

Intégration PHP / JasperReports

par Charly Clairmont ([Site perso](#))

Date de publication : 01/06/2007

Dernière mise à jour : 24/08/2007

Ce document a pour but d'expliquer comment mettre en oeuvre l'appel de rapports réalisés au format JasperReports depuis une application PHP via le pont PHP / JAVA Bridge.

- I - Introduction
- II - Installation de PHP / Java Bridge
 - II-A - installation sous GNU Linux
 - II-B - nstallation sous Windows
 - II-C - vérification du support java de PHP
 - II-D - votre première intégration entre PHP et Java
- III - Appel des fichiers JasperReports depuis PHP
 - III-A - bâtir votre modèle de document avec iReport
 - III-B - écrire le script PHP qui appelle votre fichier JRXML
- IV - Conclusion
 - Pour aller plus loin
 - Auteur

I - Introduction

PHP a gagné d'années en années une image de langage professionnel, simple et performant. Comme le montre l'Association Française des Utilisateurs de PHP dans son livre blanc « PHP en Entreprise » (1), il y a un très grand nombre de sites internet de renom, de sociétés du CAC 40 qui développent en PHP. Ce qui renforce alors cette image de robustesse. Cette notoriété du langage est aussi visible compte tenu du nombre d'articles qui lui est consacré, et le nombre très important de recherches effectuées sur Google (2).

