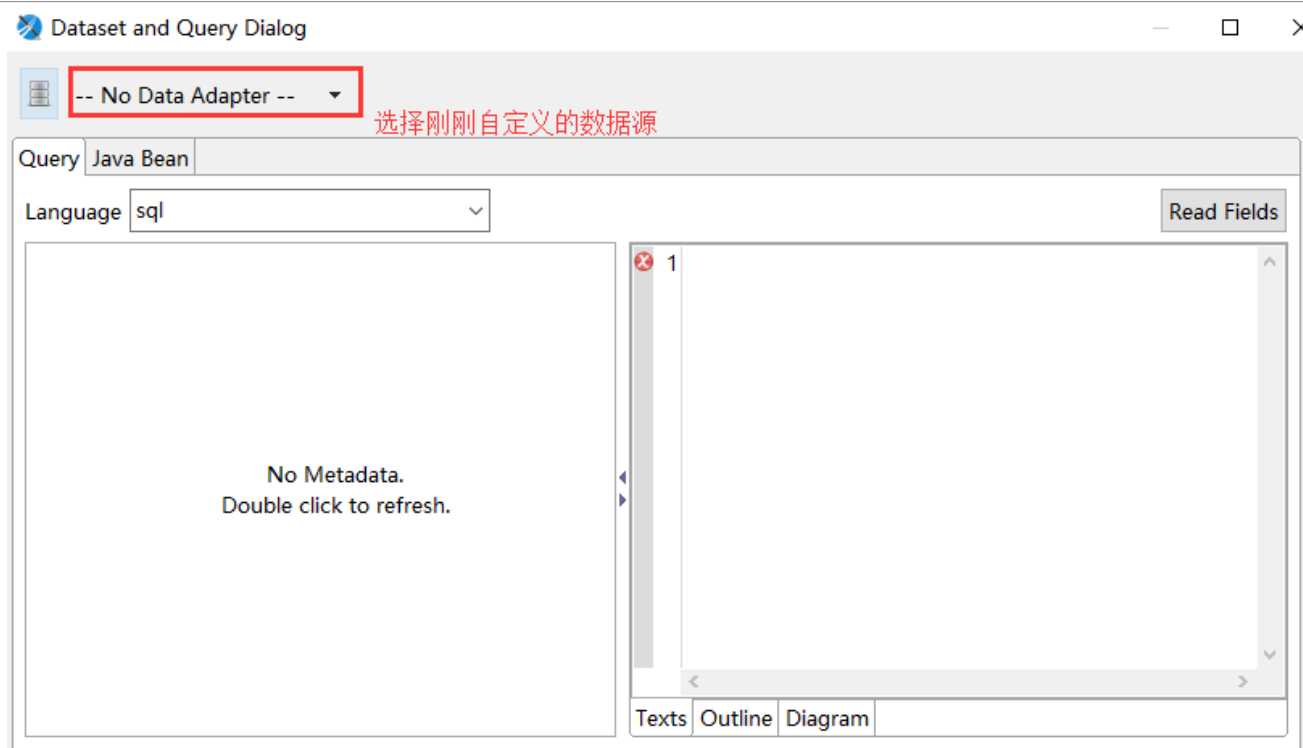
### 1.2.1.2 模板制作

#### （1）制作空白模板

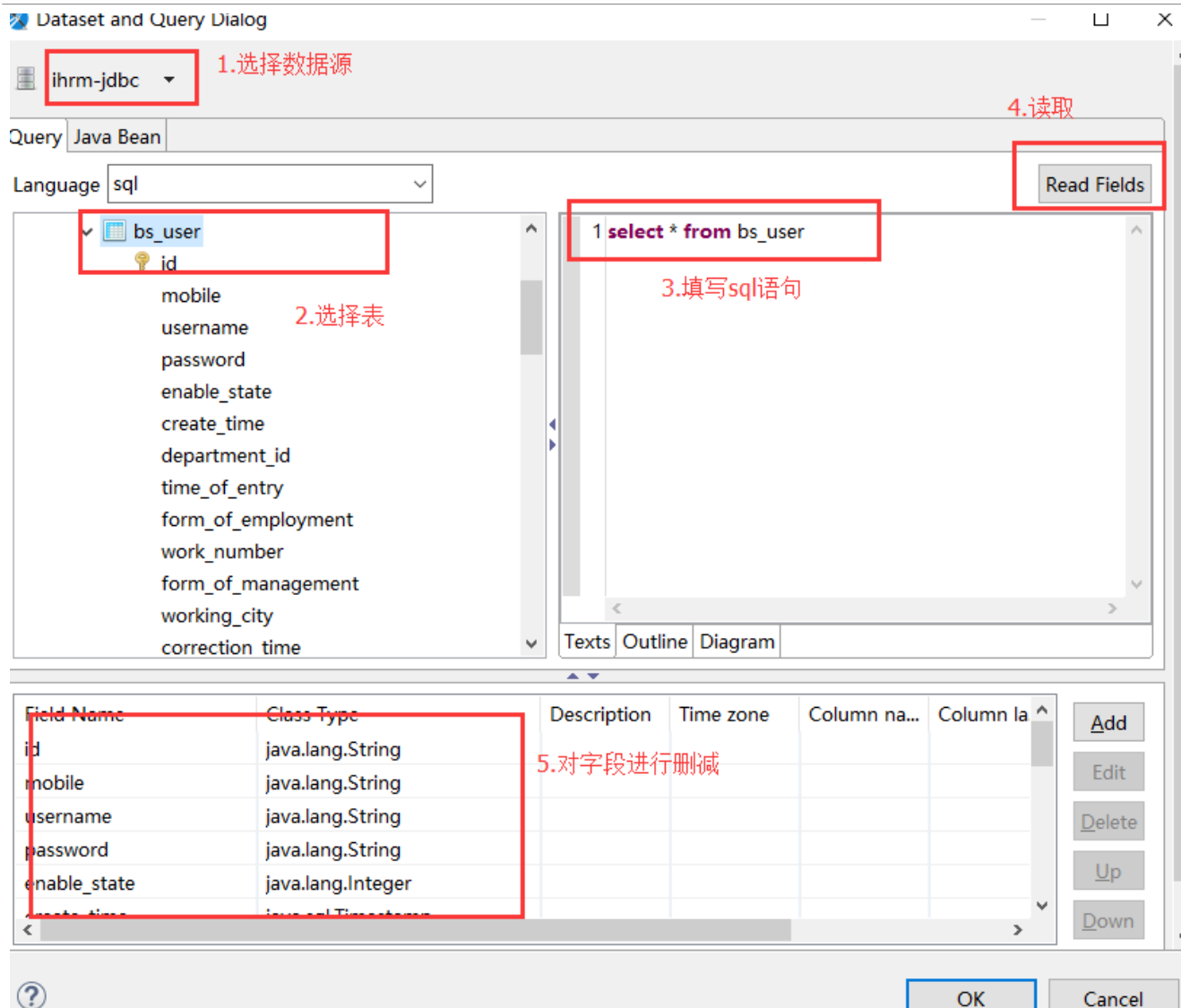
创建空白模板，并将不需要的Band

#### （2）将数据库用户字段配置到模块中

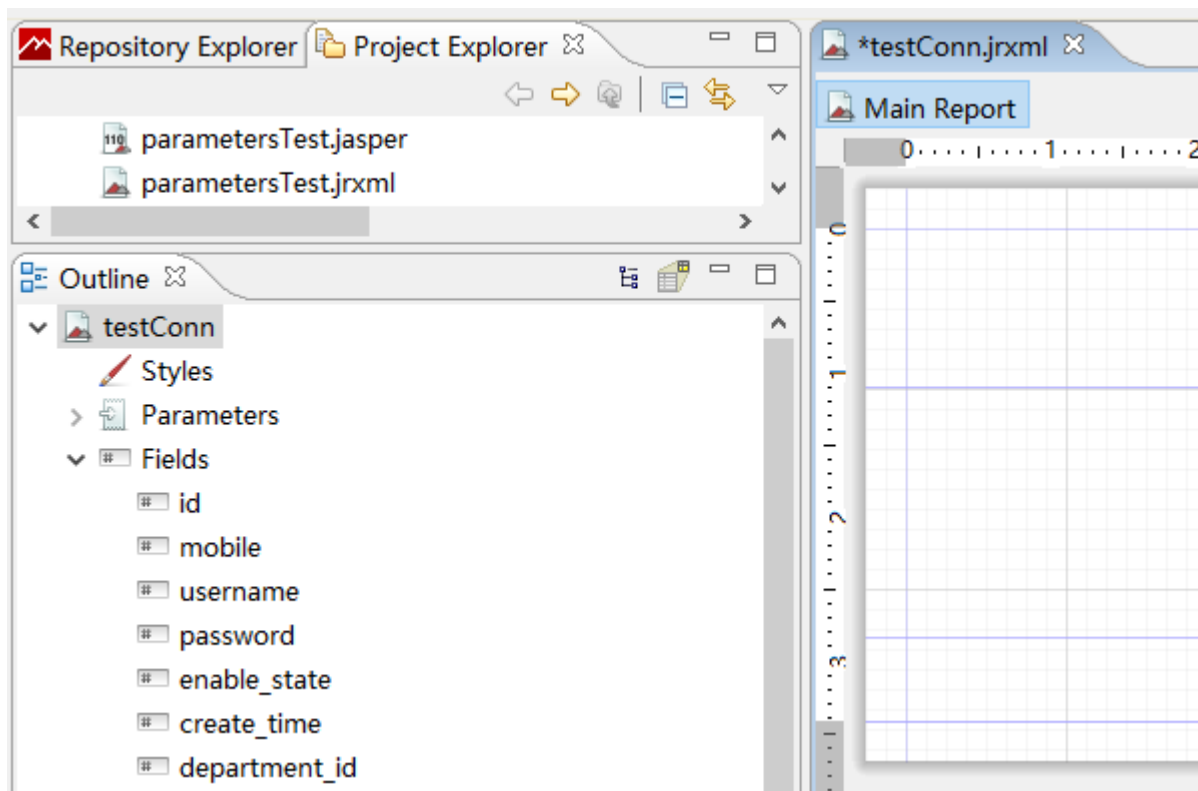
为了方便的进行模板制作，可以将需要数据库表中的字段添加到Studio中。在outline中右键模板，选择dataset and query



用户可以在 SQL 查询语句输入窗口中，输入需要查询数据的查询语句，点击右上角的“Read Fields”按钮，界面下方的字段列表中，就会显示此查询语句中所涵盖的所有字段的列表。在后面的报表设计中，我们就可以直接使用这些字段了。

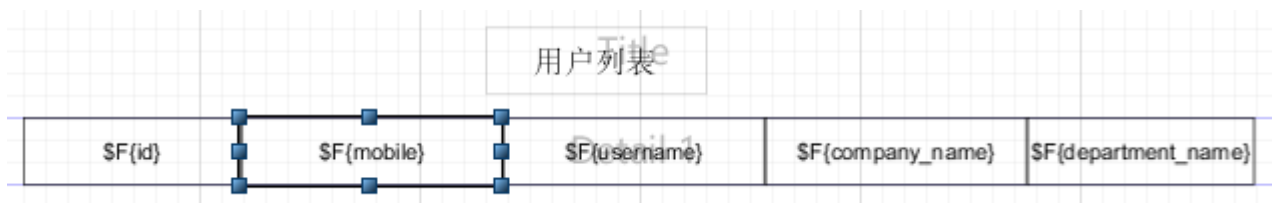


在“Fields”列表中，只保留报表中使用的字段，其他用不到的字段最好用“Delete”删掉，防止由于数据表变化，导致报表模板中的字段设置与数据表对应不上，导致报表报错。输入完毕后，点击“OK”按钮，系统即会把查询语句保存在报表模板中。



### (3) 填充Filed

将id,mobile , username等拖入到 Detail Band中设计模板如下：



### 1.2.1.3 PDF输出

```

//测试JDBC连接数据源
@GetMapping("/testJasper03")
public void createPdf(HttpServletRequest request, HttpServletResponse response)
throws Exception {
    //1.引入jasper文件
    Resource resource = new ClassPathResource("templates/testConn.jasper");
