## 安装演示版

1.将mr.war解压或者直接放到应用服务器目录下，比如,tomcat放到tomcat/webapps下面。

2. 启动tomcat。

3. 在浏览器端输入：

设计一个新报表,报表名称为 myreport1,数据源为 order,数据源后述.

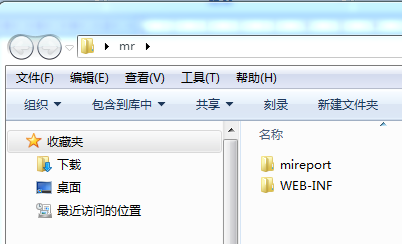
http://127.0.0.1:8099/mr/mireport/designer/r.jsp?ds=order&report= myreport1

修改报表 myreport1

http://127.0.0.1:8099/mr/mireport/designer/r.jsp?report= myreport1

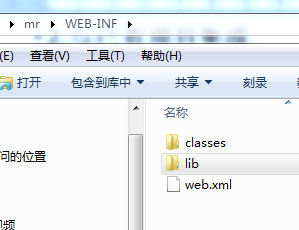
## 与已有项目集成

1.将mr.war解压，目录如下：

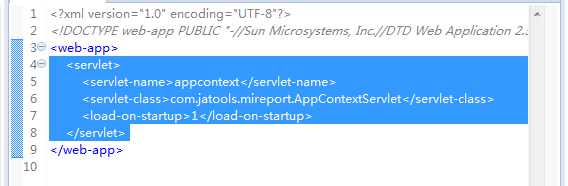


1.将mireport目录复制到你的应用目录下

2.将 WEB-INF 下的 classes和lib目录，复制到你的应用 WEB-INF目录下



3.打开 web.xml,将其中的唯一的<servlet>复制到你的应用web.xml中



4. 重启你的应用服务器。

3. 在浏览器端输入：

设计一个新报表,报表名称为 myreport1,数据源为 order.

http://127.0.0.1:8099/yourApp/mireport/designer/r.jsp?ds=order&report=myreport1

提示：

a.这里的数据源ds是必选的，当你决定设计一张报表时，必须先定好针对哪个数据源设计报表。

b.report参数，是当用户点击保存按钮时，就以report参数作为报表名称保存。

c.当你指定的myreport1，已经存在时，则系统认为你是想修改这个报表，因此，设计器的页面设置不会弹出，并且将该报表加载进来。

修改报表 myreport1

http://127.0.0.1:8099/yourApp/mireport/designer/r.jsp?report= myreport1

## 报表设计流程

#### 数据源描述文件准备

设计报表，首先准备数据源。报表数据源用一个json文件描述，以下是一个典型的数据源描述文件：

order.json

{

dataset : {

"default" : {

id : "151201"

},

factory : "com.jatools.mireport.JDBCDatasetFactory",

jdbc : "demo1",

master : {

name : "订单记录",

sql : "select 订单号,订单日期,客户,运货地址,业务员,订单状态 from orders where 订单号='${id}'"

},

detail : {

name : "订单明细",

sql : "select ID,产品,型号,单价,数量,金额 from details where 订单号='${订单号}'"

}

},

ui : {

defaults : {

"text-editable" : **false //text列在设计器时，拖放后的text是否可以编辑，此值默认为true**

},

fields : [{

type : 'label',

text : '+'

}, {

type : 'text',

field : '订单号',

editable : **true // text是可编辑时，此处设置后将忽略 ui.defaults[**"text-editable"**]里的设置**

}, {

type : 'text',

field : '订单日期'

}, {

type : 'text',

field : '客户'

}, {

type : 'text',

field : '运货地址'

}, {

type : 'text',

field : '业务员'

}, {

type : 'text',

field : '订单状态'

}, {

type : 'table',

options : {

"no-border" : **false**,

"header-rows" : 1,

"body-max-rows" : 6,

"footer-rows" : 1

},

dataset : "订单明细",

fields : [{

type : 'text',

field : 'ID',

display : "序号" // 界面上，以display，显示ID字段，没有定义display，则显示field

}, {

type : 'text',

field : '产品',

editable : **true**

}, {

type : 'text',

field : '型号'

}, {

type : 'text',

field : '单价'

}, {

type : 'text',

field : '数量'

}, {

type : 'text',

field : '金额'

}]

}]

}

}

数据源，主要描述：

a. 数据来源。

包括主、从数据集的sql查询，在测试报表时，用到的缺省参数。

b. 设计报表时的可用字段、表格

#### 数据源描述文件保存

a.准备好数据源描述文件后，可以保存在应用目录 mireport/services/datasources下面。

b.描述文件名必须以 .json为扩展名，尽量不要用中文名.

