



---

## **Framework Corporatiu J2EE**

### **Servei de Reporting**

**Versió 1.2**

Barcelona, 13 / febrer / 2007



## Històric de modificacions

Data	Autor	Comentaris	Versió
16/01/2006	Atos Origin, sae openTrends	Versió inicial del document	1.0
13/02/2007	Atos Origin, SAE	Versió 1.2 d'OpenFrame	1.2

### Llegenda de Marcadors



## Índex

<b>1.</b>	<b>INTRODUCCIÓ .....</b>	<b>4</b>
1.1.	PROPÒSIT .....	4
1.2.	CONTEXT I ESCENARIS D'ÚS .....	4
1.3.	VERSIONS I DEPENDÈNCIES .....	4
1.3.1.	<i>Versions</i> .....	5
1.3.2.	<i>Dependències Bàsiques</i> .....	5
1.3.3.	<i>Dependències Addicionals</i> .....	5
1.4.	A QUI VA DIRIGIT .....	5
1.5.	DOCUMENTS I FONTS DE REFERÈNCIA .....	6
1.6.	GLOSSARI .....	6
<b>2.</b>	<b>DESCRIPCIÓ DETALLADA .....</b>	<b>7</b>
2.1.	ARQUITECTURA I COMPONENTS .....	7
2.1.1.	<i>Interfícies i Components Genèrics</i> .....	7
2.1.2.	<i>Components implementació de JasperReports</i> .....	8
2.2.	INSTAL·LACIÓ I CONFIGURACIÓ .....	10
2.2.1.	<i>Instal·lació</i> .....	10
2.2.2.	<i>Configuració</i> .....	10
2.3.	UTILITZACIÓ DEL SERVEI .....	15
2.3.1.	<i>Realitzar el disseny del report</i> .....	15
2.3.2.	<i>Registrar el report en el servei (fitxer views.xml)</i> .....	16
2.3.3.	<i>Modelar les dades a mostrar (opcional)</i> .....	17
2.3.4.	<i>Invocar el report</i> .....	17
2.4.	EINES DE SUPORT .....	18
2.4.1.	<i>iReport</i> .....	18
2.5.	INTEGRACIÓ AMB ALTRES SERVEIS .....	20
2.6.	PREGUNTES FREQUENTS .....	20
<b>3.</b>	<b>EXEMPLES .....</b>	<b>21</b>
<b>4.</b>	<b>ANNEXOS .....</b>	<b>22</b>