    FileInputStream fis = new FileInputStream(resource.getFile());
    //2.创建JasperPrint,向jasper文件中填充数据

    ServletOutputStream os = response.getOutputStream();
    try {
        /**
         * 1.jasper文件流
         * 2.参数列表
         * 3.数据库连接
         */
        HashMap parameters = new HashMap();
    }
}
    
```





```
JasperPrint print = JasperFillManager.fillReport(fis,
parameters,getConnection());
//3.将JasperPrint已PDF的形式输出
JasperExportManager.exportReportToPdfStream(print,os);
response.setContentType("application/pdf");

} catch (JRException e) {
    e.printStackTrace();
}finally {
    os.flush();
}

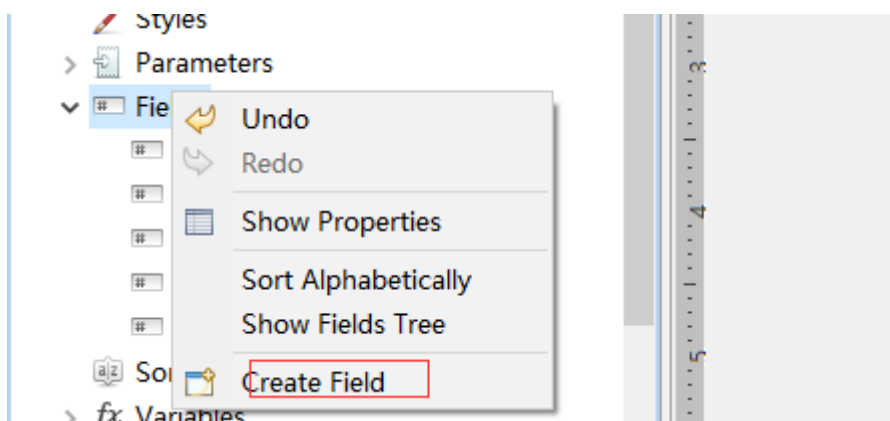
}

//创建数据库Connection
public Connection getConnection() throws Exception {
    String url = "jdbc:mysql://localhost/ihrm";
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn = DriverManager.getConnection(url, "root", "111111");
    return conn;
}
```