#### 引用数据源描述

在设计报表时，必须指定数据源文件，比如

http://127.0.0.1:8099/yourApp/mireport/designer/r.jsp?ds=order&report= myreport1

这里面的 ds参数，即是 order, 说明这个报表设计时，需要引用 :

mireport/services/datasources/order.json 这个数据源描述文件，设计器就会从该文件的描述中，找出可拖放字段，列在设计界面上。

## 调用报表打印及预览

#### 在你自己的页面上打印报表

现在，你已经有了数据源定义，也设计好了报表模板，比如 myreport1，如何在你自己的页面上，进行打印，或打印预览？下面是一个示例：

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<script type=*"text/javascript"*

src=*"mireport/designer/js/jatoolsPrinter.js"*></script>

<script type=*"text/javascript"*>

**function** doPrint(how) {

**var** myDoc = {

settings : 'auto',

marginIgnored : **true**,

documents : "mireport/service/buildPage.jsp?report=myreport1",

copyrights : '杰创软件拥有版权 www.jatools.com'

};

**var** jatoolsPrinter = getJatoolsPrinter();

**if** (how == '打印预览...')

jatoolsPrinter.printPreview(myDoc, **false**);

**else** **if** (how == '打印...')

jatoolsPrinter.print(myDoc, **true**);

**else**

jatoolsPrinter.print(myDoc, **false**);

}

</script>

</head>

<body>

<H2>打印mireport</H2>

<input type=*"button"* value=*"打印预览..."* onClick="doPrint('打印预览...')">

<input type=*"button"* value=*"打印..."* onClick="doPrint('打印...')">

<input type=*"button"* value=*"打印"* onClick="doPrint('打印')">

<!-- 声明打印控件 -->

<object id=*"ojatoolsPrinter"*

codebase=*"jatoolsPrinter.cab#version=5,4,0,0"*

classid=*"clsid:B43D3361-D075-4BE2-87FE-057188254255"* width=*"0"*

height=*"0"*>

<embed id=*"ejatoolsPrinter"* type=*"application/x-vnd.jatoolsPrinter"*

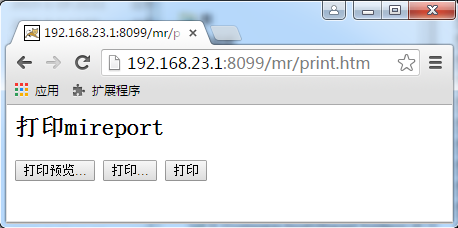
width=*"0"* height=*"0"* />

</object>

</body>

</html>

上述页面显示效果：



1. 必须引入 mireport/designer/js/jatoolsPrinter.js

2. 在</body> 前，声明控件 <object ...>...</object>

3. 设置打印按钮和调用打印控件方法

控件有两个方法，第一个参数都是同一类型 myDoc，第二个参数下述。其他属性照抄，其中需要设置的属性是documents，它是一个字符串类型的值，它用来指出你需要打印哪个报表，如下所示：

**var** myDoc = {

...

documents : "mireport/service/buildPage.jsp?report=myreport1",

...

};

表示，你想打印myreport1报表。

#### 控件的printPreview、print方法

控件有两种方法：,

printPreview(myDoc,progress)，打印预览

myDoc打印设置对象;

progress指出，在预览页生成时，是否显示进度条，true显示/false不显示

建议：仅在页数较多时显示。

print(myDoc,prompt)，打印

myDoc打印设置对象;

prompt指出，打印前是否弹出打印机对话，true弹出/false不弹出

两种方法，都接受一个myDoc打印设置对象，如：

**var** myDoc = {

settings : 'auto',

marginIgnored : **true**,

documents : "mireport/service/buildPage.jsp?report=myreport1",

copyrights : '杰创软件拥有版权 www.jatools.com'

};

其他属性不要更改，documents需要设置。documents属性是一个字符串类型的值，它用来指出你需要打印哪个报表。mireport/service/buildPage.jsp 负责页面生成，report参数指出生成哪个报表的页面，等等，那这里只有一个参数，我想传更多的参数，来控制页面生成，比如，只想打印某一月份的订单，怎么办呢？下面就讲这个。

#### 用参数控制报表打印方法

比如，有一个订单系统，允许用户在打印时，选择月份进行打印，如何做？

order.json

{

dataset : {

**default**:{month:"2012-10"}, // 默认参数

jdbc : {...},

master : {

name : "订单记录",

sql : "select 订单号,订单日期,客户,运货地址,业务员,订单状态 from orders where substr(订单日期,0,7)='${ month }'"

},...