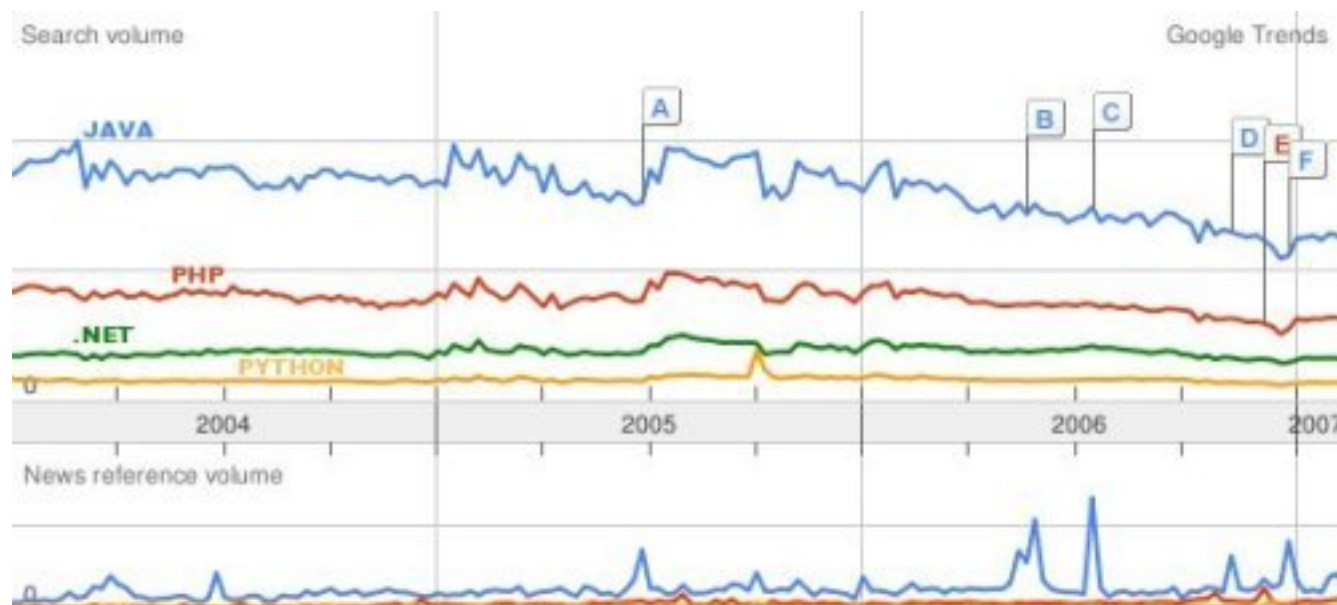


Illustration 1: Google Trends : JAVA, PHP, .NET, PYTHON

Il existe quelques bibliothèques de bon niveau pour la production de graphiques, de documents au format pdf. Toutefois, peu d'entre elles fournissent un environnement pour assister les concepteurs de rapports dans leur élaboration et leur publication. Agatha Reports (3), un projet Brésilien présente l'ambition d'apporter à PHP les outils de reporting qui lui manque. Cette solution bien que fonctionnelle ne supporte pas à ce jour PHP5 (4). De même, la nouvelle plate-forme de Zend en version 3.0.2 (5) affiche un support pour BIRT (6), un environnement de conception de rapport basé sous Eclipse et appuyé par Actuate.

Jasper Reports a offert au monde Java TM un moteur de grande qualité pour la génération de rapports rassemblant tableaux, graphiques, mêlant des sources de données très hétérogènes (rapport multi-sources). Cet outil en plus d'être très performant sait exporter vers divers formats tels que : pdf, html, xls, xml, cvs, txt, rtf et notamment un support pour Open Office depuis sa dernière version. Par ailleurs, iReport, un environnement de développement pour Jasper Reports procure une assistance de haut niveau dans la construction des autres rapports ou états. En plus, les deux projets suivent une évolution synchronisée. Ainsi, toutes les fonctionnalités ajoutées au moteur Jasper Reports sont prises en compte par iReport.

Nombreuses sont les interrogations postées sur les forums sur les outils de reporting en PHP, ou encore sur l'intégration entre PHP et des environnement de reporting tierce. Voilà une réponse : « PHP et Jasper Reports » ! Celle-ci nous semble élégante parce qu'elle apporte de la productivité, grâce à iReport qui est assez simple à prendre en main évitant alors d'apprendre l'API de la bibliothèque. Ensuite, Jasper Reports est un moteur aujourd'hui éprouvé : de nombreuses solutions libres ou propriétaires l'embarquent, des grandes sociétés comme Siemens, des administrations françaises en font un usage réguliers. Enfin, l'association PHP, Java, via PHP / Java Bridge (7) commence à gagner ses gallons. Même si certains pensent qu'il existe des marges d'amélioration (8). Ce projet rend possible l'instanciation d'objets java depuis des programmes PHP avec la syntaxe PHP.

Vous l'avez compris pour intégrer des rapports Jasper Reports à vos programmes PHP il est nécessaire d'installer PHP / Java Bridge.

II - Installation de PHP / Java Bridge

II-A - installation sous GNU Linux

L'installation sous GNU Linux est assez simple car les packages sont disponibles au format rpm. Il convient de télécharger le fichier **php-java-bridge-x.y.z-1-i386.rpm** et de l'installer :

fichier rpm pour l'installation du bridge entre php et java

```
rpm -i php-java-bridge-x.y.z-1-i386.rpm
```

où "x.y.z" correspond à la dernière version stable.

Bien sûr il est nécessaire d'avoir une machine virtuelle Java installée sur sa machine. Dans le fichier **php.ini** il faut au moins initier la variable `java.java`. Vous devez dans votre fichier **php.ini** (ou **php.d/java.ini**) ajouter les entrées :

configuration du fichier php.ini

```
java.java_home={Répertoire d'installation JAVA}  
java.java={Répertoire d'installation JAVA}/java
```

où {Répertoire d'installation JAVA} est à remplacer par le chemin vers votre installation du SDK.

Si vous souhaitez faire dialoguer vos applications PHP et un serveur J2EE installer le package `php-java-bridge-tomcat` :

fichier rpm pour exploiter une interface avec tomcat

```
rpm -i php-java-bridge-tomcat-x.y.z-1.i386.rpm
```

Pour télécharger les fichiers adéquats voir : http://sourceforge.net/project/showfiles.php?group_id=117793

