

iReport & jasperreport 기술정의서

최진원

iReport란??

- 웹보고서 오픈 소스 프레임워크인 JasperReports 를 사용하기위한 jrxml 를 만들기위한 툴이다.

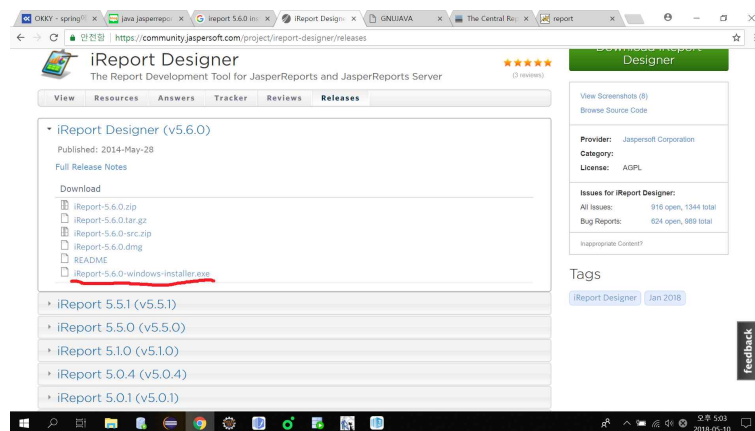
기존에는 JasperReports 홈페이지의 jrxml sample을 변형하여 JasperReports를 사용하였으나, iReport를 이용하여 간단하게 jrxml를 만들고, iReport에서 지원하지 않은 기능만 추가 수정하여 진행한다고 한다.

※내가 참고한 사이트는 맨~밑에 링크※

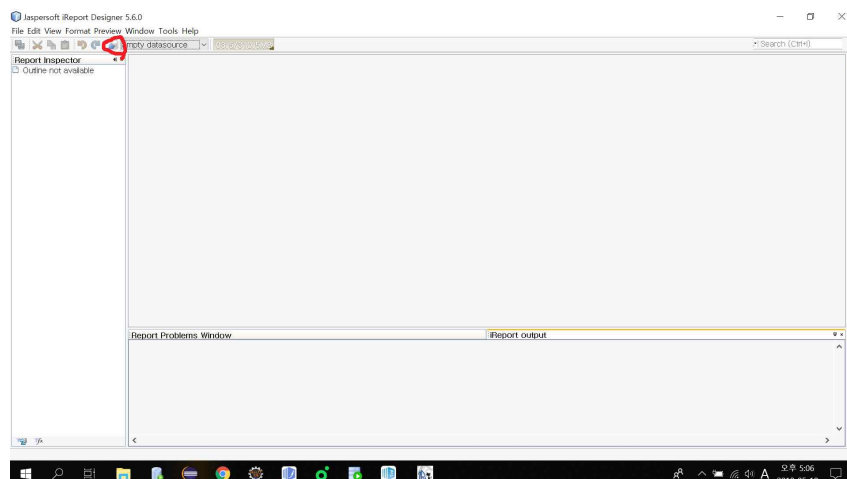
iReport는 jrxml를 쉽게 만들기위한 디자인 툴이라고 생각하면 된다. 난 5.6.0버전을 사용함.