}

报表有两个地方会调用 mireport/service/buildPage.jsp来生成页面：

a.设计器中打印测试，如下：



这个时候，没有地方可以传入参数，所以对 dataset.master.sql查询时，使用的是 dataset.default里面的默认参数。本例中，设计器打印测试时浏览到的数据是2012年10月的。

b.在自己的页面中调用报表打印，或者报表预览。

**var** myDoc = {

...

documents : "mireport/service/buildPage.jsp?report=myreport1&month=2015-10", //指定打印2015-10月份数据

// 下面没有指定month参数，则报表仍然用默认参数打印，即2012-10

//documents : "mireport/service/buildPage.jsp?report=myreport1”

...

};

getJatoolsPrinter().print(myDoc,false);

## 自定义数据集

系统提供多种方法，来获取报表数据，你可以使用 jdbc数据源，也可以是你自己的java类集合中的数据。

先看一个示例：

array.json

{

dataset : {

"default" : {

id : "151201"

},

factory : "com.jatools.mireport.SimpleDatasetFactory",

master : {

name : "订单记录"

},

detail : {

name : "订单明细"

}

},

数据源必须使用 dataset.factory属性，来指定数据集工厂，系统在生成报表时，会实例化该属性指定的类，该类必须实现接口：

com.jatools.mireport.DatasetFactory

**public** **interface** DatasetFactory {

// 创建数据集，在生成报表页面时调用

// context: 上下文，可以取得web上传入的参数

// datasource: datasource的整个json定义

// datasetName: 要取提数据集名称，定义在 dataset.master.name,或者 dataset.detail.name

Dataset create(ReportContext context, Datasource datasource, String datasetName);

}

该类实例化后，会依次调用 create 方法，生成和主从数据集，参数datasource ，datasetName也会传入。

注意：

1.datasource里面的属性，可能会根据不同的数据集工厂，使用不同的属性。比如，系统默认实现JDBCDatasetFactory会使用jdbc、sql属性，如果你从java集合中、数组中取得，这些属性就不必。

总之，DatasetFactory派生类需要多少属性，完全可以自己定义，自己解释。

2. 但有些属性，是框架使用，是必须的：dataset.factory、dataset.master.name、dataset.detail.name

## 自定义报表文件存储

默认情况下，用户设计的报表，将被存储在服务器目录 mireport\service\reports下面，但如果你想将报表存储在数据库，将如何设置？

报表框架使用接口，com.jatools.mireport.ReportManager 来处理这个问题：

**public** **interface** ReportManager {

// 保存报表模板，在点击设计器的保存按钮时调用，

// name：报表名称

// content: 报表模板内容

**public** **void** save(String name, String content);

//载入报表模板，在页面生成及设计器修改报表时调用，

// name：报表名称

// 返回: 报表模板内容

**public** Report load(String name);

}

系统提供了一个该接口的默认实现类，com.jatools.mireport.DefaultReportManager,该类的作用就是在mireport\service\reports下存取报表模板。

你可以作一个实现该接口的类，并将你的类名，替换到配置文件中 WEB-INF\mireport-config.json中

{

"reportManager" : "com.jatools.mireport.DefaultReportManager",

"datasourceManager" : "com.jatools.mireport.DefaultDatasourceManager",

"imageManager" : "com.jatools.mireport.DefaultImageManager"

}

## 自定义数据源读取

系统有一个配置文件，WEB-INF\mireport-config.json，里面有一个datasourceManager如属性，定义系统的数据源从什么地方取得，默认是类com.jatools.mireport.DefaultDatasourceManager，如下：

WEB-INF\mireport-config.json

{

"reportManager" : "com.jatools.mireport.DefaultReportManager",

**"datasourceManager" : "com.jatools.mireport.DefaultDatasourceManager",**

"imageManager" : "com.jatools.mireport.DefaultImageManager"

}

DefaultDatasourceManager实现接口com.jatools.mireport.DatasourceManager

**public** **interface** DatasourceManager {

// 载入数据源定义，在打开设计器，及报表页面生成时调用

// name：数据源名称 参见本节注

// 返回: 数据源定义内容

**public** Datasource load(String name);

}

DefaultDatasourceManager作用是，系统从 mireport\service\datasources目录下取得数据源定义

注：

a.当你打开设计器r.jsp?ds=mydatasource&report=myreport，新设计一个报表时，系统就会调用本接口的load方法，即相当于load("mydatasource"),找到该数据源，再将可用字段显示在页面上。

b.当你修改一个报表时，系统会根据保存在报表模报内的datasource名称，调用 load方法找到数据源。

## 自定义上传图片保存

默认情况下，用户上传的图片文件 ，被保存在 mireport\service\images下面，保存动作通过 com.jatools.mireport.DefaultImageManager类实现，该类被设置在系统配置文件的imageManager属性中

