

## 9.与业务结合

UReport2教学视频  
<http://pan.baidu.com/s/1boWTxF5>, 密码: 98hj

UReport2 QQ群  
UReport2的QQ群号: 423339793


本小节中我们将介绍如何在业务页面中展示报表、导出报表文件、引用报表内容等操作。

### 报表相关URL

在配置好一个包含UReport2的项目后, 就可以运行我们的项目, 打开UReport2的基于网页的报表设计器, 它的地址如下:

UReport2报表设计器URL  
[http://host\[:port\]/\[context-path\]/ureport/designer](http://host[:port]/[context-path]/ureport/designer)


如果我们在本地启动项目, 并且服务器端口号为8080, context-path为ureport2-demo, 那么我们的报表设计器的URL就是: <http://localhost:8080/ureport2-demo/ureport/designer>。

在报表设计器中, 设计好报表后, 点击左上角的  图标就可以在线预览报表 (预览报表有两种方式, 一种是不分页预览报表; 一种是分页预览报表, 如果报表数据量较大, 我们推荐预览时采用分页方式, 这样可以减轻浏览器渲染压力); 预览报表的URL如下:

UReport2预览报表URL  
[http://host\[:port\]/\[context-path\]/ureport/preview?\\_u=报表名称](http://host[:port]/[context-path]/ureport/preview?_u=报表名称)

在上面的URL中有个名为“\_u”的参数, 它是用来指定当前要预览的报表模版名称, 如果是对正在设计器中的报表进行预览, 那么它的值就是“p”, 这是一个约定的值, 表示要预览的是正在设计器中的报表, 所以可以看到我们演示时预览的URL都是这样: [http://localhost:8080/ureport2-demo/ureport/preview?\\_u=p](http://localhost:8080/ureport2-demo/ureport/preview?_u=p)。

#### 预览正在设计中的报表注意事项

在预览正在设计器中设计的报表时 (也就是\_u=p), UReport2会在点击设计器工具栏  图标时, 将设计中的报表模版信息提交到服务器, 存放于当前用户的session中 (用户会话中), 所以一旦用户session失效, 如果我们再次直接预览\_u=p类型报表时, 会看到“**com.bstek.ureport.console.exception.ReportDesignException: Report data has expired,can not do preview.**”这样的错误提示, 表示存放于session中的正在设计器中设计的报表模版已经失效, 不能预览。

当然, 如果\_u的值是一个具体的报表名称, 则不存在这种问题, 只有正在设计器中设计的报表模版在预览中采用了这种机制, 这点需要注意。

如果预览的不是正在设计器中设计的模版, 那么只需要给出具体的报表名称即可, 需要注意的是, 这里的报表名称要以其ReportProvider中要求的前缀开始, 比如[http://localhost:8080/ureport2-demo/ureport/preview?\\_u=file:test.ureport.xml](http://localhost:8080/ureport2-demo/ureport/preview?_u=file:test.ureport.xml)等。

报表预览后, 通过预览页面上方工具栏中一排按钮, 可以实现报表的PDF打印、导出Word、Excel、PDF等, 对应的相关URL如下:

URL	说明
<a href="http://host[:port]/[context-path]/ureport/pdf/show?_u=报表名称">http://host[:port]/[context-path]/ureport/pdf/show?_u=报表名称</a>	这个URL将向浏览器写入一个PDF文件流, 在Chrome、Firefox、Edge这些浏览器中将会直接在线显示PDF文件内容, 这样可以实现在线浏览PDF并打印输出。

http://host[:port]/[context-path]/ureport/word?_u=报表名称	导出对应报表的Word文件
http://host[:port]/[context-path]/ureport/excel?_u=报表名称	不分页导出对应报表的Excel文件
http://host[:port]/[context-path]/ureport/excel/paging?_u=报表名称	分页导出对应报表的Excel文件
http://host[:port]/[context-path]/ureport/excel/sheet?_u=报表名称	分页分Sheet导出对应报表的Excel文件
http://host[:port]/[context-path]/ureport/pdf?_u=报表名称&_n=下载文件名	导出对应报表的PDF文件