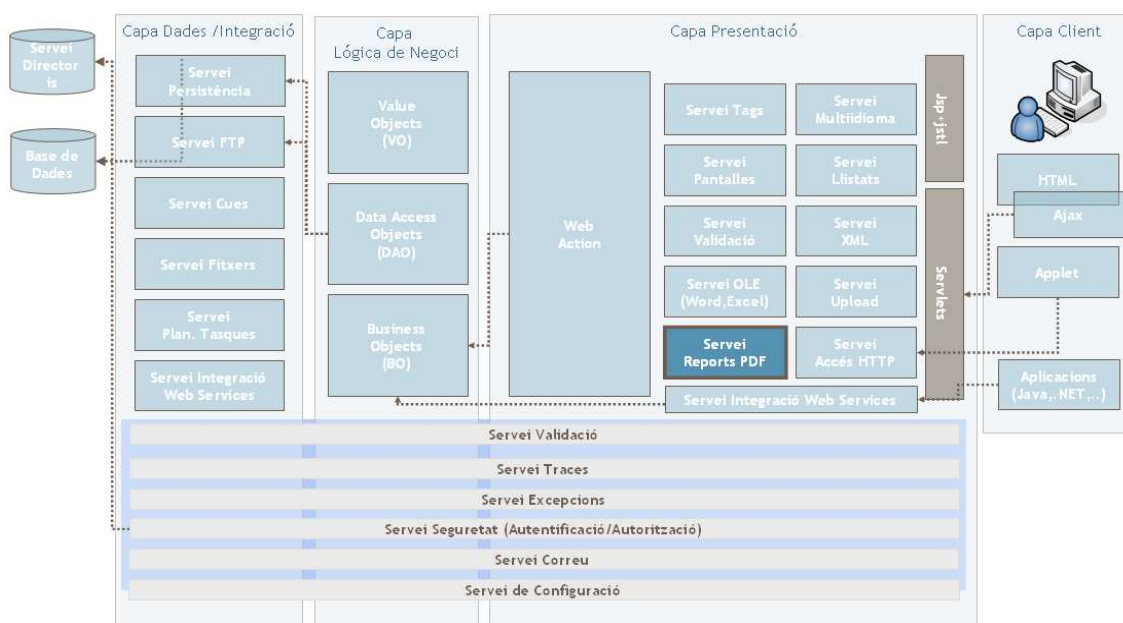
## 1. Introducció

### 1.1. Propòsit

El servei de Reporting d'openFrame permet integrar d'una manera senzilla dades des de fonts heterogènies i mostrar-les en diferents formats: PDF, XLS, CSV i HTML, dins d'un marc homogeni de desenvolupament.

### 1.2. Context i Escenaris d'Ús

El servei de Reporting es troba dins dels serveis de presentació d'openFrame.



El seu ús és necessari en cas de voler presentar informació en format report (ja sigui PDF o els formats que suporta el servei).

### 1.3. Versions i Dependències

En el present apartat es mostren quines són les versions i dependències necessàries per fer ús del Servei.

Dins la llista de dependències es mostren diferenciades:

- Dependències bàsiques. Llibreries necessàries per fer ús del servei.
- Dependències addicionals. Aquestes dependències són necessàries per poder fer ús de característiques concretes del servei o per l'ús dels tests unitaris proporcionats amb el servei.



### 1.3.1. Versions

No s'han produït canvis respecte la versió 1.0.

### 1.3.2. Dependències Bàsiques

Nom	Tipus	Versió	Descripció
openFrame-core	jar	1.0	
openFrame-services-logging	jar	1.0	Utilitzar també les dependències del Servei de Logging
openFrame-services-exceptions	jar	1.0	Utilitzar també les dependències del Servei d'Excepcions
spring	jar	1.2.5	<a href="http://www.springframework.org">http://www.springframework.org</a>
jasperreports	jar	1.0.2	
servlet-api	jar	2.4	
itext	jar	1.3.1	
poi	jar	3.0	
commons-beanutils-core	jar	1.7.0	
commons-collections	jar	3.1	
commons-digester	jar	1.7	
commons-logging	jar	1.0.4	<a href="http://jakarta.apache.org/commons">http://jakarta.apache.org/commons</a>
jdtcore	jar	3.1.0	

En cas de utilitzar un servei de openFrame és important que es facin servir també les dependències del servei en qüestió.

### 1.3.3. Dependències Addicionals

- Proves Unitàries del Servei

Nom	Tipus	Versió	Descripció
junit	jar	3.8.1	
Spring-mock	jar	1.2.5	<a href="http://www.springframework.org">http://www.springframework.org</a>
easymock	jar		

Veure l'apartat 'Instal·lació i configuració' per a més detall.

## 1.4. A qui va dirigit

Aquest document va dirigit als següents perfils:

- Programador. Per conèixer l'ús del servei.
- Arquitecte. Per conèixer quins són els components i la configuració del servei.



## 1.5. Documents i Fonts de Referència

- [1] JasperReports <http://jasperreports.sourceforge.net/>
- [2] iReport <http://ireport.sourceforge.net>

## 1.6. Glossari

### JasperReports

Es tracta d'una eina de reporting *open source* que és capaç de gestionar les dades obtingudes (d'una -o més- connexions JDBC, TableModels, JavaBeans, XML, etc.) i mostrar-les segons un disseny definit en XML.

## 2. Descripció Detallada

### 2.1. Arquitectura i Components

**openFrame** ofereix una arquitectura del servei de reporting totalment deslligada de qualsevol implementació.

Els components podem classificar-los en:

- Interfícies i Components Genèrics. Interfícies del servei i components d'ús general amb independència de la implementació escollida.
- Implementació de les interfícies basada en JasperReports.