WEB-INF\mireport-config.json

{

"reportManager" : "com.jatools.mireport.DefaultReportManager",

"datasourceManager" : "com.jatools.mireport.DefaultDatasourceManager",

**"imageManager" : "com.jatools.mireport.DefaultImageManager"**

}

类DefaultImageManager实现接口com.jatools.mireport.mageManager，你可以实现自己的类，并替换到imageManager中。

## 用iframe集成报表设计器

如果你不想让设计器另起一个页面，那么，你可以在你的页面布局里，配置一个iframe，将iframe.src指向设计器 url就可以了。

如下所示：

<!DOCTYPE html>

<html>

<head>

<title>MiReport 报表在frame中的集成演示</title>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<script>

// 设计器 r.jsp 有两个方法，saveReport,testReport 。两按钮分别调用两个方法

**function** saveReport() {

document.getElementById("designer").contentWindow.saveReport({

"type" : "sales"

}, **function**(data) {

**if** (data.success) {

alert("保存成功!")

} **else**

alert("保存失败！");

})

}

**function** testReport() {

document.getElementById("designer").contentWindow.testReport();

}

</script>

</head>

<body style="position: *relative*;">

<button onclick="saveReport()">保存</button>

<button onclick="testReport()">打印预览</button><br><br>

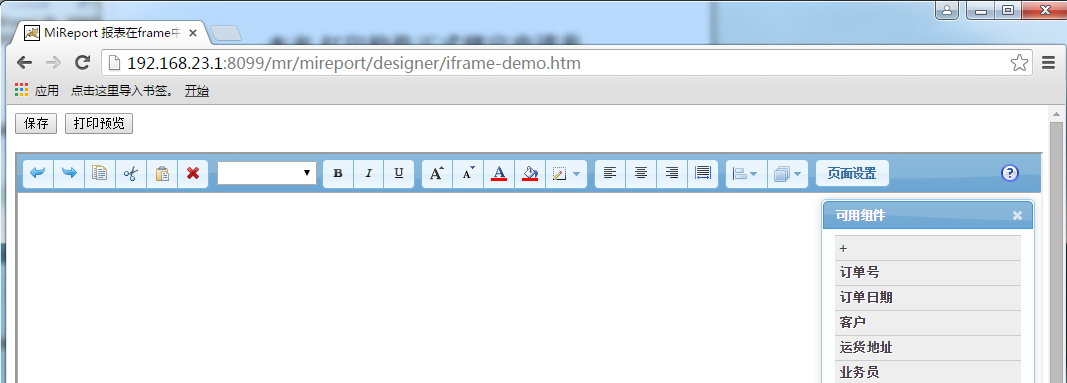
<!-- 可以用nobuttons控制，哪些按钮不可见，以jquery selector形式一次指定多个 -->

<iframe id=*"designer"* style="width: *1024px*; height: *768px*;"

src=*"r.jsp?ds=order&nobuttons=.jp-upload-template,.jp-test-print"*></iframe>

</body>

</html>



在这里，设置了两个按钮，同时隐藏了原来工具栏上默认配置的保存、打印测试这两个按钮，实际上，把两个按钮移到了iframe外面，如何做到？

#### 在自己的页面中，调用设计器中的方法

在本例中，页面中设置了两个按钮：保存、打印测试。这两个按钮分别调用在r.jsp中定义了两个全局方法：

saveReport(object,callback)

object ：saveReport方法最终会调用服务端的 save方法

ReportManager. save(String name, String content,Map<String,String> params)

saveReport里的object参数，在save方法里，变成 params.

callback：回调函数，该回调的唯一参数boolean类型，指明保存是否成功，true/false=成功/失败。

testReport()

启动设计器的打印预览窗口。

注：我们可以根据用户的需要，定制设计器全局方法，或者回调，以便更好地控制设计器

#### 隐藏设计器中的按钮

本例中，隐藏了保存、打印测试按钮，如何做？

r.jsp?ds=order&**nobuttons**=.jp-upload-template,.jp-test-print

如上所示

你只需要在 src指出的url中，带上nobuttons参数就行了，在r.jsp中，查找到某个要隐藏的 button,把它的class,加在nobuttons参数中，多个button以逗号分割。