

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)



Control de autenticación y
acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)

AdictosAlTrabajo**Terrakas 1x04**
¡¡Ya está en la web!! :-)
terrakas.com

Entra en Adictos a través de  

E-mail

Contraseña

[Inicio](#) [Quiénes somos](#) [Formación](#) [Comparador de salarios](#) [Nuestro libro](#) [Más](#)

» Estás en: Inicio » Tutoriales » JRBeanCollectionDataSource: trabajando con colecciones de datos básicos en ...

**Jose Manuel Sánchez Suárez**

Consultor tecnológico de desarrollo de proyectos informáticos.

Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

[Ver todos los tutoriales del autor](#)**Catálogo de servicios Autentia****Fecha de publicación del tutorial: 2012-10-22**Tutorial visitado 1 veces [Descargar en PDF](#)**JRBeanCollectionDataSource: trabajando con colecciones de datos básicos en Jasper Report.****0. Índice de contenidos.**

- 1. Introducción.
- 2. Entorno.
- 3. Plantilla jrxml.
- 4. Generación del informe desde java.
- 5. Conclusiones.

1. Introducción

Son muchos los tutoriales publicados en adictos al trabajo sobre jasperReport e iReport y, en concreto, ya vimos de la mano de Saúl como disponer de una [fuente de datos basada en una colección de beans](#).

En este tutorial vamos a ir un poco más allá porque no vamos a tener fuente de datos ;), usaremos el mismo concepto pero en vez de obtener una colección de beans de una fuente de datos, trabajaremos con una colección de "tipos básicos" pasada como parámetro.

Parece un contrasentido pero no, en iReport no es necesario definir una fuente de datos tipada para la generación de nuestros informes. Imaginad que no necesitamos iterar, que únicamente necesitamos la facilidad de exportar en distintos formatos o que queremos hacer uso de la plantilla para un remplazo de huecos.

En este tutorial vamos revisar dos cuestiones:

- cómo trabajar con parámetros definiendo una fuente de datos vacía, y
- cómo iterar por una colección de tipos básicos para imprimir su contenido.

El ejemplo que vamos a usar es la generación de la ficha de un cliente, una empresa que tiene distinta información no necesariamente tipada y que no se obtiene de base de datos en la propia generación del informe. Vamos a realizar una prueba simple pasando como parámetro al informe una colección de empleados, solo los nombres como cadenas.

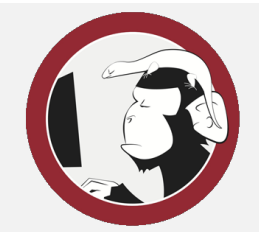
2. Entorno.

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil MacBook Pro 15' (2.4 GHz Intel Core i7, 8GB DDR3 SDRAM).
- Sistema Operativo: Mac OS X Lion 10.7.4
- iReport 4.7.1
- Jasper Report 4.7.1

3. Creación de la plantilla jrxml.

Lo primero es crear una plantilla vacía dentro del entorno de iReport, le añadimos una cabecera y un pie de página y seleccionamos como fuente de datos un "Empty Datasource":

**Síguenos a través de:****Últimas Noticias**

» Participamos en la Carrera de las Empresas 2012

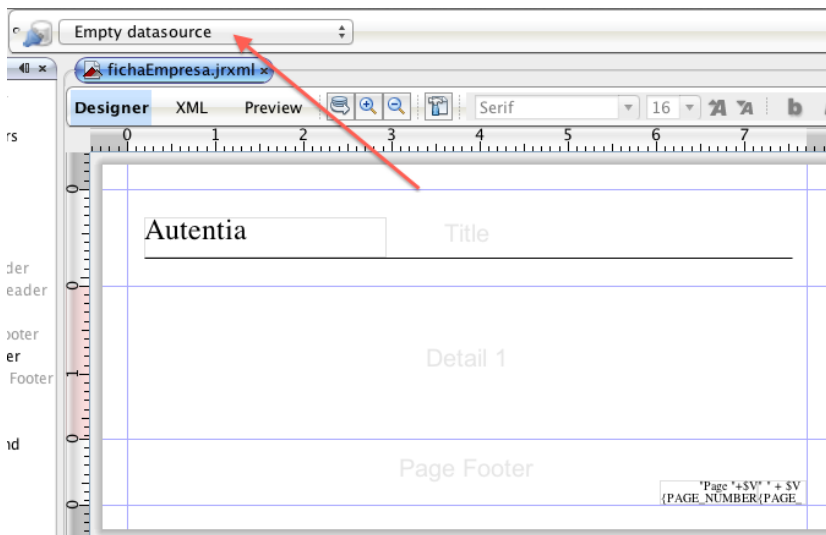
» ¡¡¡Terrakas 1x04 recién salido del horno!!!