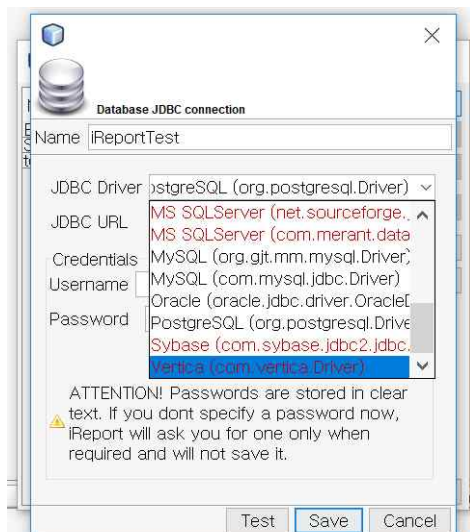
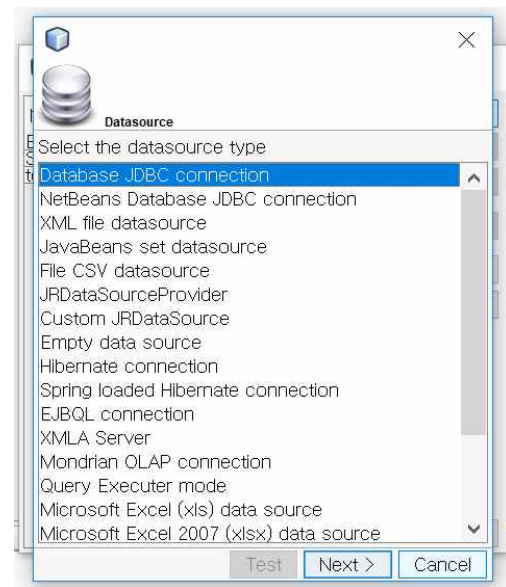
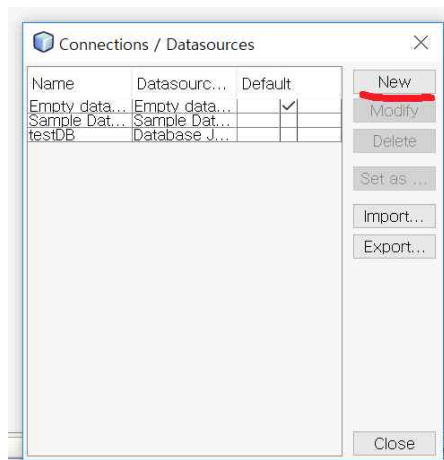
<https://community.jaspersoft.com/project/ireport-designer/releases>

인스톨러 다운

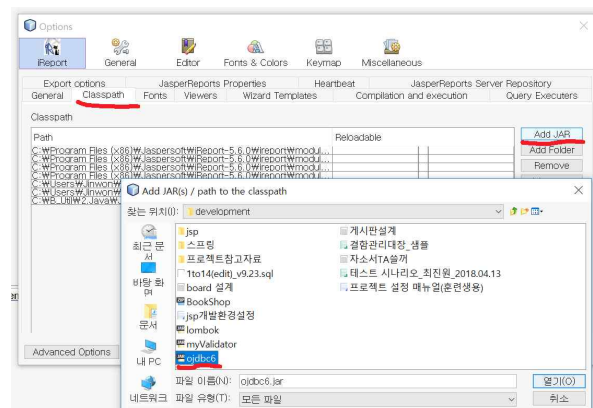
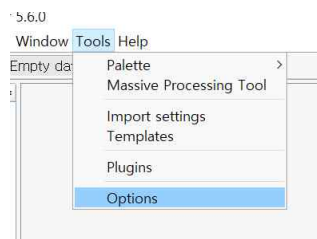


iReport 툴을 실행하고 드라이버를 연결해줘야한다.

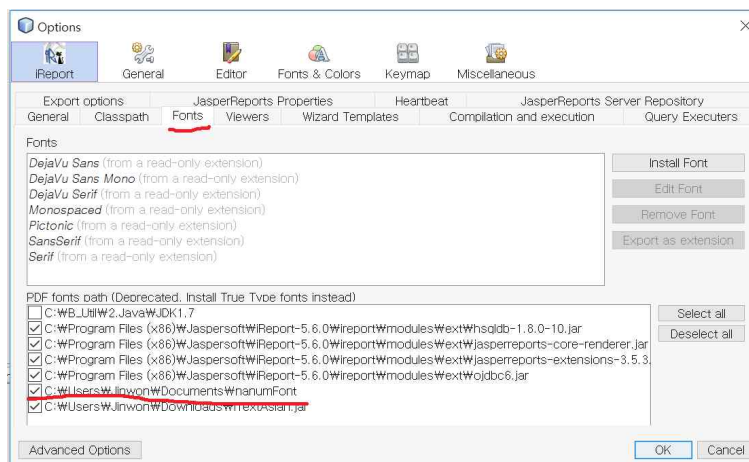
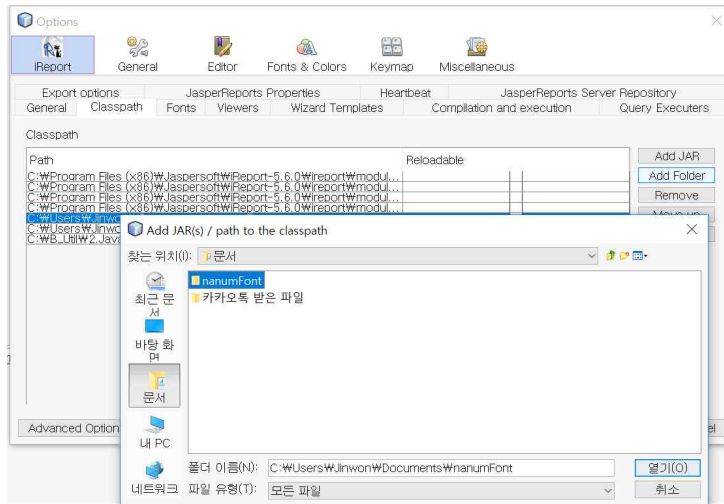