#### 关于报表参数

上面的这些URL在使用时，如果需要向报表传入参数，可以将这些参数直接放在URL后面使用，如：http://localhost:8080/ureport2-demo/ureport/word?\_u=file:test.ureport.xml&deptId=D11&age=50，将报表file:test.ureport.xml导出成word时，传入两个参数，分别是：deptId=D11以及age=50

上述的这些URL后面除了可以用\_u参数来指定要预览的报表名称外，还可以通过\_i参数来指定当前要预览报表页码，也就是说如果上述这些URL后面没有\_i参数，将会显示整个报表，如果有则显示\_i参数指定页码对应的报表，比如URL后面跟\_i=2，表示预览报表分页后第2页内容。

在http://host[:port]/[context-path]/ureport/preview对应的HTML报表预览页面中，默认页面上部工具栏会显示一排按钮，用于实现打印和导出其它格式报表，如下图所示：



在上面的工具栏中，一共有9个按钮，分别用于实现打印及导出其它格式报表，在实际使用中，如果希望这个工具栏只出现其中某些个工具的话，我们可以参数在URL后面添加\_t参数来实现，\_t参数如果不指定，那么会显示所有的按钮，否则只显示\_t参数指定的按钮。上面这些按钮的显示分别对应参数值：1、2、3、4、5、6、7、8、9，多个参数间用“,”分隔，下面是一些示例：

示例	说明
_t=1,3	显示第一个和第三个按钮
_t=1,2,5	显示第一个，第二个和第五个按钮
_t=1,2,3,4,5	显示第一个，第二个，第三个，第四个和第五个按钮

如果我们不希望显示HTML预览页面的工具栏，那么设置\_t=0即可。

#### 关于在线打印

在UReport2的报表预览页面中，可以看到UReport2提供了三种在线打印方式，分别是直接HTML打印、直接PDF打印以及PDF在线预览打印。这其中直接HTML打印是利用浏览器的HTML打印功能实现，能用于普通的A4纸类型报表页面的打印，同时它不能打印出报表中定义的页眉页脚，如果有定义的话。对于直接PDF打印以及PDF在线预览打印是服务端向浏览器中写入PDF流，利用Chrome、Firefox、Edge这些浏览器可以在线显示PDF功能实现的打印，这种是直接打印的PDF，所以可以应付各种复杂报表纸张类型的打印输出。

#### Firefox在预览PDF时存在的问题

在预览PDF时，Firefox不允许当前frame外通过javascript调用其打印功能（目前来看可能是Firefox的Bug），所以我们需要手工点击其frame内部自带的打印按钮实现打印。

在业务页面中，如果需要预览打印或导出报表，那么就可以采用上述URL实现需要的功能。

## 在代码中使用报表

在UReport2当中，我们可以使用ExportManager实现在业务代码中导出各种类型的报表，ExportManager接口源码如下：

## ExportManager源码

```
package com.bstek.ureport.export;
import java.util.Map;
import com.bstek.ureport.export.html.HtmlReport;
/**
 * @author Jacky.gao
 * @since 2016年12月4日
 */
public interface ExportManager {
    public static final String BEAN_ID="ureport.exportManager";
    /**
     * 导出Html报表
     * @param file 报表模版文件名
     * @param contextPath 当前项目的context path
     * @param parameters 参数
     * @return 返回一个HtmlReport对象，里面有报表产生的HTML及相关CSS
     */
    HtmlReport exportHtml(String file,String contextPath,Map<String, Object> parameters);
    /**
     * 导出指定页码的Html报表
     * @param file 报表模版文件名
     * @param contextPath 当前项目的context path
     * @param parameters 参数
     * @param pageIndex 页码
     * @return 返回一个HtmlReport对象，里面有报表产生的HTML及相关CSS
     */
    HtmlReport exportHtml(String file,String contextPath,Map<String, Object> parameters,int pageIndex);
    /**
     * 导出PDF报表
     * @param config 包含报表模版文件名、参数等信息的配置对象
     */
    void exportPdf(ExportConfigure config);
    /**
     * 不分页导出Excel
     * @param config 包含报表模版文件名、参数等信息的配置对象
     */
    void exportExcel(ExportConfigure config);
    /**
     * 分页导出Excel
     * @param config 包含报表模版文件名、参数等信息的配置对象
     */
    void exportExcelWithPaging(ExportConfigure config);
    /**
     * 分页分Sheet导出Excel
     * @param config 包含报表模版文件名、参数等信息的配置对象
     */
    void exportExcelWithPagingSheet(ExportConfigure config);
    /**
     * 导出Word
     * @param config 包含报表模版文件名、参数等信息的配置对象
     */
    void exportWord(ExportConfigure config);
}
```

