**JASPER REPORT 자료조사**

회사의 교육지원없이 고군분투한 내용이므로

소유권은 누구에게도 없음

참고한 자료마저 검색을 통해 작성했으므로…

그 누구의 소유권도 될수 없다.

정순구

참고자료

<http://trandent.com/article/ireport/detail/680>

더 많은 자료는 위의 사이트에서 JASPER 를 검색어로 입력

목차

[**(1) SUBREPORT,TABLE,LIST - MULTIPLE DATASOURCE** 3](#_Toc515558837)

[**(2) SUBREPORT,TABLE,LIST - MULTIPLE DATASOURCE** 9](#_Toc515558838)

[**(3) SUBREPORT** 12](#_Toc515558839)

[**(4) SUBREPORT** 14](#_Toc515558840)

[**(5) DOWNLOAD 기능 추가** 15](#_Toc515558841)

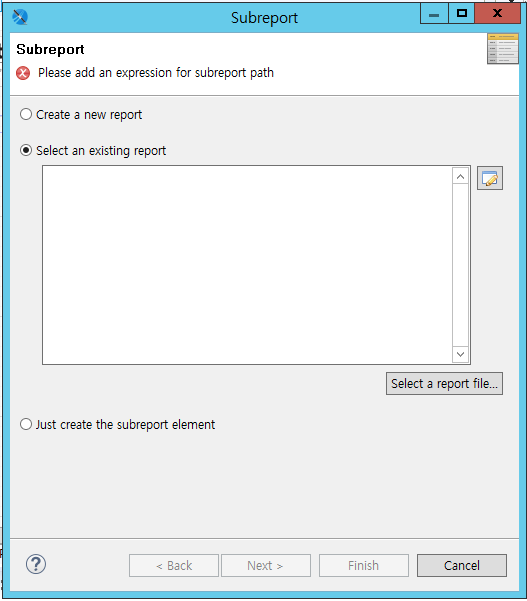
[**word, docx 출력** 18](#_Toc515558842)

[**SUBREPORT (1), IREPORT** 19](#_Toc515558843)

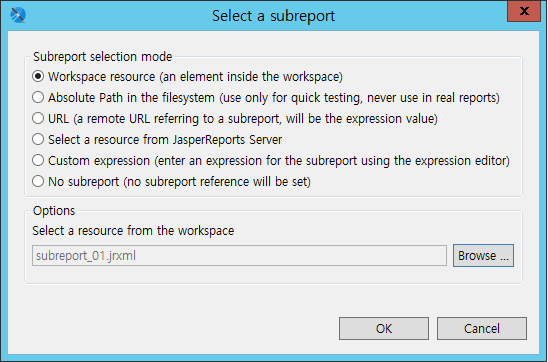
[**IREPORT 작성 (1)** 26](#_Toc515558844)

# **subdataset 추가 -jasperStudio**

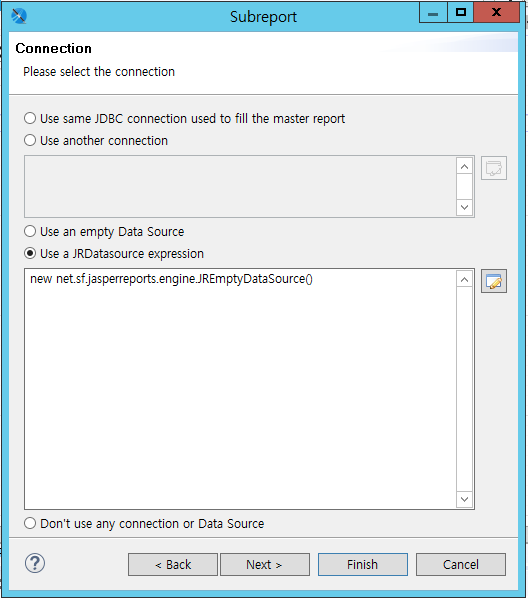
Palette에서 Subreport 클릭 -> 작성중인 보고서 jrxml에 드래그 ->



Select a report file… 을 클릭해서 작성해놓은 subreport용 jrxml파일을 선택



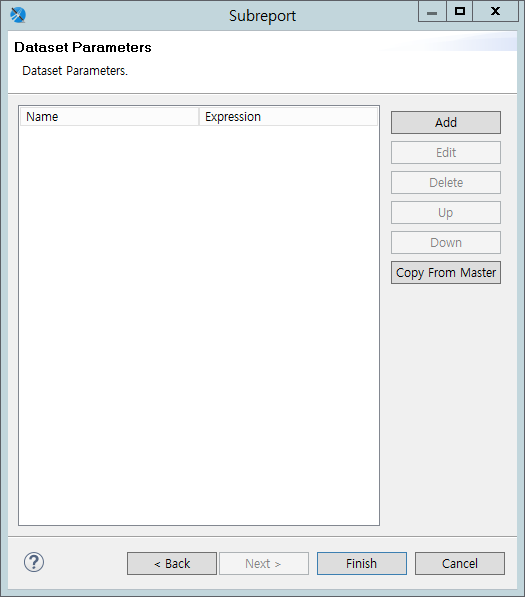
workspace에서 subreport\_01.jrxml파일을 선택한다 -> ok->Next