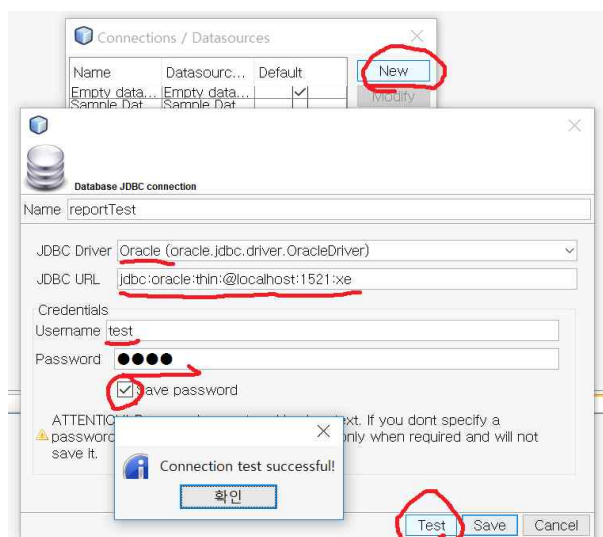
나는 오라클을 쓰는데 아마 오라클이 빨간색으로 되었을거다. jdbc드라이버를 패스에 등록해 줘야한다. 일단 디비연결창은 끄고 옵션으로 가자



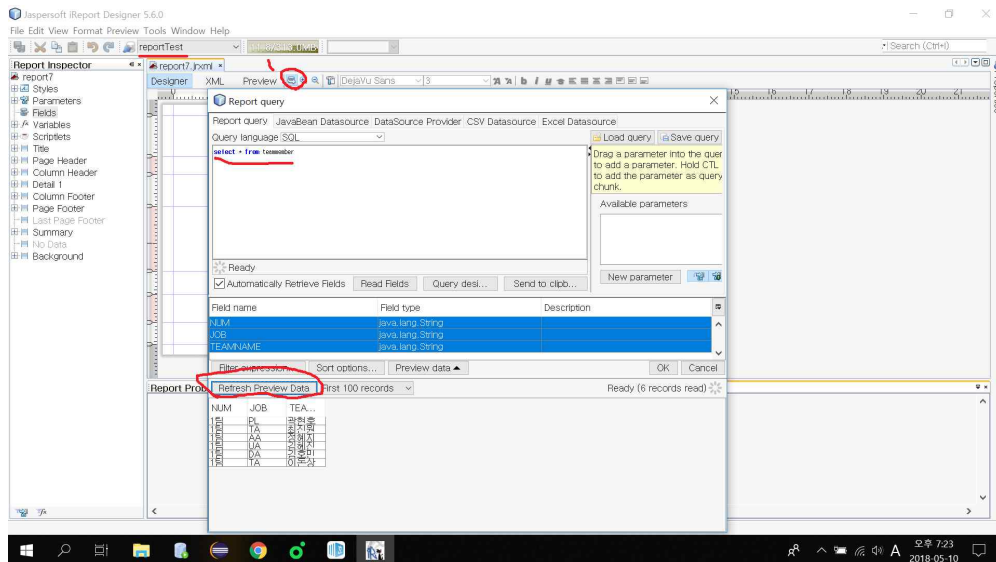
jdbc를 추가해주고 나중 한글폰트를 위해 네이버나눔폰트를 받은다음 addFolder로 추가한다.



다시 디비연결로 가서 디비를 연결해준다.