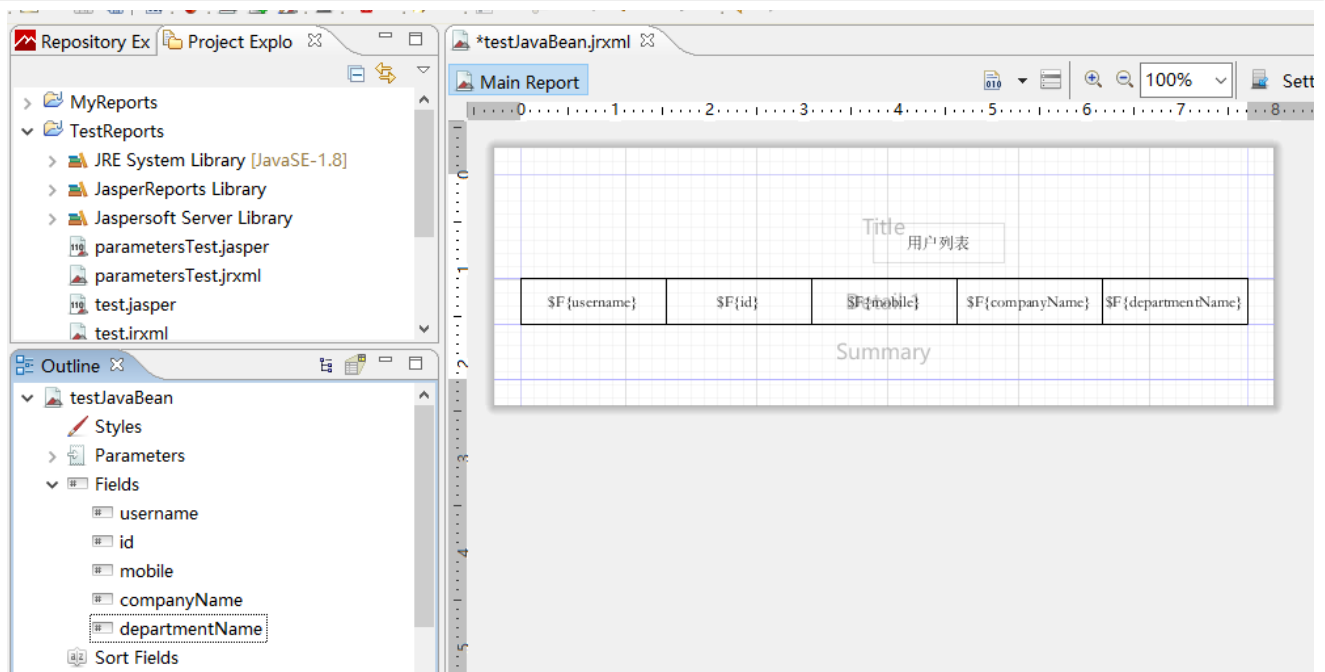
## 1.2.2 JavaBean数据源

### 1.2.2.1 创建Filed

#### (1) 创建Filed



#### (2) 构造模板



## 1.2.2.2 PDF输出

### (1) 配置实体类

```
package cn.itcast.bean;

public class User {
    private String id;
    private String username;
    private String mobile;
    private String companyName;
    private String departmentName;

    public User(String id, String username, String mobile, String companyName, String
departmentName) {
        this.id = id;
        this.username = username;
        this.mobile = mobile;
        this.companyName = companyName;
        this.departmentName = departmentName;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getUsername() {
        return username;
    }
}
```



```
public void setUsername(String username) {
    this.username = username;
}

public String getMobile() {
    return mobile;
}

public void setMobile(String mobile) {
    this.mobile = mobile;
}

public String getCompanyName() {
    return companyName;
}

public void setCompanyName(String companyName) {
    this.companyName = companyName;
}

public String getDepartmentName() {
    return departmentName;
}

public void setDepartmentName(String departmentName) {
    this.departmentName = departmentName;
}
}
```

## (2) 使用javaBean数据源

```
//测试javaBean数据源
@GetMapping("/testJasper04")
public void createPdf(HttpServletRequest request, HttpServletResponse response)
throws Exception {
    //1.引入jasper文件
    Resource resource = new ClassPathResource("templates/testJavaBean.jasper");
    FileInputStream fis = new FileInputStream(resource.getFile());
    //2.创建JasperPrint,向jasper文件中填充数据

    ServletOutputStream os = response.getOutputStream();
    try {
        HashMap parameters = new HashMap();
        //构造javaBean数据源
        JRBeanCollectionDataSource ds = new
        JRBeanCollectionDataSource(getUserList());
        /**
         * 1.jasper文件流
         * 2.参数列表
         * 3.JRBeanCollectionDataSource
         */
        JasperPrint print = JasperFillManager.fillReport(fis, parameters, ds);
    }
```

```
//3.将JasperPrint已PDF的形式输出
JasperExportManager.exportReportToPdfStream(print,os);
response.setContentType("application/pdf");

} catch (JRException e) {
    e.printStackTrace();
}finally {
    os.flush();
}
}

//创建数据库Connection
public List<User> getUserList() throws Exception {
    List<User> list = new ArrayList<>();
    for (int i=1;i<=5;i++) {
        User user = new User(i+"", "testName"+i, "10"+i, "企业"+i, "部门"+i);
        list.add(user);
    }
    return list;
}
```

## 2 分组报表

### 2.1 概述

有两种情况会使用分组报表：

- 美观和好看的显示。
- 当数据分为两层表时，经常需要批量打印子表的数据。打印时，常常需要按照父表的外键或关联值进行自动分组，即每一条父表记录所属的子表记录打印到一组报表中，每组报表都单独计数及计算页数。