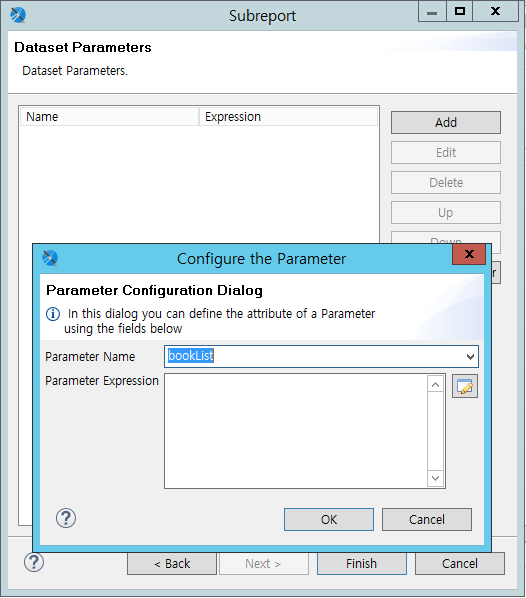
Use a JRDataSource expression을 체크해서 선택, NEXT

위 화면에서 new net.sf.jasperreports.engine.JREmptyDataSource()가 보일 것이다.

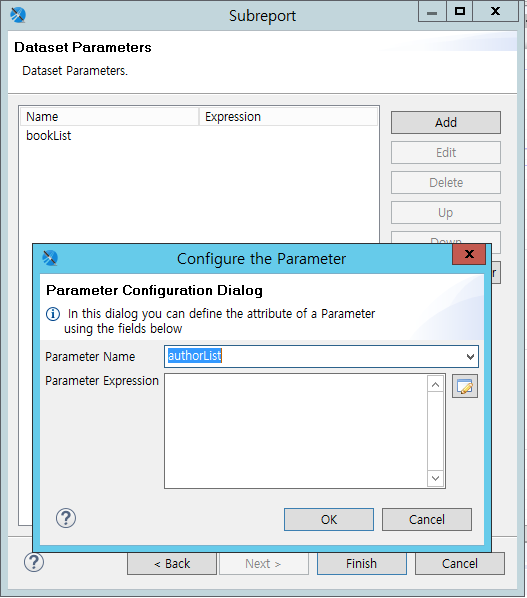
Spring에서 JREmptyDataSource클래스로 dataSource를 연동해서 로직 Value만을 넘겨서 처리할 계획이다.



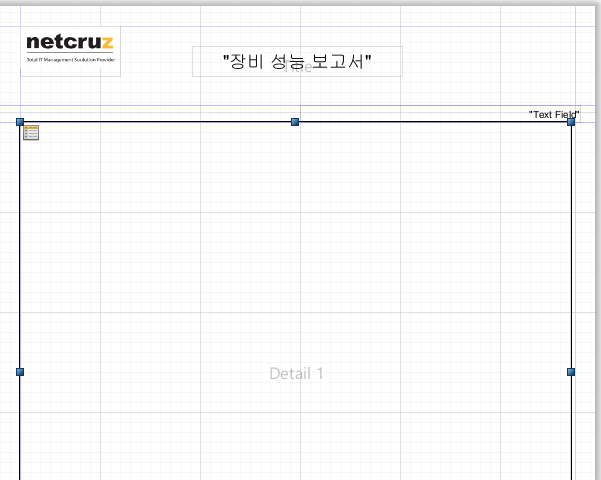
Add 버튼클릭



Ok 버튼 클릭

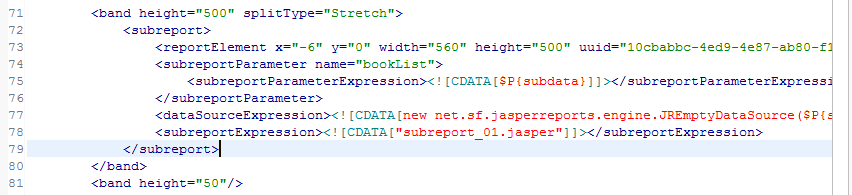


authorList도 같은방식으로 추가후 FINISH



그림과 같이 subreport의 사이즈를 적당히 수정해준다.