#### 2.1.1. Interfícies i Components Genèrics



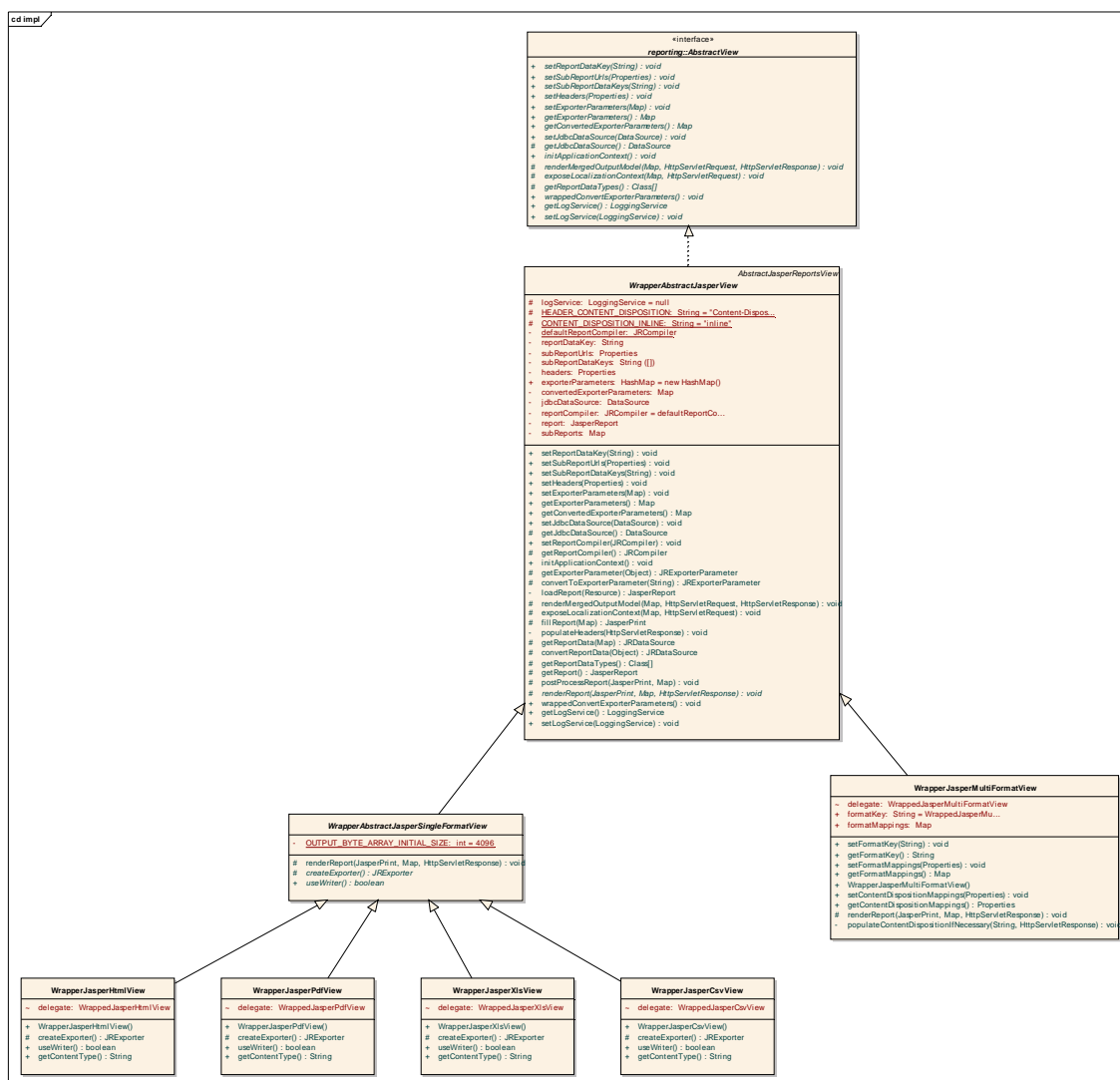
Component	Package	Descripció
-----------	---------	------------



Component	Package	Descripció
ReportingController	net.opentrends.openframe.services.reporting	Controlador de Spring necessari per renderitzar la visualització d'un report com una view. <b>Serà cridat, pel framework quan sigui necessari.</b>
AbstractView	net.opentrends.openframe.services.reporting	Interfície del servei pels tractament dels reports. <b>Ofereix entre d'altres (veure la documentació disponible al Javadoc per a més referència):</b> <ul style="list-style-type: none"><li>protected renderMergedOutputModel(Map model, HttpServletRequest request, HttpServletResponse response); les subclasses han d'implementar el mètode per renderitzar els reports segons les dades del model.</li></ul>