在应用中，可以通过选择需要打印的父表记录，将父表记录的 ID 传入，由报表自动进行分组。

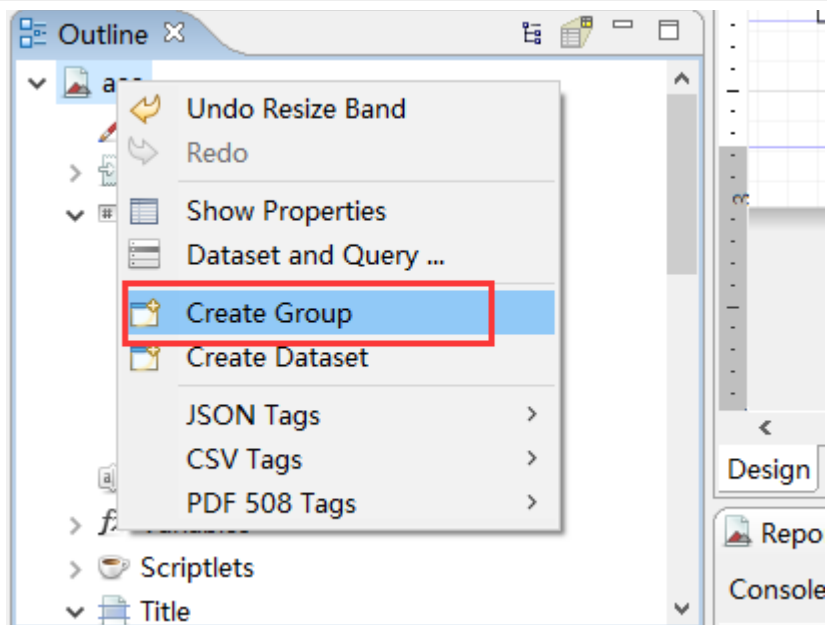
### 2.2 设置分组属性

#### (1) 新建模板

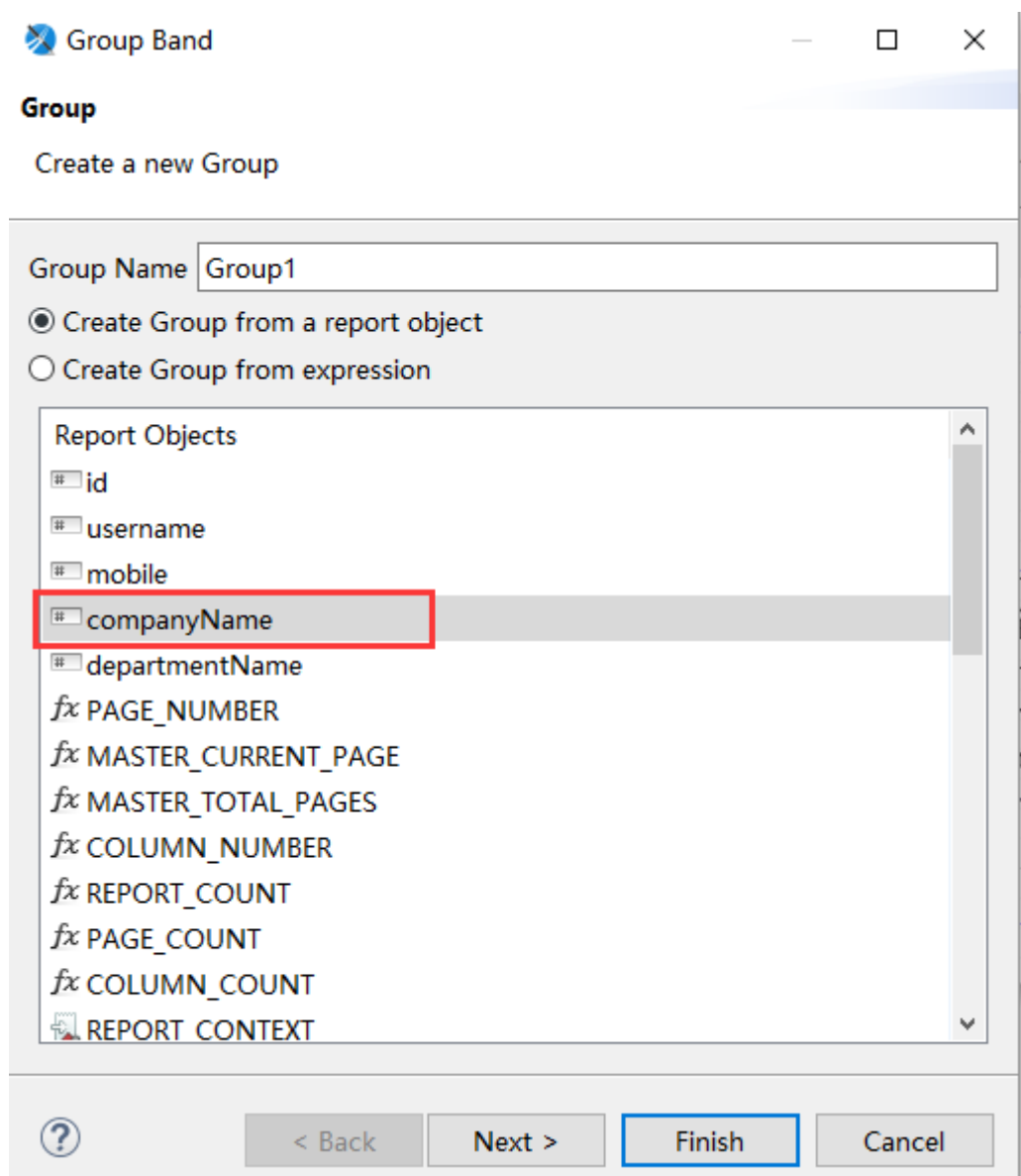
使用用户列表模板完成分组案例

#### (2) 新建报表群组

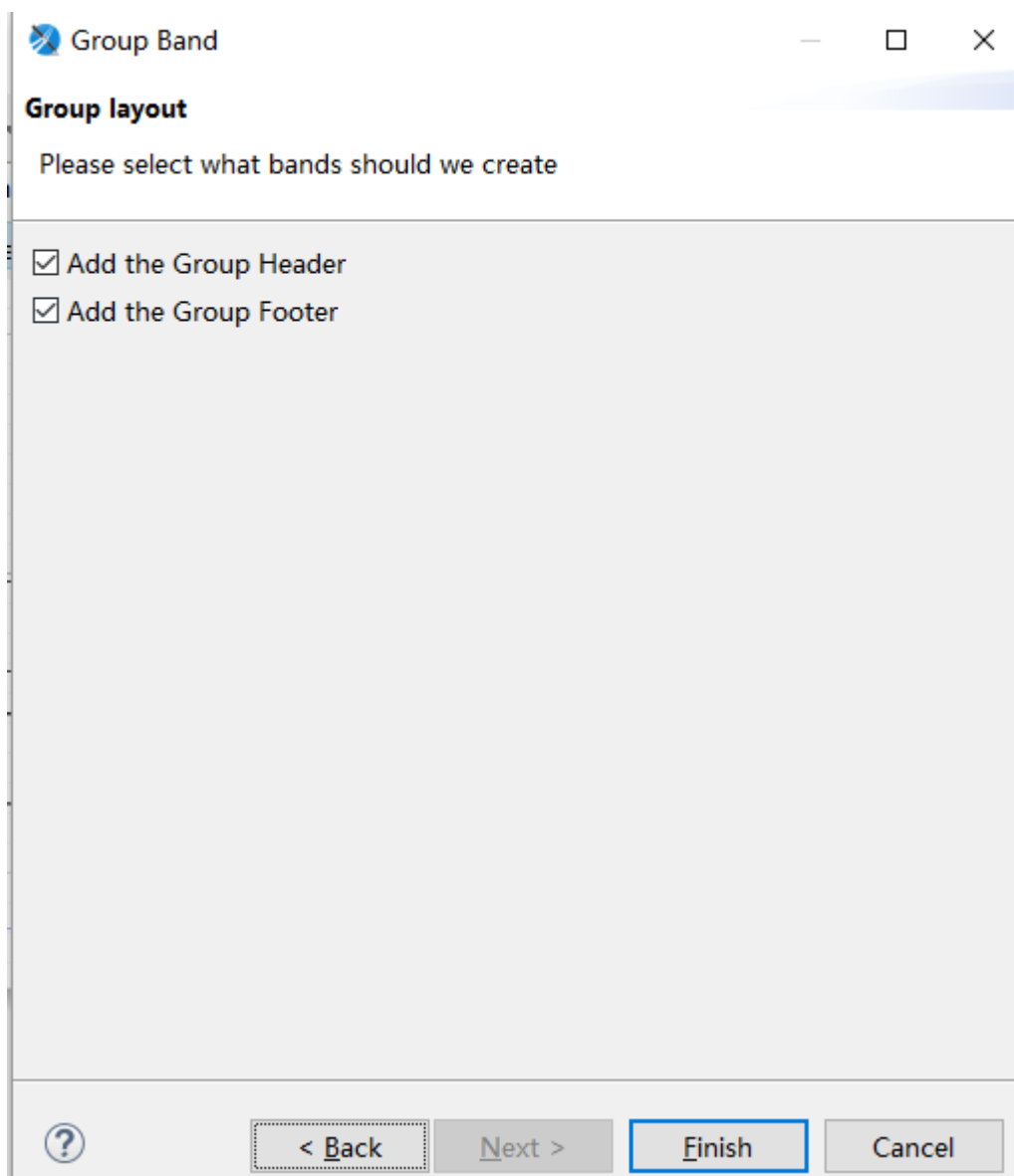
选中报表名称点击右键，选择菜单中的“Create Group”。



需要设置分组的名、分组字段。也可以设置按照指定的函数、方法处理后进行分组

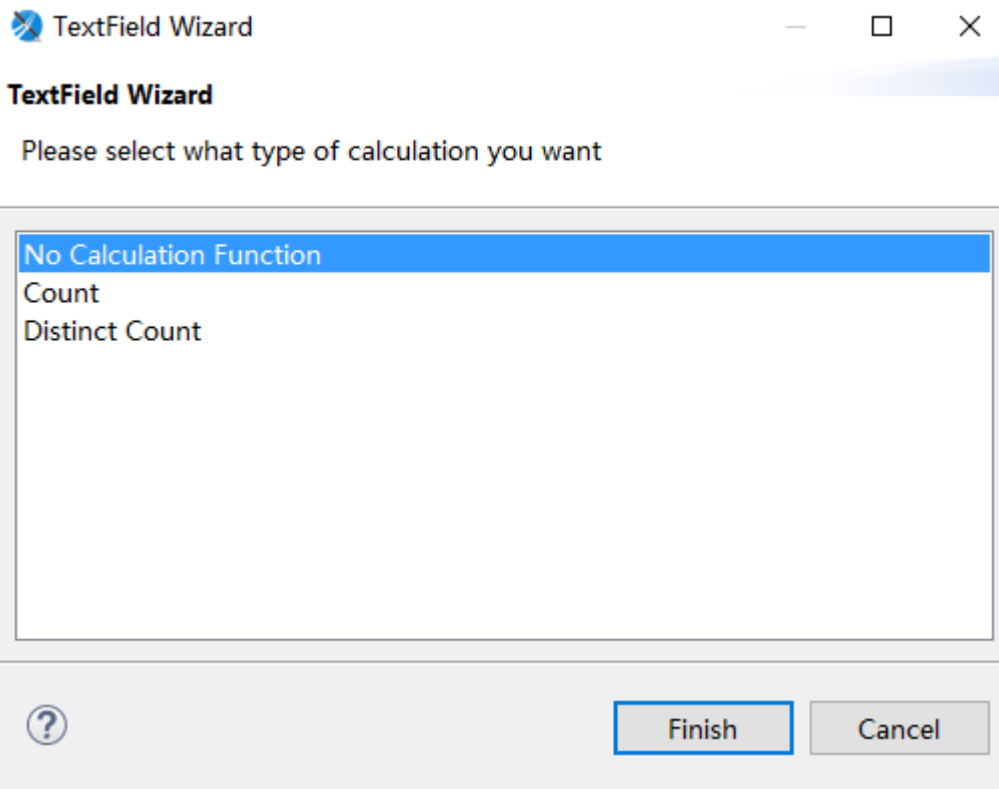