완성된 화면에서 [Source]탭을 클릭해서 소스화면을 확인한다.



|  |
| --- |
| <subreport>  <reportElement x="-6" y="0" width="560" height="500" uuid="10cbabbc-4ed9-4e87-ab80-f1fbb4dcefbb"/>  <subreportParameter name="bookList">  <subreportParameterExpression><![CDATA[$P{bookList}]]></subreportParameterExpression>  </subreportParameter>  <dataSourceExpression><![CDATA[new net.sf.jasperreports.engine.JREmptyDataSource($P{bookList}.getData())]]></dataSourceExpression>  <subreportExpression><![CDATA["subreport\_01.jasper"]]></subreportExpression>  </subreport> |

<subReportParameterExpression></subParameterExpression>

<dataSourceExpression></dataSourceExpression>

태그 내부에 위와 같이 입력해준다.

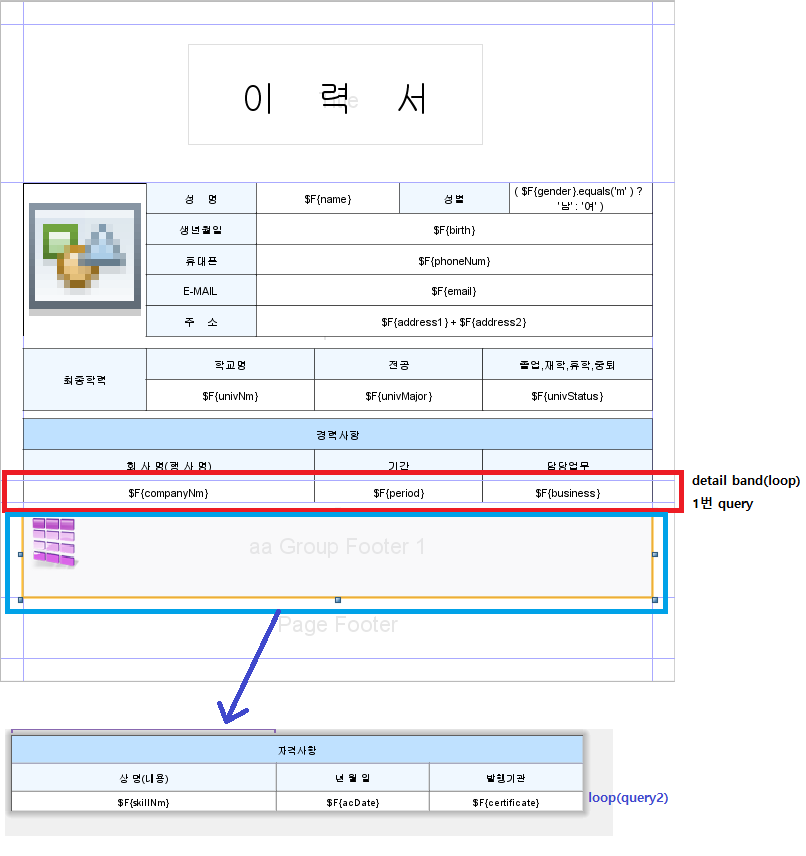
# **(1) SUBREPORT,TABLE,LIST - MULTIPLE DATASOURCE**

ireport로 디자인하는경우 쿼리 두개를 돌리려면 dataset이 두개가 있어야 한다.

반복되는 데이터의 출력도 detail band에서만 되기 때문에 loop를 돌리려면 subreport를 쓰던지 table을 쓰던지 list를 써서

별도의 sub datasource를 통해 돌려야 한다.

아래의 예제는 이력서 양식이다.



detail band에서는 경력사항을 반복 출력하고

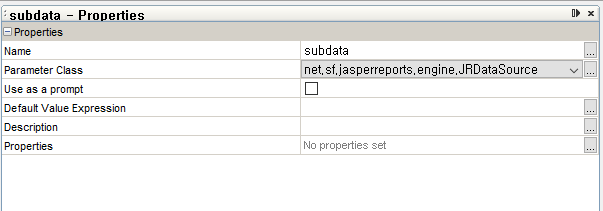
하단의 자격사항은 쿼리가 다르기 때문에 테이블을 이용하여 sub datasource 에 담아 반복 출력한다.

설정

1. report에 parameter를 추가한다.

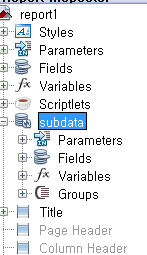


parameter class는 net.sf.jasperreports.engine.JRDataSource로 설정한다.

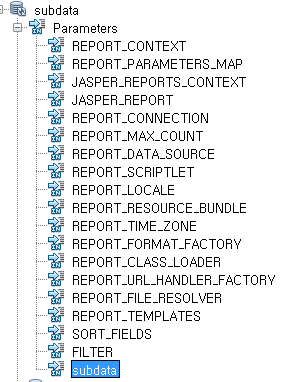


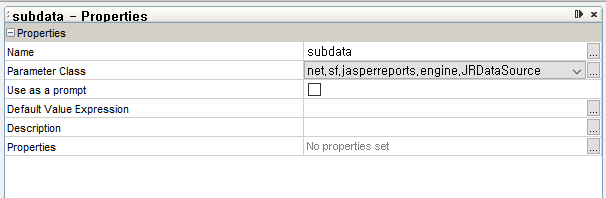
2. 리포트  우클릭 - add dataset

subdata라는 이름으로 dataset을 추가했다.