» Estreno Terrakas 1x04: "Terraka por un día"

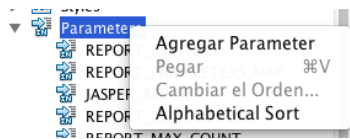
» Nuevos cursos de gestión de la configuración en IOS y Android

» La regla del Boy Scout y la Oxidación del Software

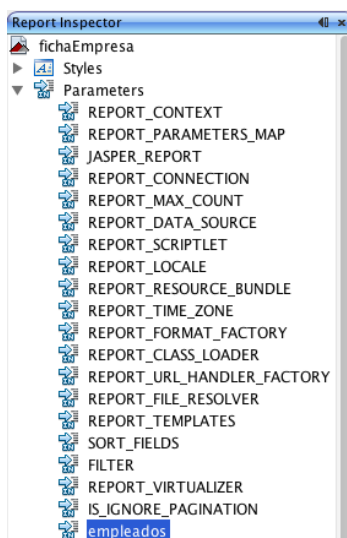
[Histórico de noticias](#)



Como no vamos a trabajar con una fuente de datos, lo siguiente es crear un parámetro, para ello "botón derecho" sobre "Parameters" > "Agregar Parameter":

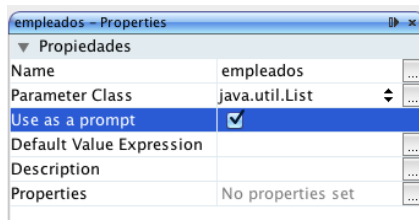


Le asignamos un nombre, en nuestro caso "empleados":

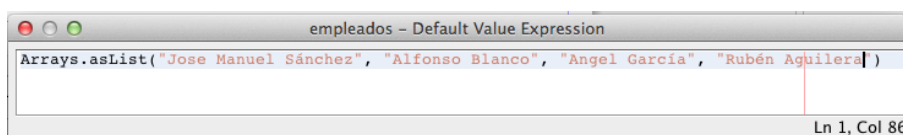


Y accediendo a la ventana de propiedades, le damos una tipología en "Parameter Class", en nuestro caso "java.util.List" y, además:

- con "Use as prompt" le indicamos si queremos que, en el entorno del iReport, muestre una ventana de petición del valor para las pruebas, y
- en "Default Value Expression" podemos indicar un valor por defecto para la propiedad.



Pulsando sobre "Default Value Expression" podemos asignar algo como lo siguiente para añadir un List<String>



Últimos Tutoriales

- » Código de barras con iReport
- » Uso de StoryBoards en desarrollo iOS
- » [Code Snippets en XCode 4](#)
- » Mountain Lion - Git "Command Not Found"
- » Integración de Spring Web Flow 2 con JSF2

Últimos Tutoriales del Autor

- » Integración de Spring Web Flow 2 con JSF2
- » JSF2 Flash scope
- » Obtención de los literales de i18n de base de datos en JSF2.
- » Introducción a Spring Data: soporte para JPA.
- » Plantillas para los métodos equals y hashCode en Eclipse, usando la librería de Apache Commons Lang.