按照字段“companyName”进行分组。设置完毕，点击“Next”。系统显示细节设置界面。此处可以设置是否加入“group header”和“group footer”区。建议保持默认选中，加入这两个区域，这样可以控制在每组报表的结尾，打印相应的信息，例如统计信息等。

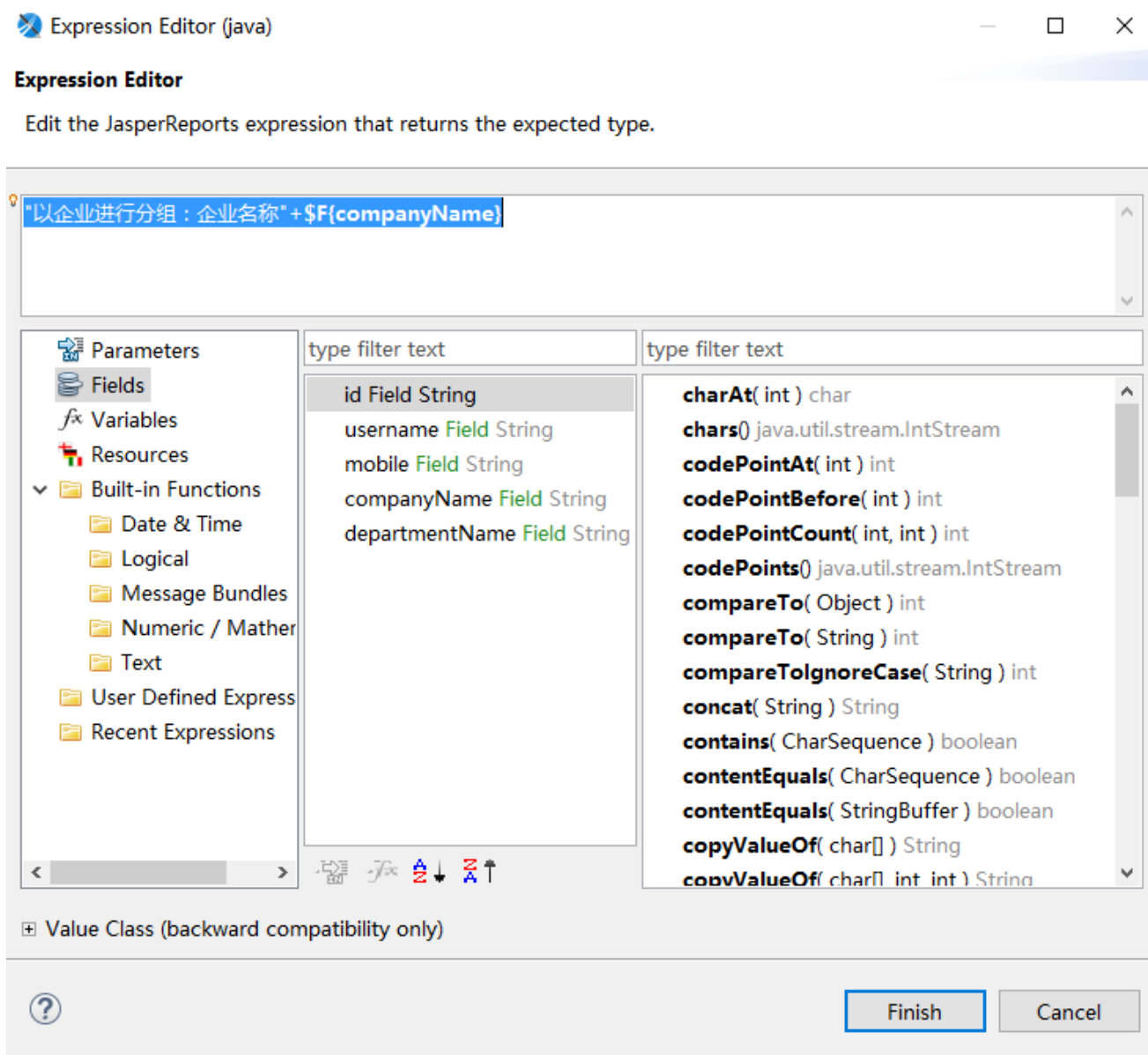


### (3) 放置报表数据

将companyName拖入 Group Header中，会跳出 TextField Wizard框，选中 NoCalculation Function



双击 `$F{deptId}` 会弹出Expression editor框



## 2.3 添加分组Band

将需要作为表头打印的内容拖入 CompanyGroup Header1 栏，将字段拖入 detail 栏，将每个分组结尾需要打印的内容放入 Companygroup footer 栏，将页脚需要打印的内容放入 Page Footer栏，如下图。