### **2.1.2. Components implementació de JasperReports**





Component	Package	Descripció
WrapperAbstractJasperView	net.opentrends.openframe.services.reporting.impl	Implementació de la interfície AbstractView. Es tracta de la classe base per tots els tipus de reporting. Coordina el procés de renderització.
WrapperJasperMultiFormatView	net.opentrends.openframe.services.reporting.impl	Classe que permet que el format de renderització es faci en temps d'execució utilitzant un paràmetre en el model.
WrapperAbstractJasperSingleFormatView	net.opentrends.openframe.services.reporting.impl	Classe que extén de WrapperAbstractJasperView i proveeix de la lògica bàsica de renderització pels tipus de reporting amb format prefixat (PDF, HTML, etc.)
WrapperJasperHtmlView	net.opentrends.openframe.services.reporting.impl	Implementació de

Component	Package	Descripció
ew	eporting.impl	WrapperAbstractJasperSingleFormatView que renderitza els resultats dels reports en format HTML.
WrapperJasperPdfView	net.opentrends.openframe.services.reporting.impl	Implementació de WrapperAbstractJasperSingleFormatView que renderitza els resultats dels reports en format PDF.
WrapperJasperXlsView	net.opentrends.openframe.services.reporting.impl	Implementació de WrapperAbstractJasperSingleFormatView que renderitza els resultats dels reports en format XLS.
WrapperJasperCsvView	net.opentrends.openframe.services.reporting.impl	Implementació de WrapperAbstractJasperSingleFormatView que renderitza els resultats dels reports en format CSV.

Veure la documentació disponible al Javadoc per a més referència.

## 2.2. Instal·lació i Configuració

### 2.2.1. Instal·lació

La instal·lació del servei requereix de la utilització de la llibreria 'openFrame-services-reporting' i les dependències indicades a l'apartat 'Introducció - Versions i Dependències'.

### 2.2.2. Configuració

La configuració del Servei de Reporting implica realitzar els següents passos:

- 1) Definir la configuració bàsica del servei (abans de generar qualsevol vista)
- 2) Definir la localització dels reports i el seu tipus de visualització

#### Definició de la Configuració Bàsica

La configuració bàsica implica:

- a) Definir el mapeig de les urls amb sufix '.report' al Servlet principal

```
<servlet>
  <servlet-name>application</servlet-name>
  <servlet-class>

  org.springframework.web.servlet.DispatcherServlet
  </servlet-class>
  <load-on-startup>2</load-on-startup>
</servlet>
...
<servlet-mapping>
  <servlet-name>application</servlet-name>
  <url-pattern>*.report</url-pattern>
</servlet-mapping>
...
```



***Fitxer de configuració: web.xml***

*Ubicació proposada:* <PROJECT\_ROOT>/src/main/webapp/WEB-INF

Definir que totes les URIs que finalitzin amb “.report” les tractarà **openFrame** amb el servlet **org.springframework.web.servlet.DispatcherServlet**:

Usar el següent codi:

```
...
<servlet>
  <servlet-name>application</servlet-name>
  <servlet-class>
    org.springframework.web.servlet.DispatcherServlet
  </servlet-class>
  <load-on-startup>2</load-on-startup>
</servlet>
...
<servlet-mapping>
  <servlet-name>application</servlet-name>
  <url-pattern>*.report</url-pattern>
</servlet-mapping>
...
```

b) Definir la localització de les definicions de les vistes

```
<bean id="viewResolver"
...>
```

***Fitxer de configuració: openFrame-services-reporting.xml***

*Ubicació proposada:* <PROJECT\_ROOT>/src/main/resources/spring

Configurar un bean amb un identificador (es recomana usar 'viewResolver' ) i els següents atributs:

Atribut	Requerit	Descripció
class	Sí	Defineix el tipus de resolver de Spring utilitzarem per indicar on trobar el fitxer amb la definició dels reports.  Es recomana utilitzar la notació XML: org.springframework.web.servlet.view.XmlViewResolver

Configurar les següents propietats:

Propietat	Requerit	Descripció
-----------	----------	------------



Propietat	Requerit	Descripció
location	Sí	Referència a la localització del fitxer amb la definició dels reports.