Categorías del Tutorial

[Informes](#)

Últimas ofertas de empleo

- 2011-09-08 [Comercial - Ventas - MADRID.](#)
- 2011-09-03 [Comercial - Ventas - VALENCIA.](#)
- 2011-08-19 [Comercial - Compras - ALICANTE.](#)
- 2011-07-12 [Otras Sin catalogar - MADRID.](#)
- 2011-07-06 [Otras Sin catalogar - LUGO.](#)

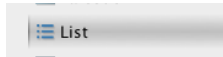
 Jose Manuel Sánchez
sanchezsuaresj

[adictosaltrabaj](#) Uso de StoryBoards en el desarrollo con iOS - [kcy.me/brf6](#), de nuevo de la mano de [@rcanalesmora](#)
3 days ago · reply · retweet · favorite

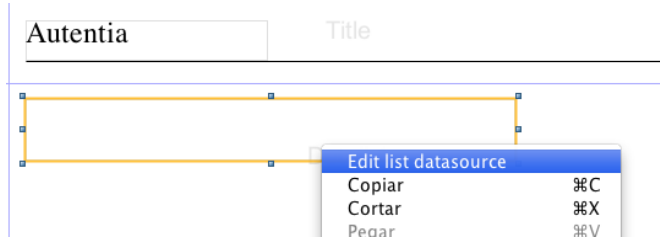
[adictosaltrabaj](#) Code Snippets en XCode 4 - [kcy.me/bq1v](#) de la mano de [@rcanalesmora](#)
4 days ago · reply · retweet · favorite

[sanchezsuaresj](#) Integración de Spring Web Flow 2 con #JSF2 - [kcy.me/bo9k](#) @adictosaltrabaj

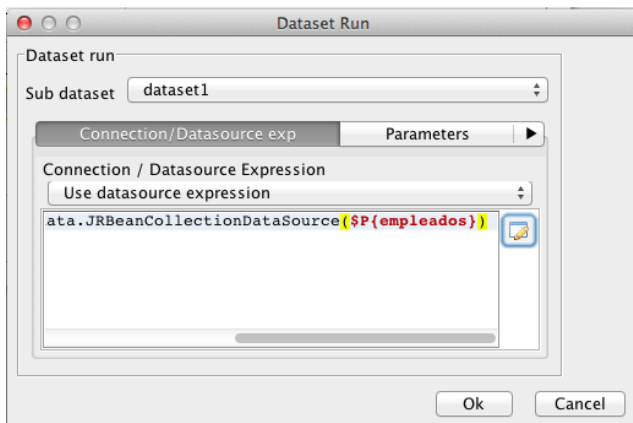
Con ello, ya tenemos preparado nuestro parámetro para el entorno de pruebas y ahora vamos a añadir un componente visual de tipo lista para iterar por su contenido y mostrar el nombre de los empleados.



Pulsando sobre icono anterior y arrastrándolo al área de la plantilla lo tendremos disponible para su edición:

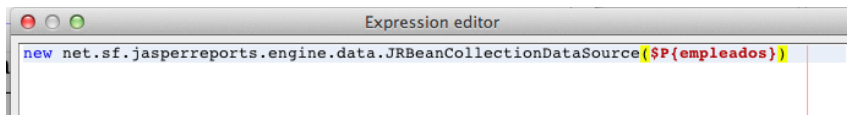


Una vez incluido en la sección correspondiente de la plantilla "botón derecho" > "Edit datasource" mostrará una ventana como la que sigue:



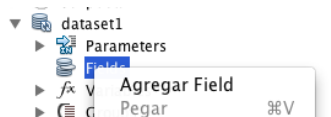
En este punto son importantes dos cuestiones:

- "Sub dataset" definirá los parámetros de entrada, campos, variables,... que vivirán en el ámbito de la lista y que, no tienen por qué coincidir con los de la plantilla padre, ahí será donde definiremos un campo "ad hoc".
- Connection / Datasource Expression: indica la fuente de datos para la lista que puede ser la misma fuente de datos que el informe padre u otra, en nuestro caso definimos una expresión usando la clase `net.sf.jasperreports.engine.data.JRBeanCollectionDataSource` y pasando como argumento al constructor una referencia al parámetro anteriormente definido.

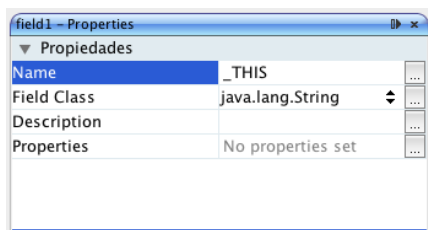


Lo siguiente será definir un campo que haga referencia a cada uno de los items de la colección, si tuviéramos un objeto tipado, añadiríamos los campos a mostrar de la clase o el bean en cuestión, como trabajamos con tipos básicos añadiremos un campo con la palabra reservada `_THIS`, que hará referencia a cada una de las cadenas dentro de la iteración interna de la lista.