用户列表

id	用户名	手机号	所属企业	所属部门
----	-----	-----	------	------

"以企业进行分组：企业名称"+\$F

\$F{id}	\$F{username}	\$F{mobile}	\$F{companyName}	\$F
---------	---------------	-------------	------------------	-----

Company Group Footer 1

"共计："+"\$V{totalNum}

Page Footer

"当前页："+"\$V

## 2.4 PDF输出

```
//测试分组
@GetMapping("/testJasper05")
public void createPdf(HttpServletRequest request, HttpServletResponse response)
throws Exception {
    //1.引入jasper文件
    Resource resource = new ClassPathResource("templates/testGroup.jasper");
    FileInputStream fis = new FileInputStream(resource.getFile());
    //2.创建JasperPrint,向jasper文件中填充数据

    ServletOutputStream os = response.getOutputStream();
    try {
        HashMap parameters = new HashMap();
        //构造javaBean数据源
        JRBeanCollectionDataSource ds = new
JRBeanCollectionDataSource(getUserList());
        /**
         * 1.jasper文件流
         * 2.参数列表
         * 3.JRBeanCollectionDataSource
         */
        JasperPrint print = JasperFillManager.fillReport(fis, parameters,ds);
        //3.将JasperPrint已PDF的形式输出
        JasperExportManager.exportReportToPdfStream(print,os);
        response.setContentType("application/pdf");

    } catch (JRException e) {
        e.printStackTrace();
    } finally {
        os.flush();
    }
}

//创建数据库Connection
public List<User> getUserList() throws Exception {
```

```
List<User> list = new ArrayList<>();
for(int i=1;i<=3;i++) {
    User user = new User("it00"+i, "itcast"+i, "1380000000"+i, "传智播客", "讲师");
    list.add(user);
}

for(int i=1;i<=3;i++) {
    User user = new User("hm00"+i, "itheima"+i, "1880000000"+i, "黑马程序员", "讲师");
    list.add(user);
}

return list;
}
```

效果如下：

用户列表

id	用户名	手机号	所属企业	所属部门
----	-----	-----	------	------

以企业进行分组：企业名称传智播客

it001	itcast1	13800000001	传智播客	讲师
it002	itcast2	13800000002	传智播客	讲师
it003	itcast3	13800000003	传智播客	讲师

共计：3

以企业进行分组：企业名称黑马程序员

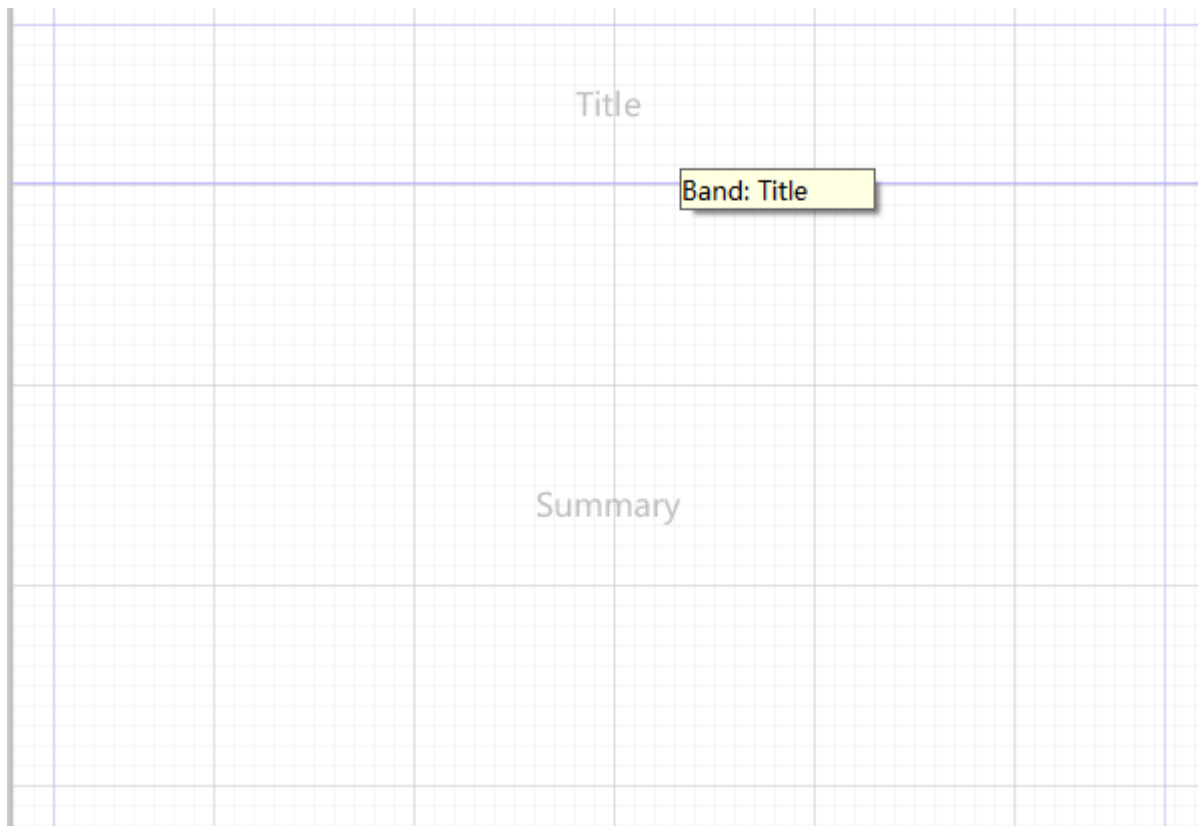
hm001	itheima1	18800000001	黑马程序员	讲师
hm002	itheima2	18800000002	黑马程序员	讲师
hm003	itheima3	18800000003	黑马程序员	讲师

共计：3

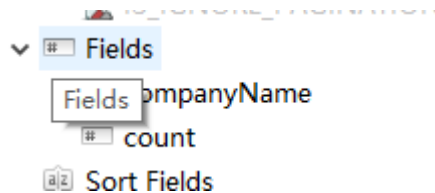
## 3 Chart图表

### 3.1 创建模板

(1) 创建模板，删除不需要的band，保留title和summary。



## (2) 创建fields



## (3) 创建chart图标

第一步：palette面板找到chart图表，拖拽到band中

第二步：选择需要的图表类型



第三步：设置图表参数

Chart Wizard

Chart Data Configuration

Configure how data are used by your chart

Series: Pie series (0) Hyperlink 1.设置series

Value:  ...