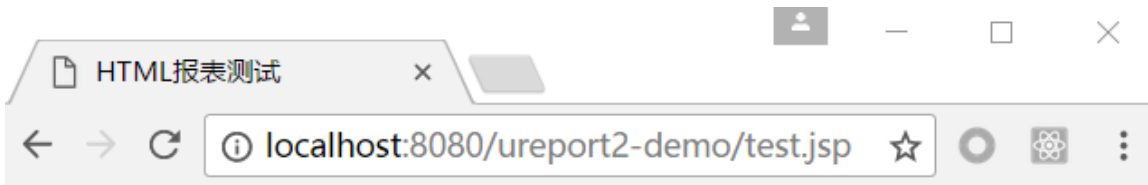
因为ExportManager接口实现是配置在Spring当中，所以要使用ExportManager接口，我们需要首先通过Spring的ApplicationContext取到ExportManager实例对象，我们可以通过ExportManager.BEAN\_ID来取到对应的Bean实例。我们来看下如何采用ExportManager导出HTML报表，并将报表内容嵌入到一个JSP中。

首先我们需要采用报表设计器设计好一个报表模版文件，将其保存，比如报表模版保存后的文件名为demo.ureport.xml，然后在项目中创建一个JSP（这里之所以选择JSP，是因为它最为简单，可以保证所有的J2EE开发者都能看懂，实际使用时可能是MVC框架、Ajax等，但如果了解了在JSP中用法，其它的就简单了），在JSP中输入相应代码，导出Html报表，并将其写入到JSP中，完整的JSP源码如下：

#### 导出HTML报表的JSP文件源码

```
<%@page import="com.bstek.ureport.export.html.HtmlReport"%>
<%@page import="java.util.HashMap"%>
<%@page import="java.util.Map"%>
<%@page import="com.bstek.ureport.export.ExportManager"%>
<%@page import="com.bstek.ureport.Utills"%>
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>HTML报表测试</title>
</head>
<body>
<!-- 通过一个HTML链接来导出目标报表模版的PDF格式报表 -->
<a href="<%=request.getContextPath() %>/ureport/pdf?_u=file:demo.ureport.xml">导出PDF格式报表</a>
<p></p>
<%
ExportManager
exportManager=(ExportManager)Utills.getApplicationContext().getBean(ExportManager.BEAN_ID);
Map<String,Object> parameters=new HashMap<String,Object>();
HtmlReport htmlReport =
exportManager.exportHtml("file:demo.ureport.xml",request.getContextPath(),parameters);
//输出Css样式
out.println("<style type=\"text/css\">");
out.println(htmlReport.getStyle());
out.println("</style>");
//输出报表内容
out.println(htmlReport.getContent());
%>
</body>
</html>
```

启动项目，浏览这个JSP，就可以看到生成的HTML报表内容，如下图所示：



[导出PDF格式报表](#)

D12	硕士	安林	2027.00
		白小波1	2034.00
	大专	金一	6586.00
	本科	张世斌	4964.00
	博士	周淑清	3218.00
D13	大专	程玉	6020.00
	高中	方世泽	5378.00
		胡文鹏	2298.00
	中专	古爱玲	2828.00
	本科	刘占卫	3533.00
		王维斌	6882.00
		郑琪	6991.00
小学	邱仲潘	4408.00	
D24	博士	陈昊	3015.00
	本科	林俊	5873.00
		王永富	6136.00
	高中	刘建国	5488.00

可以看到，在运行后的JSP中，可以点击上方的链接导出PDF格式报表，下方的表格则是这个报表的HTML格式展现，实际使用时还可以在这个页面里加入一些其它与业务相关元素，具体做法这里就不再赘述。