Exemple:

```
...  
<!-- REPORTING service -->  
<bean id="viewResolver" class="org.springframework.web.servlet.view.XmlViewResolver" >  
  <property name="location" value="classpath:reports/views.xml" />  
</bean>  
...
```

c) Declaració dels forwards globals per poder usar el servei des d'una Action

```
<forward name="reportPDF"  
  ...>  
<forward name="reportXLS"  
  ...>
```

***Fitxer de configuració: openFrame-services-reporting.xml***

*Ubicació proposada:* <PROJECT\_ROOT>/src/main/resources/spring

Definir un forward per cada format de renderització emprat, seguint el següent patró:

```
<forward      name="report<FORMAT>"  
  path="/view.report?contentType=<format>"  
  redirect="true" />
```

Usar el següent codi:

```
...  
<global-forwards>  
...  
  <!-- Forward to report -->  
  <forward name="reportPDF" path="/view.report?contentType=pdf" redirect="true"  
/>  
  <forward name="reportXLS" path="/view.report?contentType=xls" redirect="true"  
/>  
  <forward name="reportCSV" path="/view.report?contentType=csv" redirect="true"  
/>  
  <forward name="reportHTML" path="/view.report?contentType=html" redirect="true" />  
...  
</global-forwards>  
...
```



d) Declaració del control·lador assignat als reports

```
<bean id="reportingController"
    ...>
<entry key="/**/*.*.report"
    ...>
<prop key="view.report">...
```

**Fitxer de configuració: openFrame-services-reporting.xml**

Ubicació proposada: <PROJECT\_ROOT>/src/main/resources/spring

Definir el controlador a utilitzar en la renderització dels reports, per això s'ha de definir el següent bean:

```
<bean id="reportingController" class="net.opentrends.openframe.services.reporting.ReportingController" />
```

En la definició del bean “ExportInterceptor”, a la propietat “redirections” s'ha d'afegir l'entrada:

```
<entry key="/**/*.*.report" value="/view.report" />
```

En la definició del bean “urlMapping”, a la propietat “mappings” s'ha d'afegir l'entrada:

```
<prop key="view.report">reportingController</prop>
```

Exemple:

```
...
<bean id="ExportInterceptor"
class="net.opentrends.openframe.services.web.spring.BufferedResponseSpringIntercep
tor">
  <property name="redirections">
    <map>
      ...
      <entry key="/**/*.*.report" value="/view.report" />
    </map>
  </property>
</bean>
<bean id="urlMapping"
class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
  ...
  <property name="mappings">
```



```
<props>
...
<prop key="view.report">reportingController</prop>
</props>
</property>
</bean>
...

<bean id="reportingController"
class="net.opentrends.openframe.services.reporting.ReportingController" />
```

```
<bean id="xxxReport"
...>
```

Localització dels reports i el  
seu tipus de visualització

### ***Fitxer de configuració: views.xml***

Ubicació proposada: <PROJECT\_ROOT>/src/main/resources/reports

La definició de cada report requereix configurar un bean amb un identificador (es recomana usar '<nom\_report>Report') i els següents atributs:

Atributs:

Atribut	Requerit	Descripció
class	Sí	Defineix la classe que s'emprarà per realitzar el renderitzat del report.  Exemple: net.opentrends.openframe.services.reporting.impl.WrapperJasperMultiFormatView

També es poden configurar les següents propietats:

Propietat	Requerit	Descripció
url	Sí	Localització del fitxer <i>jrxml</i> amb el disseny del report.
contentType	Sí	Format del fitxer a mostrar (pdf, xls, html, csv)