Label:  ...

3.设置value和label

Min slice percentage:

Max slices to show:

Other Section

Dataset: [Main Dataset] Return Values Parameters Parameters Map

Increment on: None ...

Reset on: Report ...

Key:  ...

2.设置key

Legend: First Second Third Fourth Fifth

Buttons: < Back Next > Finish Cancel

- Key：圆饼图的内容是什么，也就是下面的 First，Second...的内容
- Value：这个圆饼图的比例依据，根据 Value 属性来显示每个 Key 占的比例
- Label：显示标签

## 3.2 PDF输出

### 3.2.1 实体类

```
public class UserCount {  
    private String companyName;  
    private Integer count;  
  
    public UserCount(String companyName, Integer count) {  
        this.companyName = companyName;  
        this.count = count;  
    }  
  
    public String getCompanyName() {  
        return companyName;  
    }  
  
    public void setCompanyName(String companyName) {  
        this.companyName = companyName;  
    }  
  
    public Integer getCount() {  
        return count;  
    }  
}
```



```
}

    public void setCount(Integer count) {
        this.count = count;
    }
}
```

### 3.2.2 PDF输出

```
//测试图表
@GetMapping("/testJasper06")
public void createPdf(HttpServletRequest request, HttpServletResponse response)
throws Exception {
    //1.引入jasper文件
    Resource resource = new ClassPathResource("templates/testChart.jasper");
    FileInputStream fis = new FileInputStream(resource.getFile());
    //2.创建JasperPrint,向jasper文件中填充数据

    ServletOutputStream os = response.getOutputStream();
    try {
        HashMap parameters = new HashMap();
        //parameters.put("userCountList",getUserList());
        //构造javaBean数据源
        JRBeanCollectionDataSource ds = new
JRBeanCollectionDataSource(getUserList());
        /**
         * 1.jasper文件流
         * 2.参数列表
         * 3.JRBeanCollectionDataSource
         */
        JasperPrint print = JasperFillManager.fillReport(fis, parameters,ds);
        //3.将JasperPrint已PDF的形式输出
        JasperExportManager.exportReportToPdfStream(print,os);
        response.setContentType("application/pdf");

    } catch (JRException e) {
        e.printStackTrace();
    }finally {
        os.flush();
    }
}

//创建数据库Connection
public List<UserCount> getUserList() throws Exception {
    List<UserCount> list = new ArrayList<>();
    UserCount uc1 = new UserCount("传智播客",10);
    UserCount uc2 = new UserCount("黑马程序员",10);
    list.add(uc1);
    list.add(uc2);
    return list;
}
```

## 4 父子报表

### 4.1 概述

复杂报表或数据内容较多的时候，可以使用子报表解决。

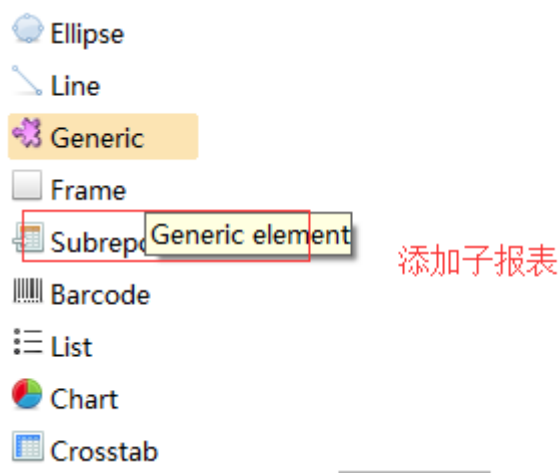
### 4.2 制作父报表

首先制作父报表，就是调用子报表的一个基础报表。主报表的作用有如下两种：

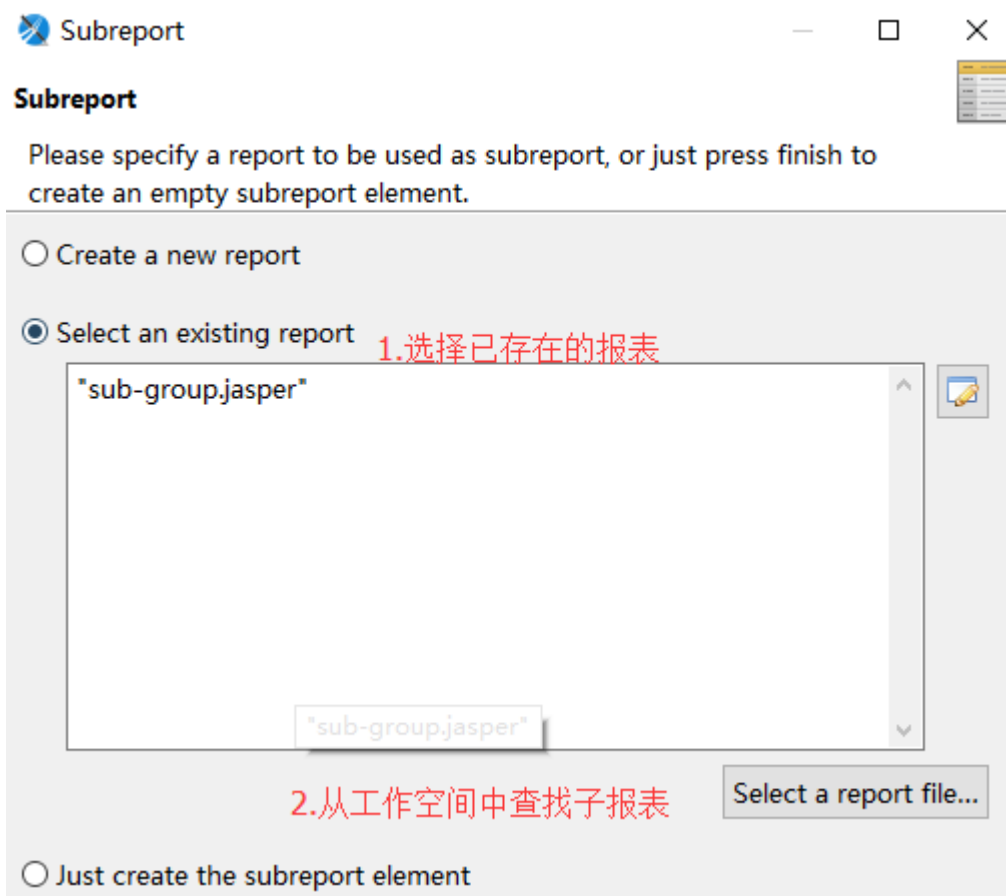
- 父报表中需要显示数据，使用子报表弥补studio设计的不足
- 父报表不需要显示任何数据，只是作为子报表的载体。适用于复杂报表的设计

