



IREPORT 5.6



La presente guía es un instructivo rápido de como consumir reportes creados con Ireport en Java EE.

La forma más fácil de explicarlo es mediante un flujo de 3 pasos.

- 1) La vista se comunica con el controlador.
- 2) El controlador se comunica con la clase del reporte.
- 3) El reporte invoca un archivo de extensión .jasper y fin del proceso.

Así de simple.

COMO CONSUMIR UN REPORTE EN IREPORT FLUJO

controladores ▼ 🖺 reportes Vista ReporteConDosParametrosFecha.xhtml rptfechas.java ImprimirAltasBean.java rptaltas.jasper dbConexcion.java ReporteConUnParametro.xhtml ImprimeReciboBean.java dbConexcion.java rptrecibo.java rptrecibodepago.jasper La vista invoca al controlador, la vista El controlador invoca al La clase de conexión En resumen, es un Los archivos .jasper, son manda los datos al controlador y el reporte o "clase" o código es solo la cadena de código de programa los reporte compilados controlador, enviara los datos a la clase de programa que va generar conexión a la base de que va permitir la que vamos a publicar del reporte. el reporte. datos, que es generación de un dentro de un aplicativo web de Java EE. necesaria para la objeto tipo MIME, que puede PDF, EXCEL, conexión del report WORD ETC. Y que va invocar un archivo tipo .jasper de iReport







Los controladores generalmente son de tipo Request o ViewScope y van invocar una clase del reporte.

Esa clase de tipo reporte es la que contiene toda la programación relativa con iReport.

El controlador es el responsable de pasarle o entregarle, todos los parámetros y la ruta física del reporte a la clase de iReport.

En el ejemplo del recibo del empleado.

Hacemos el envío del número de recibo a la clase y desde el controlador.



Y la clase rptrecibo a su vez recibe el parámetro y el path de la ubicación del reporte, tal como lo apreciamos en los pasos 1 y 3.

```
public class rptrecibo {
    public void rptRecibo()
    {
    }
    public void getReporte(String ruta, int var NroRecibo) throws ClassNotFoundException
             , InstantiationException, IllegalAccessException, SQLException, ParseException
        Connection conexion = null;
        try {
        //Conexcion a JODBC Mysql
       conexion = getConnection();
        //Se definen los parametros si es que el reporte necesita
       Map parameter = new HashMap();
          El put de la variable debe ser identico al creado en el reporte
       parameter.put("nro_Recibo", var_NroRecibo);
       File file = new File(ruta);
       HttpServletResponse response = (HttpServletResponse) FacesContext.getCurrentInstance().getE:
       JasperReport jasperReport = (JasperReport) JRLoader.loadObjectFromFile(file.getPath());
       JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, parameter, conexion);
```







Parámetros de tipo Fecha

Generalmente en una clase de tipo request declaramos dos variables de tipos creamos el método que va consumir el XHTML, y le mandamos los parámetros a la clase de reporte <rptfechas>.

```
▼ 🔓 Source Packages
                               61
  ▼ III controladores
                               62
                                        //Declaracion del Metodo para invocar el reporte desde el xhtml
        ImprimeReciboRean java
                              63
                                        public void rptImprimirAltas() throws SQLException, ClassNotFoundException
    🚳 Impi
                              64
                                 , InstantiationException, IllegalAccessException, ParseException {
  ▼ E reportes
                               65
     rptfechas.java
                                            //Instancia la clase o el metodo rptAltas, que es la encargada de generar el un archivo pdf
     rptrecibo.java
                               67
                                            ▼ 🖺 sys
      dbConexcion.java
                                            // Se captura y se instancia el Contexto de la Vista levantada por View<mark>S</mark>cope
     sysutil.java
                                            FacesContext fc = FacesContext.getCurrentInstance();
                              70
     sysutillmpl.java
                              71
▶ math Test Packages
                              72
                                                            servlet para determinar la ruta real de la Aplicacion
► libraries
                                            ServletContext sc = (SarvletContext) fc.getExternalContext().getContext();
                              73
► a Test Libraries
                                                          sc.getRealPath("rpt/rptaltas.jasper");
                              74
                                            String path =
75
    MANIFEST.MF
                              76
                                            rpt.getReporte(path, this.fdesde, this.fhasta);
   context.xml
                              77
   web.xml
                              78
                                              Cuando finaliza de procesar el Contexto del Servlet dentro de la vista, lo muestra en pa
                              79
                                            fc.getCurrentInstance().responseComplete();
```

LA CLASE RPTFECHA, CONTIENE UN METODO LLAMADO getReporte, que el se expone y consume en los controladores y recibe dos parámetros de tipo fecha y la ruta física donde se encuentra el reporte.

```
public void getReporte(String ruta, Date fdesde, Date fhasta ) throws ClassNotFoundException
       , InstantiationException, IllegalAccessException, SQLException, ParseException {
Connection conexion = null;
//Declaramos un objecto que hereda todos los metodos y propiedades la sysutilImpl()
sysutil objFecha = new sysutilImpl();
trv {
//Conexcion a JODBC Mysql
conexion = getConnection();
 //Se definen los parametros si es que el reporte necesita
Map parameter = new HashMap();
    iReporte solo trabaja bien varias de Tipo TimeStamp para las fechas
java.sql.Timestamp desde = objFecha.getDateTime(fdesde); //Convertimos lo datos de Tipo Date a
                                                                                                    Time:
java.sql.Timestamp hasta = objFecha.getDateTime(fhasta);
                                                            //Convertimos lo datos de Tipo Date a
parameter.put("p desde", desde); // Usamos el put para enviarle los datos al reporte, la variable "p de
parameter.put("p_hasta", hasta); // Usamos el put para enviarle los datos al reporte, la variable "p_ha
 File file = new File(ruta);
 HttpServletResponse response = (HttpServletResponse) FacesContext.getCurrentInstance().getExternalConte
```