Exemple:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">

<beans>

...

```



```
<bean id="RotationReport"
      class="net.opentrends.openframe.services.reporting.impl.RotationReport"
      <property name="url"
value="classpath:reports/rotation/RotationReport.jrxml" />
    </bean>
    <bean id="CustomersReport"
class="net.opentrends.openframe.services.reporting.impl.WrapperJasperMultiFormatView" >
      <property name="url"
value="classpath:reports/customers/CustomersReport.jrxml" />
    </bean>
    ...
</beans>
```

## 2.3. Utilització del Servei

openFrame defineix 2 maneres d'integrar les dades d'un report:

- Utilitzant el "data source" específic proporcionat per JasperReports
- Injectant les dades des del model de l'aplicació (obtingudes mitjançant el servei de persistència, fitxers de *properties*, etc.)

Com exemple d'utilització del servei de Reporting s'inclou un exemple en el que es visualitza un report senzill. A continuació es mostren els passos a seguir per integrar un report dins d'una aplicació:

### 2.3.1. Realitzar el disseny del report

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jasperReport PUBLIC "-//JasperReports//DTD Report Design//EN"
"http://jasperreports.sourceforge.net/dtds/jasperreport.dtd">

<jasperReport name="RotationReport" pageWidth="595" pageHeight="842"
columnWidth="555" leftMargin="20" rightMargin="20" topMargin="30"
bottomMargin="30">
  <title>
    <band height="782">
      <staticText>
        <reportElement mode="Opaque" x="0" y="0" width="90"
height="350" backcolor="#c0c0c0"/>
        <textElement rotation="Left">
          <font size="14"/>
        </textElement>
        <text><![CDATA[rotation="Left"
textAlignment="Left"
verticalAlignment="Top"]]></text>
      </staticText>
      <staticText>
        <reportElement mode="Opaque" x="100" y="0" width="90"
height="350" backcolor="#c0c0c0"/>
        <textElement textAlignment="Center"
verticalAlignment="Bottom" rotation="Left">
          <font size="14"/>
        </textElement>
        <text><![CDATA[rotation="Left"
textAlignment="Center"
verticalAlignment="Bottom"]]></text>
      </staticText>
    </band>
  </title>
</jasperReport>
```



```

        <textElement textAlignment="Right"
verticalAlignment="Middle" rotation="Left">
        <font size="14"/>
        </textElement>
        <text><![CDATA[rotation="Left"
textAlignment="Right"
verticalAlignment="Middle"]]></text>
        </staticText>
        <staticText>
        <reportElement mode="Opaque" x="255" y="410" width="90"
height="350" forecolor="#ffffff" backcolor="#404040"/>
        <textElement textAlignment="Right"
verticalAlignment="Middle" rotation="Right">
        <font size="14"/>
        </textElement>
        <text><![CDATA[rotation="Right"
textAlignment="Right"
verticalAlignment="Middle"]]></text>
        </staticText>
        <staticText>
        <reportElement mode="Opaque" x="355" y="410" width="90"
height="350" forecolor="#ffffff" backcolor="#404040"/>
        <textElement textAlignment="Center"
verticalAlignment="Bottom" rotation="Right">
        <font size="14"/>
        </textElement>
        <text><![CDATA[rotation="Right"
textAlignment="Center"
verticalAlignment="Bottom"]]></text>
        </staticText>
        <staticText>
        <reportElement mode="Opaque" x="455" y="410" width="90"
height="350" forecolor="#ffffff" backcolor="#404040"/>
        <textElement rotation="Right">
        <font size="14"/>
        </textElement>
        <text><![CDATA[rotation="Right"
textAlignment="Left"
verticalAlignment="Top"]]></text>
        </staticText>
    </band>
</title>
</jasperReport>

```

### **Nota**

Per realitzar el disseny del report es recomana l'utilització d'una eina visual tipus iReport. Veure l'apartat "Eines de Suport".

## **2.3.2. Registrar el report en el servei (fitxer views.xml)**

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
"http://www.springframework.org/dtd/spring-beans.dtd">

<beans>
    ...
    <bean id="RotationReport"
class="net.opentrends.openframe.services.reporting.impl.WrapperJasperMultiFormatVi
ew" >

```





```
</bean>
...
</beans>
```

### 2.3.3. Modelar les dades a mostrar (opcional)

Per renderitzar el report amb dades modelades per un *Action*, s'ha de definir específicament un Map i informar-lo amb parells “nom/valor” que el servei de reporting utilitzarà per omplir el report al visualitzar-lo.

```
/**
 * Modeling data for report
 */
public ActionForward reportRotation(Category vo, StrutsContext context) {
    // Get data for render report
    Map model = getModel();

    // Save data in session
    context.getRequest().getSession()
        .setAttribute(ReportingController.REPORTING_CONTROLLER_MODEL, model);

    // Set view name to render
    context.getRequest().getSession()
        .setAttribute(ReportingController.REPORTING_CONTROLLER_REPORTID,
            "RotationReport");

    // Forward to report Controller
    return context.getActionMapping().findForward("reportPDF");
}

private Map getModel() {
    Map model = new HashMap();
    Collection beanData = getData();
    model.put("myBeanData", beanData);
    return model;
}

protected List getData() {
    List list = new ArrayList();
    for (int x = 0; x < 10; x++) {
        Category bean = new Category();
        bean.setId(""+x);
        bean.setName("openFrame");
        list.add(bean);
    }
    return list;
}
```

#### ⚠ Nota

Si el report utilitza el seu propi “*data source*” (o no mostra dades dinàmiques) no serà necessari realitzar aquest pas.

### 2.3.4. Invocar el report

Per invocar el report dins de l'aplicació es pot realitzar mitjançant:



- Un **Action** (si s'ha de modelar les dades):