2. sub dataset에도 parameter를 추가한다.





3.xml 탭으로 가서 source를 보면

<subDataset name="subdata" uuid="3a2bb95c-3f1f-44b5-8fa7-f8b04c127ce7">

<parameter name="subdata" class="net.sf.jasperreports.engine.JRDataSource" isForPrompting="false"/> -> sub dataset parameter

<queryString>

<![CDATA[SELECT aa,bb,cc FROM table]]>

</queryString>

필드

필드

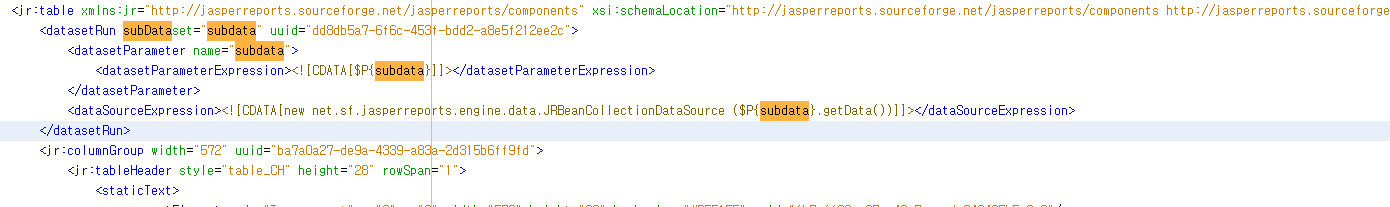
필드

필드

</subDataset>

<parameter name="subdata" class="net.sf.jasperreports.engine.JRDataSource" isForPrompting="false"/> -> report parameter

이렇게 되어있어야한다.

그후 하단에 dataRun을 찾는다.

datasetParameterExpression = $P{subdata},

dataSourceExpression = new net.sf.jasperreports.engine.data.JRBeanColletionDataSource ($P{subdata}.getData())

로 작성한다.

이렇게 되어 있으면 ireport에서 할일은 끝났다

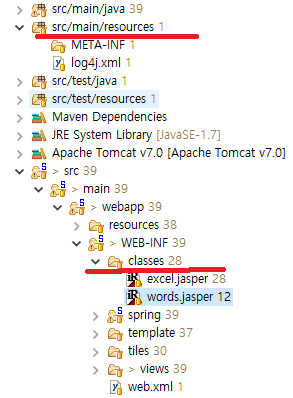
# **(2) SUBREPORT,TABLE,LIST - MULTIPLE DATASOURCE**

1번이 끝났으면 2번은 스프링에 적용하는 방법이다.

1번에서 만든  jrxml을 컴파일 하여 .jasper 파일을 만든다.

word.jrxml 이었으면 word.jasper 파일이 나올것이다.

1. .jasper파일을 WEB-INF/classes 폴더 또는 src/main/resources 폴더 안에 넣는다.



2. controller 작성

@RequestMapping(value = "/reports/download",method = RequestMethod.POST)

public ModelAndView download(HttpServletRequest request)

throws Exception {

ArrayList<ResumeVO> list = new ArrayList<ResumeVO>();

list = resumeDAOService.selectMemberInfo(memSeq);

**//리스트에 쿼리를 돌려 각자 사용할 데이터를 담는다. list = 리포트 쿼리**

     ArrayList<ResumeVO> list2 = new ArrayList<ResumeVO>();

list2 = resumeDAOService.selectSkills(list.get(0).getResumeSeq());

**// list2 = sub dataset 에서 설정한 쿼리.**

**//list에 담을때 주의할 점은 ireport에 설정한 필드명이과 쿼리의 컬럼이 일치해야됨**

URL location = getClass().getResource("/words.jasper");  // .jasper 파일 경로 불러오기

String reportPath = location.getPath();       // .jasper 파일 경로 저장

JRBeanCollectionDataSource src = new JRBeanCollectionDataSource(list,false);      // 리포트 메인 쿼리의 값을 담을 데이터 소스

JRBeanCollectionDataSource src2 = new JRBeanCollectionDataSource(list2,false);    // sub dataset 쿼리 값을 담을 데이터 소스

Map<String,Object> parameters = new HashMap<String,Object>();

parameters.put("datasource", src);

parameters.put("**subdata**", src2);          **// 1장에서 설정한 dataset 이름과 같아야됨**

JasperPrint jasperPrint = JasperFillManager.fillReport(reportPath, parameters,src);