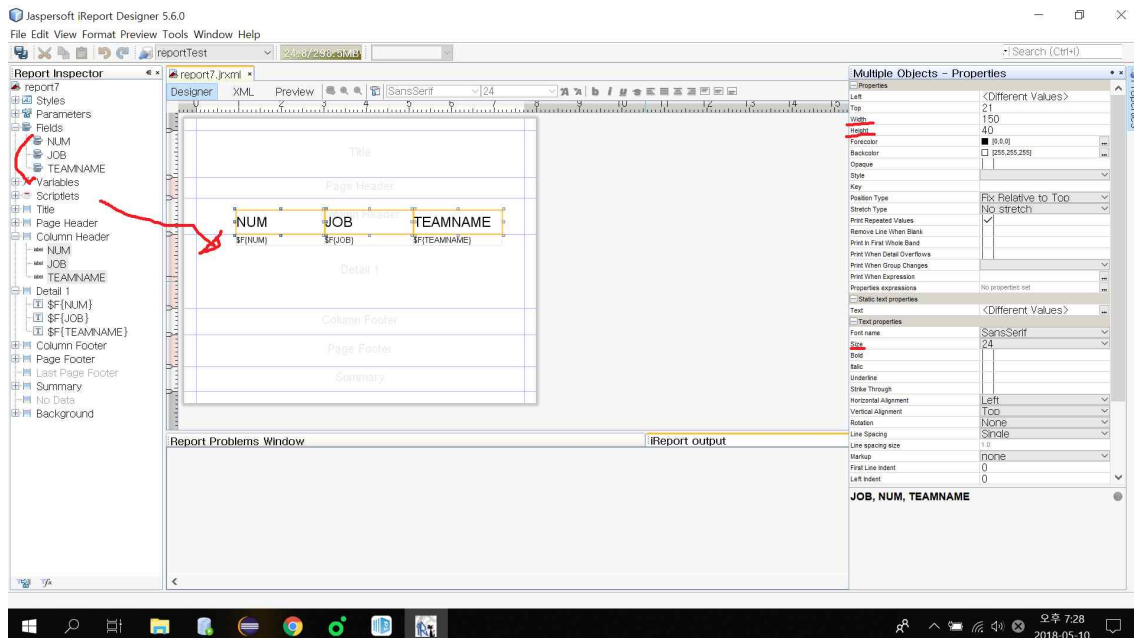
디비연결을 완료했으면 File -> new로 새로운 템플릿을 만들어준 다음 사용할 쿼리를 입력해 주고 데이터를 받아온다.



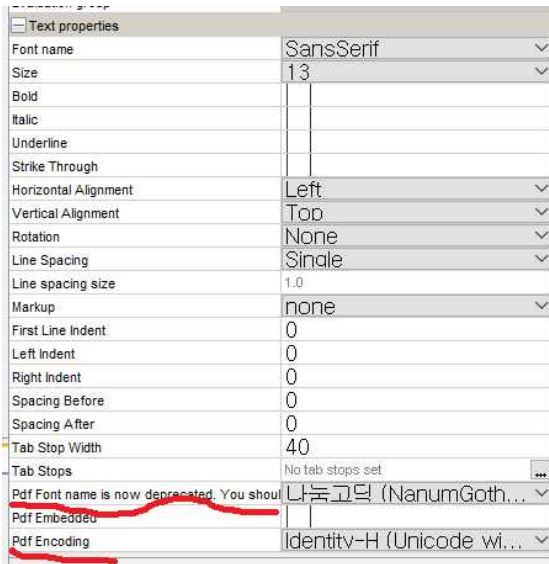
디비의 데이터를 가지고오면 왼쪽의 Fields메뉴에 컬럼이 나온다.

그걸 드래그앤드랍으로 끌어서 Detail간에 놓으면 컬럼명과 파라미터가 놓여지고 오른쪽의 프로퍼티의 옵션들로 넓이,높이,글자크기,폰트 등등을 조절할 수 있다.