Sobre el dataset1, el que usa la lista, pulsamos "botón derecho" en "Fields" > "Agregar Field"



El nombre para nuestro campo es la palabra reservada `_THIS`



Con ello podemos arrastrar dentro del contenido del componente de lista el campo `${_THIS}`

preparando el cambio para faces
flow

6 days ago · reply · retweet · favorite

sanchezsuaresj Uso de CSS3
Media Queries: consulta de
Medios y responsive Design...



Join the conversation

Empleados

Si compilamos el informe para una previsualización en el entorno de iReport, primero nos solicita un valor para nuestro parámetro "empleados", si os fijáis en el "suggest" indica el tipo y un ejemplo

Parameter prompt

You can provide a value for the parameter:
empleados

The class type is: java.util.List (foo,bar,test)

OK Use default

Si pulsamos sobre "Use default" usará lo definido por defecto a nivel de parámetro y debería mostrar un informe como el siguiente:

Autentia

Empleados

Jose Manuel Sánchez
Alfonso Blanco
Angel García
Rubén Aguilera

Podéis descargar el jrxml usado en el ejemplo en el siguiente enlace: [Ficha de empresa en formato jrxml](#).

4. Generación del informe desde java.

Si el informe anterior lo tuviéramos que generar desde código bastaría con un servicio como el siguiente:

```
1 package com.autentia.training.jasperReports;
2
3 import java.util.Map;
4
5 import net.sf.jasperreports.engine.JREmptyDataSource;
6 import net.sf.jasperreports.engine.JRException;
7 import net.sf.jasperreports.engine.JasperCompileManager;
8 import net.sf.jasperreports.engine.JasperExportManager;
9 import net.sf.jasperreports.engine.JasperFillManager;
10 import net.sf.jasperreports.engine.JasperPrint;
11 import net.sf.jasperreports.engine.JasperReport;
12 import net.sf.jasperreports.engine.design.JasperDesign;
13 import net.sf.jasperreports.engine.xml.JRXmlLoader;
14
15 public class ReportExporter {
16
17     public void toPDF(String reportName, Map<String,Object> parameters) throws JRExcepti
18
19         final String reportSource = getClass().getClassLoader().getResource(reportName).
20
21         final JasperDesign jd = JRXmlLoader.load(reportSource);
22
23         final JasperReport report = JasperCompileManager.compileReport(jd);
24
25         final JasperPrint print = JasperFillManager.fillReport(report, parameters, new J
26
27         final String reportTarget = reportSource.substring(0,reportSource.lastIndexOf('/')
28
29         JasperExportManager.exportReportToPdfFile(print, reportTarget);
30
31     }
32
33 }
```

Recibe como parámetro el nombre de fuente jrxml y un mapa de parámetros a pasar al informe.

Obtenemos la ubicación en disco para escribir el informe en el mismo path, usamos un JREmptyDataSource y le pasamos el mapa de parámetros.

Podemos realizar una prueba de generación a través de un método main, pasándole un listado de empleados como cadenas dentro del mapa de parámetros:

```
1 private static final List<String> empleados =
2     Arrays.asList("Jose Manuel Sánchez", "Alfonso Blanco", "Angel García", "Rubén Ag
3
4 public static void main(String[] args) throws JRException {
```

```
5 final Map<String,Object> parameters = new HashMap<String,Object>();
6 parameters.put("empleados", empleados);
7 final ReportExporter reportExporter = new ReportExporter();
8 reportExporter.toPDF("fichaEmpresa.jrxml", parameters);
9 }
```

El resultado será el mismo que el obtenido desde el entorno de iReport.

5. Conclusiones.

Lo ideal es trabajar con objetos tipados, pero esta posibilidad nos abre las puertas para trabajar con objetos desnormalizados.

Un saludo.

Jose

jmsanchez@autentia.com

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)



Por favor, vota +1 o compártelo si te pareció interesante



Anímate y coméntanos lo que pienses sobre este **TUTORIAL**:

» **Regístrate** y accede a esta y otras ventajas «



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Copyright 2003-2012 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