HttpSession session = request.getSession();

String path = session.getServletContext().getRealPath("")+"/resources/download/resume/";    // 이력서의 저장 경로

JRDocxExporter exporter = new JRDocxExporter();  //워드로 저장하기 위해 JRDocxExporter 사용

exporter.setParameter(JRExporterParameter.JASPER\_PRINT, jasperPrint);

exporter.setParameter(JRExporterParameter.OUTPUT\_FILE\_NAME, "이력서.docx");

exporter.exportReport();  // 워드로 내보낸다

return DownloadController("이력서.docx");     // 저장 후 바로 다운로드를 위해서 downloadController로 넘김 이 컨트롤러는 예전에 한 정리에 있음

}

public ModelAndView DownloadController(String path) throws UnsupportedEncodingException {

File file = new File(path);

return new ModelAndView("download", "downloadFile", file);

}

Download 세팅은

<http://trandent.com/board/iReport/detail/240>

여기서 1,3번을 보면 됩니다.

# **(3) SUBREPORT**

**SubReport의 사용**

앞서 iReport에서 디자인한 jrxml 파일을 예로 들면 SubReport의 datasource의 이름을 subdata로 설정 하였다.

Multiformat-view를 이용하여 PDF 파일로 출력 하는 경우 Controller와 jasperreport-views.properties를 수정해야 한다.

부모 report file과 자식 report file은 WEB\_INF\reports\ 폴더 내에 위치 시킨다.

**ReportController.java**

@Controller

**public class ReportController {**

@RequestMapping(value = "/reports/test.do")

**public ModelAndView test(**

@ModelAttribute(“testVO") TestVO testVO,

Map<String, Object> map, ModelMap model) **throws Exception{**

**본인 프로젝트에 맞는 방법으로**

**쿼리를 실행하여 값을 리스트에 담는다**

**-Subreport용 datasource는 list2에 담는다.**

          List<Map<String,Object>> list = new ArrayList<Map<String,Object>>();

**List<Map<String,Object>> list2 = new ArrayList<Map<String,Object>>();**

**return JasperReportUtil.*render("multiformat-view", list, list2, "pdf");              //     render에 list, list2를 리턴한다.***

**}**

**public static class JasperReportUtil extends Exception {**

**private static final long *serialVersionUID = 1L;***

**public static ModelAndView render(String templateId,**

**List<Map<String, Object>> list, List<Map<String, Object>> list2, String fmt) {**

  JRBeanCollectionDataSource src = **new**JRBeanCollectionDataSource(list);

  JRBeanCollectionDataSource src2 = **new JRBeanCollectionDataSource(list2);**

  ModelAndView mv = **new ModelAndView();**

  mv.setViewName(templateId);

  mv.addObject("format", fmt);

  mv.addObject("datasource", src);

**mv.addObject("subdata",src2);              //**iReport에서 설정한 이름과 같아야 한다.

**return mv;**

  }

}

**​Jasperreports-views.properties**

multiformat-view.(class)=org.springframework.web.servlet.view.jasperreports.JasperReportsMultiFormatView

multiformat-view.url=/WEB-INF/reports/파일명.jrxml            // 부모 jrxml 파일명

multiformat-view.reportDataKey=datasource

**multiformat-view.subReportDataKeys=subdata**

# **(4) SUBREPORT**

**WORD, EXCEL형식 SubReport사용**

**ReportController.java**

@Controller

**public class ReportController {**

  @RequestMapping(value = "/reports/test.do")

**public ModelAndView test(**

  @ModelAttribute(“testVO") TestVO testVO,

  Map<String, Object> map, ModelMap model) **throws Exception{**

**---본인 프로젝트에 맞는 방법으로 쿼리를 실행하여 값을 리스트에 담는다---**

          List<Map<String,Object>> list = new ArrayList<Map<String,Object>>();

**List<Map<String,Object>> list2 = new ArrayList<Map<String,Object>>();       SubReport용 Data List**

**~생략~**

          JRBeanCollectionDataSource src = **new**JRBeanCollectionDataSource(list);

**JRBeanCollectionDataSource src2 = new JRBeanCollectionDataSource(list2);** **SubReport용 Datasource**

          Map parameters = **new HashMap();**

          parameters.put("dataSource", src);

          parameters.put("subdata", subSrc);          iReport에서 설정한 이름을 키값으로 Datasource를 넣는다.