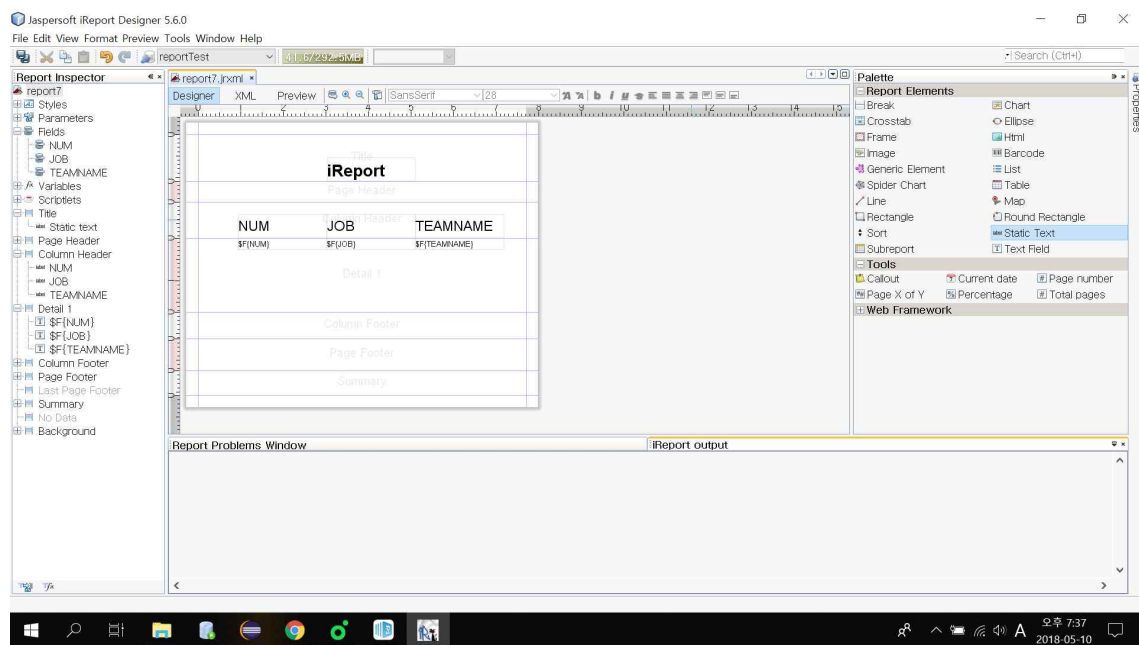
※(중요) 나중에 스프링과 연결할 때 Fields의 컬럼명을 키값으로 데이터를 갖고 오기 때문에 VO의 변수명과 대소문자구별까지 같아야한다.



한글 데이터의 깨짐 방지를 위해 오른쪽에 텍스트프로퍼티의 폰트를 패스에 넣어준 한글폰트로 지정해주고 인코딩을 Identity-H로 줘야한다.



나머지 보고서의 디자인들은 상단메뉴의 윈도우->팔레트에서 추가할 수 있다.(한글로 출력할 경우 한글 폰트 설정 필수)

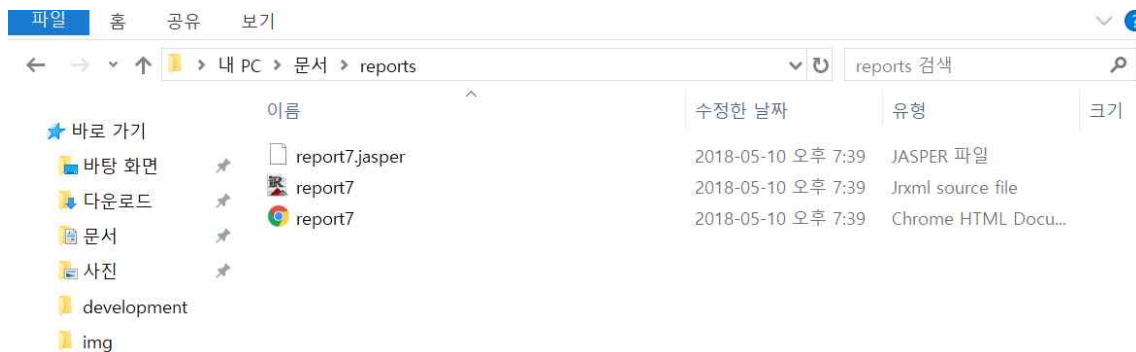


프리뷰를 눌러보면 디자인한 파일을 미리보기할수 있으며, 상단메뉴의 프리뷰 ->pdf로 설정할 경우 pdf로 미리보기할 수 있다.