### 4.3 制作子报表

点击组件面板上的“Subreport”按钮，拖动到报表工作区上。

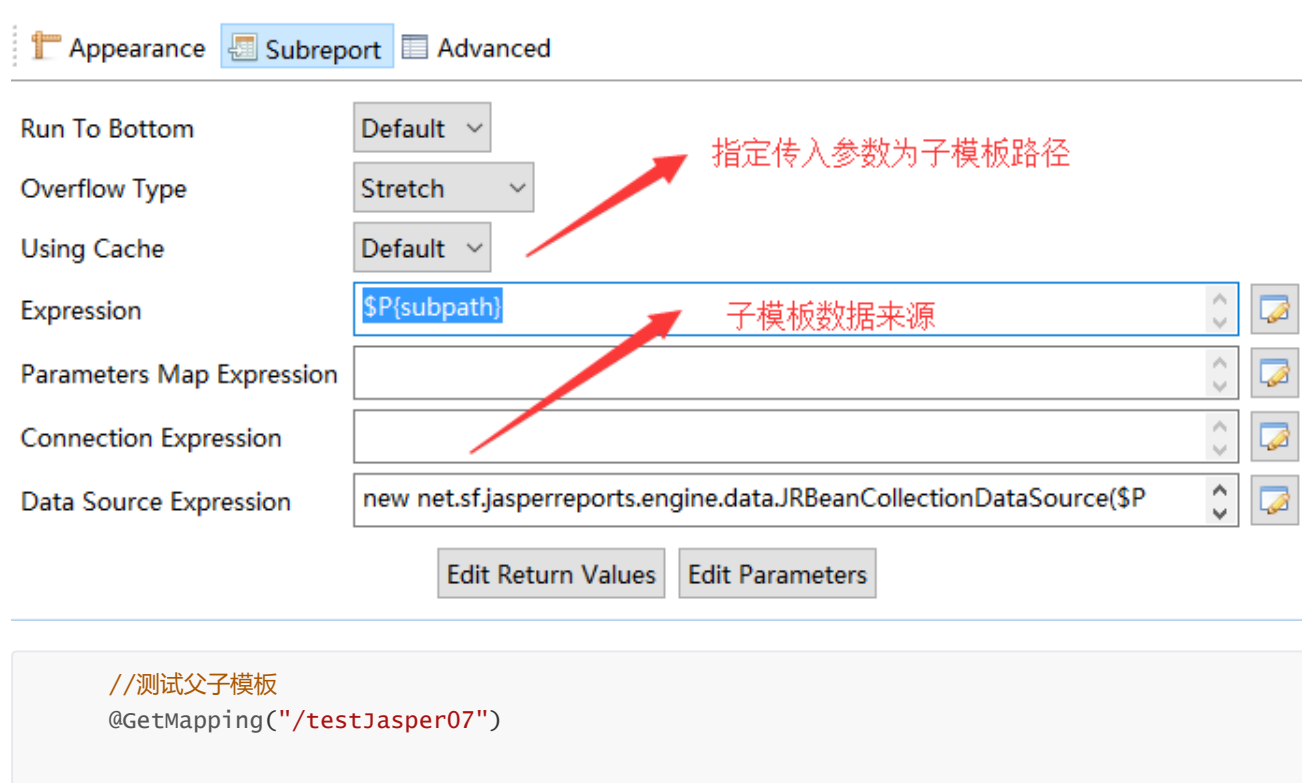


系统会自动弹出子报表选择窗口。可以选择创建一个新报表，还是使用一个已有的报表作为子报表。



选择“Create a new report”，可以立即制作新的子报表；如果选择“Select an existing report”，则可以调用已有的报表作为子报表；如果选择“Just create the subreport element”，系统会生成一个子报表区，可以在之后挂接需要的子报表。

## 4.4 参数传递





```
public void createPdf(HttpServletRequest request, HttpServletResponse response)
throws Exception {
    //1.引入jasper文件
    Resource resource = new ClassPathResource("templates/main.jasper");
    FileInputStream fis = new FileInputStream(resource.getFile());
    //2.创建JasperPrint,向jasper文件中填充数据

    ServletOutputStream os = response.getOutputStream();
    try {
        HashMap parameters = new HashMap();
        Resource subResource = new ClassPathResource("templates/sub-group.jasper");
        parameters.put("subpath",subResource.getFile().getPath());
        parameters.put("sublist",getUserList());
        parameters.put("sublist",getUserList());
        JRBeanCollectionDataSource ds = new
JRBeanCollectionDataSource(getUserList());
        JasperPrint print = JasperFillManager.fillReport(fis, parameters,new
JREmptyDataSource());

        //3.将JasperPrint已PDF的形式输出
        JasperExportManager.exportReportToPdfStream(print,os);
        response.setContentType("application/pdf");

    } catch (JRException e) {
        e.printStackTrace();
    }finally {
        os.flush();
    }
}

//创建数据库Connection
public List<User> getUserList() throws Exception {
    List<User> list = new ArrayList<>();
    for(int i=1;i<=3;i++) {
        User user = new User("it00"+i, "itcast"+i, "1380000000"+i, "传智播客", "讲
师");
        list.add(user);
    }

    for(int i=1;i<=3;i++) {
        User user = new User("hm00"+i, "itheima"+i, "1880000000"+i, "黑马程序员", "讲
师");
        list.add(user);
    }

    return list;
}
```

## 5 用户档案下载



## 5.1 搭建环境

### (1) 配置坐标

```
<dependency>
    <groupId>net.sf.jasperreports</groupId>
    <artifactId>jasperreports</artifactId>
    <version>6.5.0</version>
</dependency>
<dependency>
    <groupId>org.olap4j</groupId>
    <artifactId>olap4j</artifactId>
    <version>1.2.0</version>
</dependency>
<dependency>
    <groupId>com.lowagie</groupId>
    <artifactId>itext</artifactId>
    <version>2.1.7</version>
</dependency>
```

### (2) 解决乱码问题

```
net.sf.jasperreports.extension.registry.factory.simple.font.families=net.sf.jasperreports.engine.fonts.SimpleFontExtensionsRegistryFactory
net.sf.jasperreports.extension.simple.font.families.lobstertwo=stsong/fonts.xml
```

## 5.2 实现用户档案下载

```
/**
 * 打印员工pdf报表x
 */
@RequestMapping(value="/{id}/pdf",method = RequestMethod.GET)
public void pdf(@PathVariable String id) throws IOException {
    //1.引入jasper文件
    Resource resource = new ClassPathResource("templates/profile.jasper");
    FileInputStream fis = new FileInputStream(resource.getFile());

    //2.构造数据
    //a.用户详情数据
    UserCompanyPersonal personal = userCompanyPersonalService.findById(id);
    //b.用户岗位信息数据
    UserCompanyJobs jobs = userCompanyJobsService.findById(id);
    //c.用户头像      域名 / id
    String staffPhoto = "http://pkbivgfrm.bkt.clouddn.com/"+id;

    System.out.println(staffPhoto);

    //3.填充pdf模板数据，并输出pdf
```



```
Map params = new HashMap();

Map<String, Object> map1 = BeanMapUtils.beanToMap(personal);
Map<String, Object> map2 = BeanMapUtils.beanToMap(jobs);

params.putAll(map1);
params.putAll(map2);
params.put("staffPhoto", "staffPhoto");

ServletOutputStream os = response.getOutputStream();
try {
    JasperPrint print = JasperFillManager.fillReport(fis, params, new
JREmptyDataSource());
    JasperExportManager.exportReportToPdfStream(print, os);
} catch (JRException e) {
    e.printStackTrace();
}finally {
    os.flush();
}
}
```