          JasperPrint jasperPrint = JasperFillManager.*fillReport(reportPath,***parameters***,src);*

**~생략~**

# **(5) DOWNLOAD 기능 추가**

REPORT DOWNLOAD 기능 추가. Report 출력 시, WEB\_INF/reports/ 폴더에 생성된 파일 다운로드

1.\*\*\*-servlet.xml 설정

<bean id=*"download" class="egovframework.com.aam.report.DownloadView" />  // DownloadView.java를 생성할 위치*

<bean class=*"org.springframework.web.servlet.view.BeanNameViewResolver">*

  <property name=*"order" value="0"/>*

</bean>

<bean id=*"viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">*

  <property name=*"prefix" value="/" />*

  <property name=*"suffix" value=".jsp" />*

  <property name=*"order">*

  <value>1</value>

  </property>

</bean>

**​**2. ReportController.java

@RequestMapping("/reports/Download.do")

**public ModelAndView DownloadController(String type) throws UnsupportedEncodingException {**

  System.*out.println(type);*

  String fileName = **null;**

  URL location = getClass().getResource("/word.jasper");

  String reportPath = location.getPath();

  String path = reportPath.substring(0, reportPath.length() - 11);

**if(type.equals("excel")){**

  fileName=*prjName +excelName;*

  }**else if(type.equals("word")){**

  fileName=*prjName + wordName;*

  }

  path = **new String(path.getBytes("8859\_1"), "UTF-8");**

  fileName = **new String(fileName.getBytes("8859\_1"), "UTF-8");**

  String fullPath = path + fileName;

  File file = **new File(fullPath);**

**return new ModelAndView("download", "downloadFile", file);**

}

​3. DownloadView.java

**public class DownloadView extends AbstractView{**

•

**public DownloadView(){**

**super.setContentType("application/download; charset=utf-8");**

    }

•

    @Override

**protected void renderMergedOutputModel(Map<String, Object> model, HttpServletRequest request, HttpServletResponse response) throws Exception {**

•

         File file = (File)model.get("downloadFile"); //downloadFile의 이름으로 삽입된 파일!

         System.*out.println("DownloadView --> file.getPath() : " + file.getPath());*

         System.*out.println("DownloadView --> file.getName() : " + file.getName());*

         response.setContentType(getContentType());

         response.setContentLength((**int)file.length());**

         String userAgent = request.getHeader("User-Agent");

**boolean ie = userAgent.indexOf("MSIE") > -1;**

         String fileName = **null;**

**if(ie){**

                 fileName = URLEncoder.*encode(file.getName(), "utf-8").replace("+","%20");*

             } **else {**

                 fileName = **new String(file.getName().getBytes("utf-8"), "iso-8859-1").replace("+","%20");**

             }// end if;

​response.setHeader("Content-Disposition", "attachment; filename=\"" + fileName + "\";");

         response.setHeader("Content-Transfer-Encoding", "binary");

         OutputStream out = response.getOutputStream();

         FileInputStream fis = **null;**

**try {**

             fis = **new FileInputStream(file);**

             FileCopyUtils.*copy(fis, out);*

         } **catch(Exception e){**

             e.printStackTrace();

         }**finally{**

**if(fis != null){**

**try{**

                     fis.close();

                 }**catch(Exception e){**

                     e.printStackTrace();

                 }

             }

         }// try end;

         out.flush();

    }

}

# **word, docx 출력**

WORD DOCX 문서 출력

Multiformat-view에서는 docx 형식의 출력을 지원하지 않는다. 따라서 JasperPrint와 JRDocxExporter를 사용하여 출력한다.

Excel 형식으로 출력하고 하는 경우 JRDocxExporter 대신 JRXlsExporter를 사용한다.

1.WEB-INF\reports\ 폴더 안에 iReport로 디자인한 jasper 파일을 넣는다.

2.ReportController.java

 @Controller

**public class ReportController {**

  @RequestMapping(value = "/reports/test.do")

**public ModelAndView test(**

  @ModelAttribute(“testVO") TestVO testVO,

  Map<String, Object> map, ModelMap model) **throws Exception{**

**//---본인 프로젝트에 맞는 방법으로 쿼리를 실행하여 값을 리스트에 담는다---**

          List<Map<String,Object>> list = new ArrayList<Map<String,Object>>();

          URL location = getClass().getResource("/파일명.jasper");

          String reportPath = location.getPath();                       //      jasper파일의 경로를 불러와 reportPath에 저장한다.

          JRBeanCollectionDataSource src = **new**JRBeanCollectionDataSource(list);**//**list로 datasource를 만든다.

          JasperPrint jasperPrint = JasperFillManager.*fillReport(reportPath,***new**HashMap() *,src);*

          JRDocxExporter exporter = **new**JRDocxExporter();

         exporter.setParameter(JRExporterParameter.*JASPER\_PRINT, jasperPrint);*

         exporter.setParameter(JRExporterParameter.*OUTPUT\_FILE\_NAME, “저장할 경로”+“파일명.docx”);*

exporter.exportReport();                                                       //        지정한 경로에 docx 파일이 생성된다.

# **SUBREPORT (1), IREPORT**

•SubReport

•하나의 Report에서는 한 개의 쿼리만 사용 가능

하고 Detail band 외의 band에서는 자료의 반복 출력이

안되기 때문에 두개의 report 페이지를 연결하여 두 개

이상의 쿼리를 사용하여 값을 가져와 출력하기 위해

사용한다.