팁 : Designer탭의 디테일칸을 줄이면 파라미터간의 상하 여백을 줄일 수 있다.



보고서의 디자인 작업이 끝났으면 새로운 파일을 만들 때 설정했던 저장경로에 가보면 .jasper파일과 .jrxml파일이 생성되었는데 .jrxml파일이 우리가 디자인을 했던 파일이고 .jasper파일은 컴파일된 파일이다. 스프링과 연동시 jrxml파일로 연동을하면 스프링이 컴파일 을해서 실행하고 .jasper파일을 연동시 컴파일 필요없다 (결론으로 뭘쓰던 상관없)



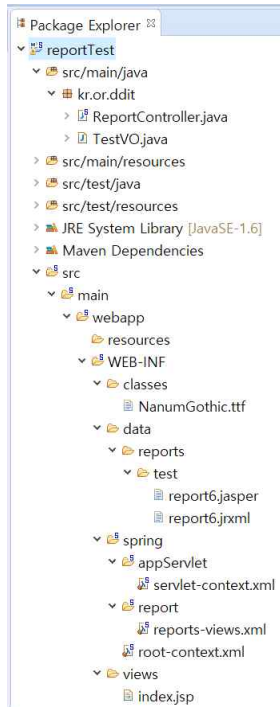
-----여기까지가 iReprot 디자인 -----

이제 스프링과 연동해보겠다. 스프링 엠비씨 프로젝트로 테스트프로젝트를 생성해주었다. 먼저 라이브러리를 추가해줬는데 iReport버전과 jasperreports의 버전을 맞춰준다. groovy는 나도 왜넣은지 모르겠고 itextpdf는 pdf파일을 위해 필요한 라이브러리다(필수)

```
<dependency>
<groupId>net.sf.jasperreports</groupId>
<artifactId>jasperreports</artifactId>
<version>5.6.0</version>
</dependency>
<dependency>
<groupId>org.codehaus.groovy</groupId>
<artifactId>groovy-all</artifactId>
<version>2.3.4</version>
</dependency>
<dependency>
<groupId>com.itextpdf</groupId>
<artifactId>itextpdf</artifactId>
<version>5.4.0</version>
</dependency>
```

일단 내 디렉토리 구조다. 테스트에 필요한것들만 생성했다.

index는 어쩌다 보니 view폴더에 들어가있는데 jrxml파일을 스프링과 연동하는게 목적이기 때문에 의미가 없다.



먼저 xml부터 수정해보겠다. servlet-context.xml 에 다음 코드를 추가한다.

```
<beans:bean class="org.springframework.web.servlet.view.XmlViewResolver">
  <beans:property name="location" value="/WEB-INF/spring/report/reports-views.xml"/>
  <beans:property name="order" value="1"/>
</beans:bean>
```

다음으로 /WEB-INF/spring폴더안에 report폴더를 만들고 reports-views.xml파일을 만든다. 폴더구조는 서블릿의 로케이션의 값과 맞춰준다.

reports-views.xml 에 다음 코드를 추가한다.

```
<bean id="pdfReport" class="org.springframework.web.servlet.view.jasperreports.JasperReportsPdfView"
  p:url="/WEB-INF/data/reports/test/report6.jrxml"
  p:reportDataKey="datasource" |
/>
```

xml 설정은 끝났다.

다음으로 아~까 디자인할 때 썼던 한글폰트를 classes 폴더안에 넣어준다.(디렉토리구조참고) 그다음 만들었던 .jasper파일과 .jrxml파일을 reports-views.xml의 url 경로와 맞춰서 넣어준다. jrxml로만 연동할꺼기 때문에 jrxml파일만 넣어줘도 무관.

VO하나를 만들어준다.

여기서 중요한건 아까 디자인할 때 강조했던 컬럼명과 변수명이 일치해야된다(대소문자필수)

```
1 package kr.or.ddit;
2
3 import lombok.Data;
4
5 @Data
6 public class TestVO {
7     private String NUM;
8     private String JOB;
9     private String TEAMNAME;
10 }
11 }
```

마지막 자바단이다.