```
http://localhost:8080/openFrame-samples-  
Petstore/categories.do?reqCode=reportRotation
```

- Un **servlet** (només si el report utilitza el seu propi “*data source*”). S'han d'especificar els següents paràmetres:
  - **reportId**: identificador del report segons s'hagi definit en el fitxer **views.xml**
  - **contentType**: tipus de format a visualitzar (csv, pdf, xls o html)

```
http://localhost:8080/openFrame-samples-  
Petstore/view.report?contentType=pdf&reportId=RotationReport
```

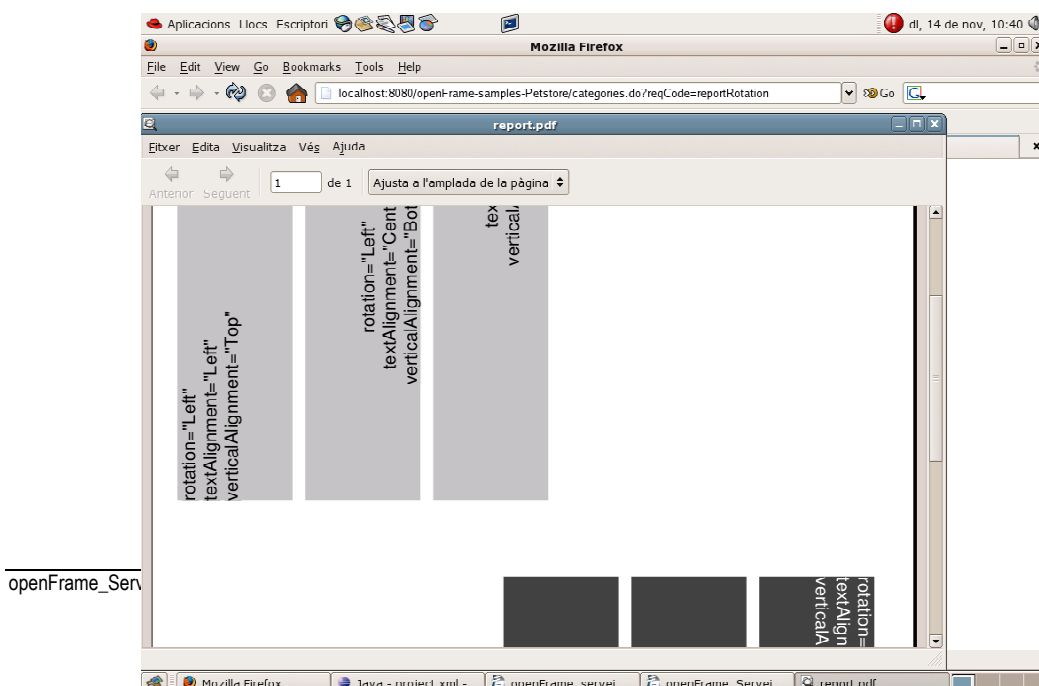
## 2.4. Eines de Suport

### 2.4.1. iReport

El disseny d'un report representa una plantilla que serà utilitzada pel servei per mostrar contingut dinàmic en el format escollit. Està definit en fitxers XML que han de tenir una estructura predefinida. Aquests