                        (\* list나 table을 이용하는 방법도 있음.)



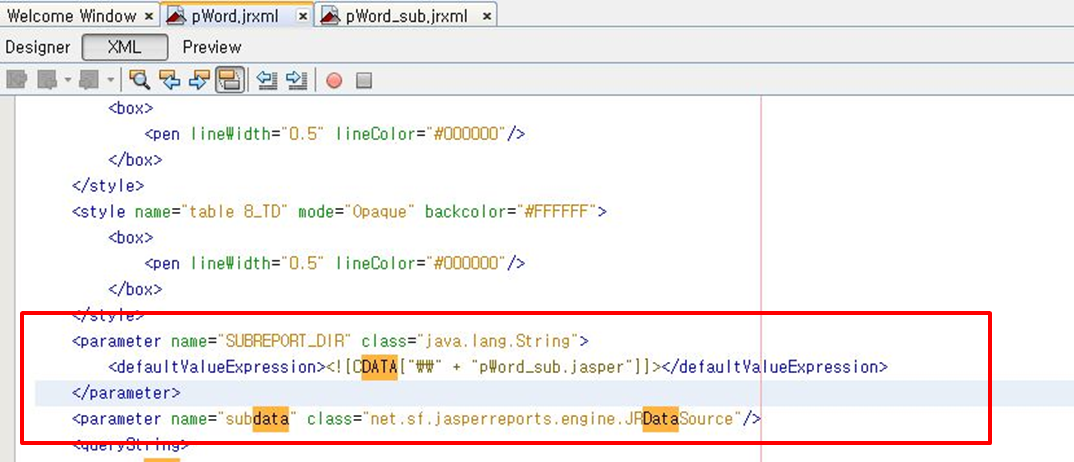
•SubReport를 Spring 프로젝트에서 사용하기 위해서는 parameter 세팅을 해주어야 한다.

•부모의 jrxml파일의 XML탭을 열어 parameter를 설정한다.

•<defaultValueExpression> -> 자식 jasper 파일의 경로 </defaultValueExpression>

•-jasper 파일은 jrxml파일을 compile하게 되면 생성됨

•<parameter name=“subdata” -> spring에서 사용할 datasource명                                                                                                                                                                                  Class=“net.sf.jasperreports.engine.JRDataSource”/>



•<subreport></subreport> 부분을 찾아 그림과 같이 수정한다.

•<dataSourceExpression> -> 앞에서 설정한 이름($P{이름} = parameter를 의미)

•<subreportExpression> -> 자식 report의 경로 (현재 부모파일과 같은 위치로 설정)



• Preview상에서는 자식 리포트 파일의 값은 출력되지 않을 수 있다.

​

[***#ireport***](http://trandent.com/article/ireport/detail/235)

HKH2  
2014-07-12 13:07

SHARE

* [](javascript:fbShare())

* [](javascript:sendLink())

* [](javascript:shareNaver())

* [](javascript:shareTwitter())

로그인

로그인 해주세요.

그러면 JSP에서 호출은 어떻게 하나요? 그냥 주소뒤에 .do로 호출하면 끝인가요?

제가 JSP에서 버튼 클릭 후 .do 로 요청했는데 pdf 파일이 백지로 표출되네요ㅠㅠ

답글

mk2016/10/07 14:42:35

댓글 감사합니다. 위에 소스랑 설정 그대로 작업 했는데 실행이 안됩니다.

@RequestMapping(value = "/help/openHelp\_report.do")

public ModelAndView test(

Map map, ModelMap model) throws Exception{

//본인 프로젝트에 맞는 방법으로

//쿼리를 실행하여 값을 리스트에 담는다

List<map> list = new ArrayList<map>();

return JasperReportUtil.render("multiformat-view", list, "pdf");