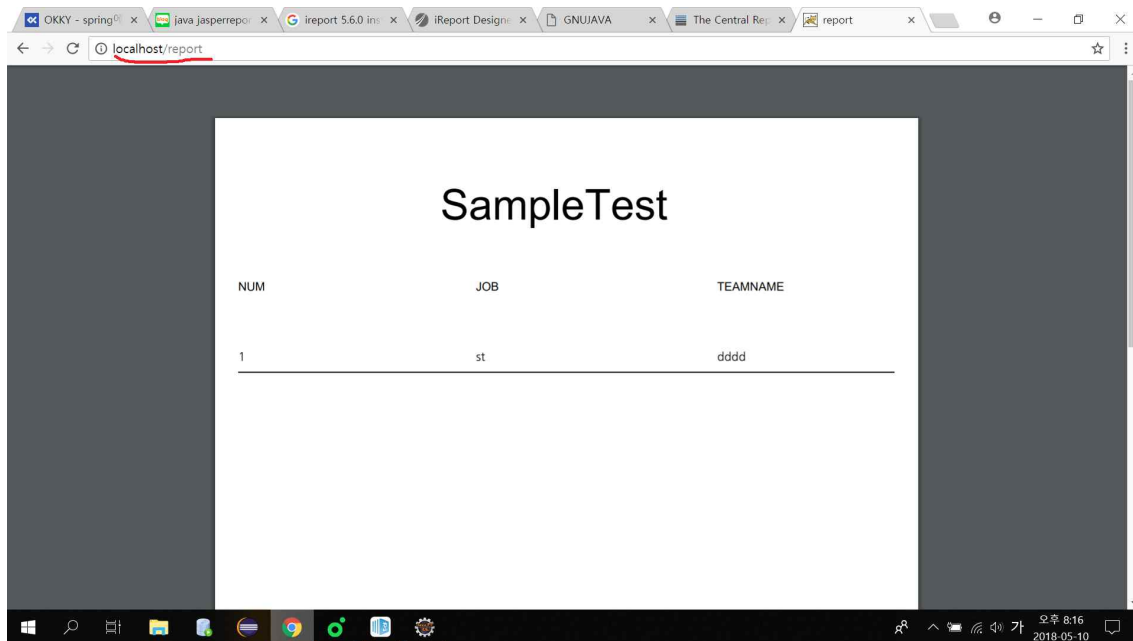
report메서드를 보면된다. 테스트예제기 때문에 DB에 접근을 안하고 임시로 VO에 값을 넣어 줬다. 활용할땐 디비에 접근해 리스트타입으로 데이터를 갖고오면된다.

데이터를 담은 리스트를 jrbeancollectiondatasource를 통해 jasperreport엔진에 전달해준다.

컨트롤러 끝~

```
1 package kr.or.ddit;
2
3
4 import java.util.ArrayList;
5
6 @Controller
7 public class ReportController {
8
9     @RequestMapping(value="/")
10    public String index() {
11        return "index";
12    }
13
14    @RequestMapping(value="/report", method={RequestMethod.POST, RequestMethod.GET})
15    public ModelAndView report(ModelAndView modelAndView) throws Exception{
16
17        Map<String, Object> parameterMap = new HashMap<String, Object>();
18        List<TestVO> userList = new ArrayList<TestVO>();
19        TestVO vo = new TestVO();
20        vo.setNUM("1");
21        vo.setJOB("st");
22        vo.setTEAMNAME("dddd");
23        userList.add(vo);
24
25        JRDataSource dataSource = new JRBeanCollectionDataSource(userList);
26        parameterMap.put("datasource", dataSource);
27        modelAndView = new ModelAndView("pdfReport", parameterMap);
28
29        return modelAndView;
30    }
31 }
```


서버를 실행하고 컨트롤러에서 지정한 report를 호출하면~ 잘나옴.



내가 참고한 사이트들... 최소 30곳은 봤지만 그 중 쓸만한 사이트만 링크

iReport 디자인에 참고 - <https://community.jaspersoft.com/wiki/designing-report>

스프링과의 연동에 참고 -

<https://m.blog.naver.com/junsu60/220477294301>

http://gnujava.com/board/article_view.jsp?article_no=6739&menu_cd=18&board_no=5&table_cd=EPAR01&table_no=01