XML s'han de compilar per poder realitzar les operacions d'omplir dades, aquesta tasca de compilació la realitza el servei de manera “on the fly”.

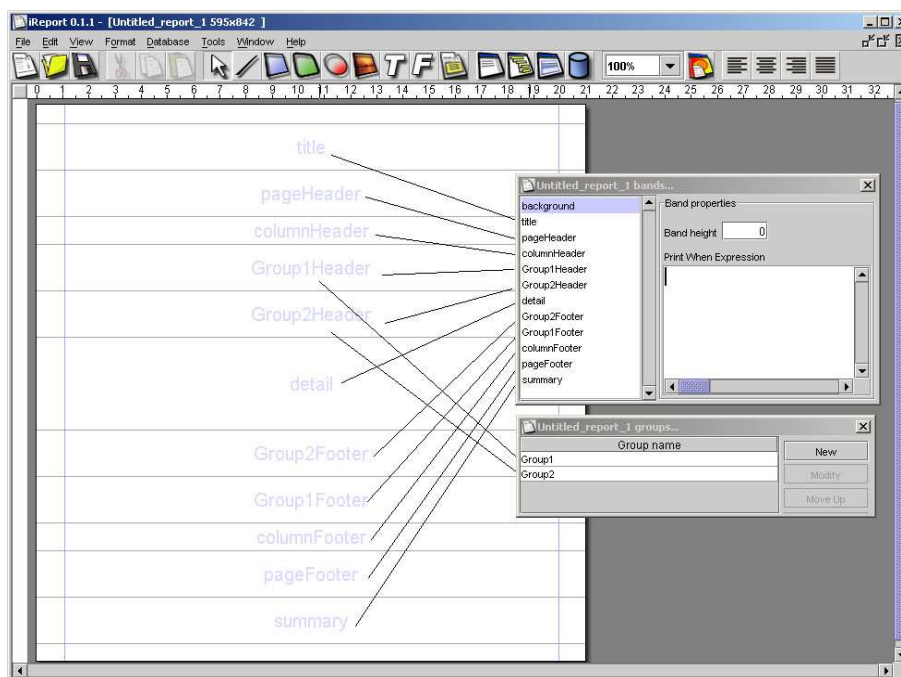
És aconsellable utilitzar una eina visual per realitzar el disseny dels reports, aquesta eina ha de facilitar l'edició de reports complexos amb gràfics, imatges, subreports, etc. Una eina que compleix tots aquests requisits és iReport <http://ireport.sourceforge.net>.

Algunes de les característiques que proporciona iReport són:

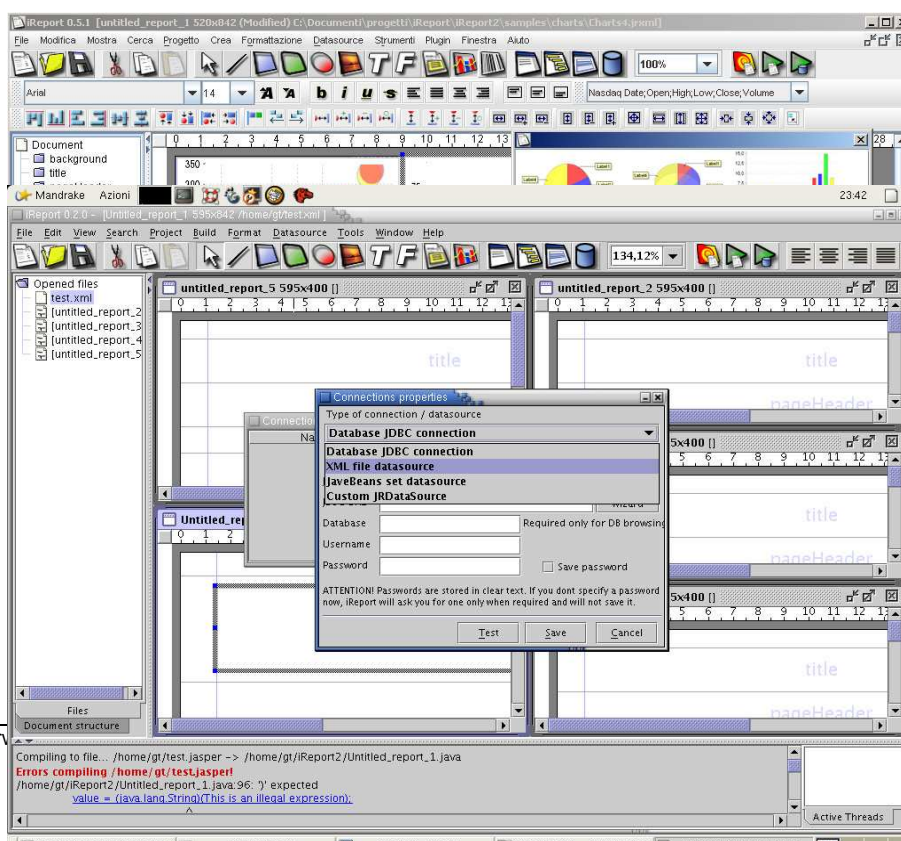




- Suport per gràfics:
- Agrupacions i seccions



- Diferents tipus de connexions





- Formateig del document

## **2.5. Integració amb Altres Serveis**

## **2.6. Preguntes Freqüents**



### **3. Exemples**



## **4. Annexos**