}

이렇게 호출하면 에러는 없고 실행이 안됩니다. 이렇게 호출했을경우 정상적으로 구동 되는경우 미리보고 화면이 나와야 되는겁니까?</map</map

답글

김현기2016/05/15 00:50:39

testVO는 안쓰셔도 됩니다

그냥 리스트로 받아서 넘겨주면 됩니다

답글

HKH22016/05/15 00:50:39

# **IREPORT 작성 (1)**

•Title band

•좌측 Report Inspector에서 파일명 우클릭 Properties – Title on a new page 클릭

 -> 리포트 표지작성 가능

•Page header / page footer band

•모든 페이지마다 반복 출력가능

•Detail band

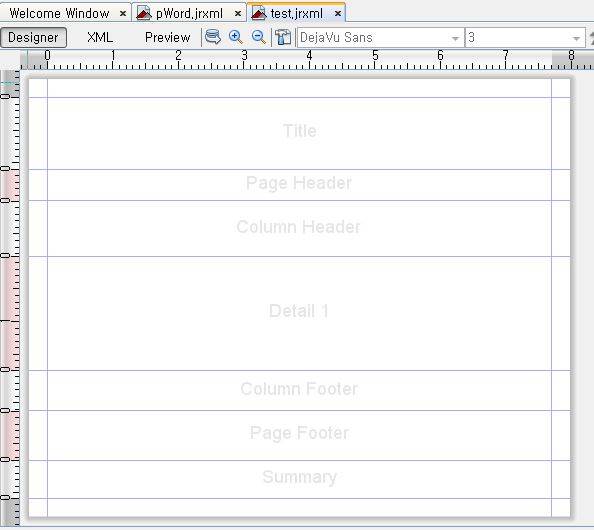
•값이 여러 개인 경우 한 페이지 내에서 반복 출력가능

•Column header / Column Footer band

•모든 페이지에서 출력 가능하며 크기를 자유자제로 조절가능

•Summary

•요약페이지. 맨 마지막에 새 페이지로 따로   
출력가능



•Group

Report Inspector에서 report를 우클릭

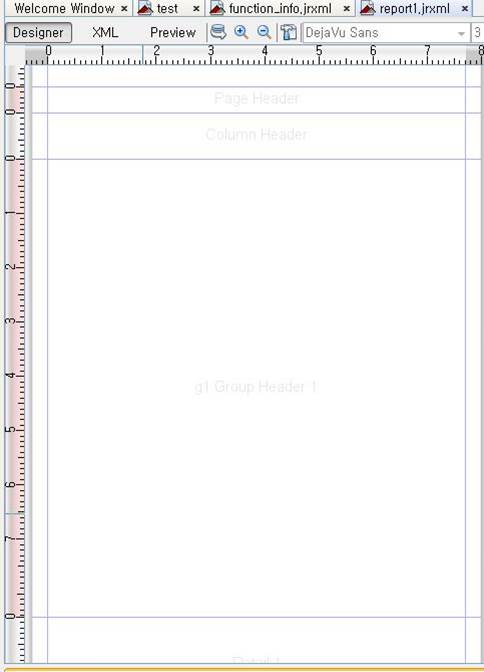
Add Group

Group Header 우클릭

Maximize Band Height

리포트 내 다른 디자인의 새로운 페이지

      추가 가능



•Field

•Report Inspector에서 리포트명 우클릭-> Edit Query

쿼리 입력 후 Read Field를 클릭하면 자동으로 field가

추가되며, $F{필드명}을 통해 Report에 값을 출력 할 수

있다.

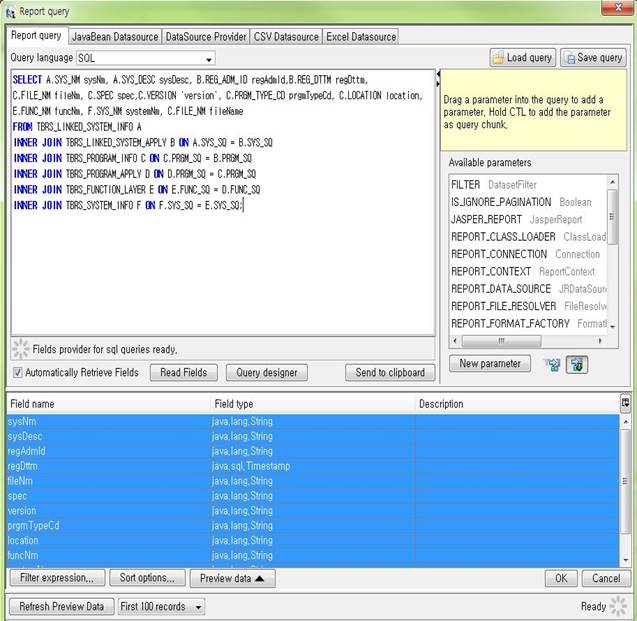
Report Inspector의 Fields 탭에서 drag & drop으로도

사용 가능하다.

Spring project에서 쿼리 실행시 list에 들어 오는 값과

Jrxml파일에서 설정한 field name이 mapping되어

값이 화면에 출력된다.



아래 그림과 같이 특정 필드를 기준으로 각 페이지 마다 하위 필드의 리스트를

표현 하고 싶을 경우

Group생성-> Group band property –> Group Expression-> $F{기준 필드명}을 입력, Start on a new page check

필드의 내용이 바뀔 때 마다 새 페이지에 하위 필드 반복 출력.Excel로 출력 시 내용을 새로운 Sheet마다 출력하고 싶은 경우 Report Inspector-> report명 우클릭-> properties -> Properties -> Add

-> Property name = net.sf.jasperreports.export.xls.one.page.per.sheet   Property value = TRUE 입력