II-B - installation sous Windows

Ici, il faut télécharger **php-java-bridge_y.x.z_j2ee.zip**, le décompresser dans un répertoire temporaire. Parmi les fichiers décompressés il y a une archive web **JavaBridge.war**. Décompressez le aussi et copiez les fichiers **JavaBridge.jar** et **java-x86-windows.dll** respectivement depuis `WEB-INF/lib` et `WEB-INF/cgi` dans le **répertoire de vos extensions php**. Enfin, il vous faut ajouter quelques entrées dans votre fichier `php.ini`.

configuration php.ini

```
Extension=php_java.dll
```

Votre fichier `php.ini` intègre alors la ligne précédente et est proche de ce qui suit :

configuration php.ini

```
; windows Extensions
; Note that ODBC support is built in, so no dll is needed for it.
; Note that many DLL files are located in the extensions/ (PHP 4) ext/ (PHP 5)
...
...
;extension-php_mbstring.dll
;extension-php_bz2.dll
...
...
extension-php_java.dll
...
...
```

Enfin configurez le connecteur Java pour PHP

configuration php.ini

```
;;;;;;;;;;;;;
; Module Settings ;
;;;;;;;;;;;;;

[java]
java.java_home = "c:\Program File\Java\jdk1.5.0_10\bin"
java.java = "c:\Program File\Java\jdk1.5.0_10\bin\javaw.exe"
java.class.path = "c:\php\ext\JavaBridge.jar"
java.library.path = "c:\php\ext"
;java.hosts = "127.0.0.1:8080"
;java.servlet = On
java.log_level = 2
```

Bien sûr les différentes variables sont à initialiser avec les valeurs relatives à votre environnement Java (le répertoire où est installé votre SDK), ou de votre serveur d'application J2EE.

pour plus d'informations : <http://www.dsl.uow.edu.au/~sk33/php5java.htm>

II-C - vérification du support java de PHP

Vous devez redémarrer votre serveur apache et vérifier que vous avez bien le support java de PHP.

Créez une page `phpinfo.php` et insérez y les lignes suivantes :

phpinfo()

```
<?php
phpinfo();
?>
```

Sauvez ce fichier, publiez le sur votre environnement web, et appelez la page depuis votre navigateur. Vous devrez observer le support java de php.

java

java support	Enabled
java bridge	4.0.8
java.java_home	/usr/local/java/jdk1.5.0_06
java.java	/usr/local/java/jdk1.5.0_06/bin/java
java.log_file	<stderr>
java.log_level	no value (use back-end's default level)
java.persistent_connections	On
java.security_policy	Off
java command	JAVA_HOME=/usr/local/java/jdk1.5.0_06 LD_LIBRARY_PATH=/usr/lib/php/modules: /usr/local/java/jdk1.5.0_06/bin/java -Djava.library.path=/usr/lib/php/modules -Djava.class.path=/usr/lib/php/modules/JavaBridge.jar -Djava.awt.headless=true -Dphp.java.bridge.base=/usr/lib/php/modules php.java.bridge.Standalone LOCAL:@java-bridge-9f5 1
java status	running
java server	@java-bridge-9f5

Illustration 4: module php / java bridge

II-D - votre première intégration entre PHP et Java

Créez un fichier java.php et collez les lignes suivantes :

appel de java depuis php

```

<?php
// créer une instance de la classe Java java.lang.System dans PHP
$system = new Java('java.lang.System');

// accéder aux propriétés
echo 'Java version=' . $system->getProperty('java.version') . ' <br />';
echo 'Java vendor=' . $system->getProperty('java.vendor') . ' <br />';
echo 'OS=' . $system->getProperty('os.name') . ' ' .
    $system->getProperty('os.version') . ' on ' .
    $system->getProperty('os.arch') . ' <br />';

// Exemple avec java.util.Date
$formater = new Java('java.text.SimpleDateFormat',
    "EEEE, MMMM dd, yyyy 'at' h:mm:ss a zzzz");

echo $formater->format(new Java('java.util.Date'));
?>

```

Enregistrez cette page web et appelez la depuis votre navigateur.



Illustration 5: première intégration entre php et java

III - Appel des fichiers JasperReports depuis PHP

III-A - bâtir votre modèle de document avec iReport

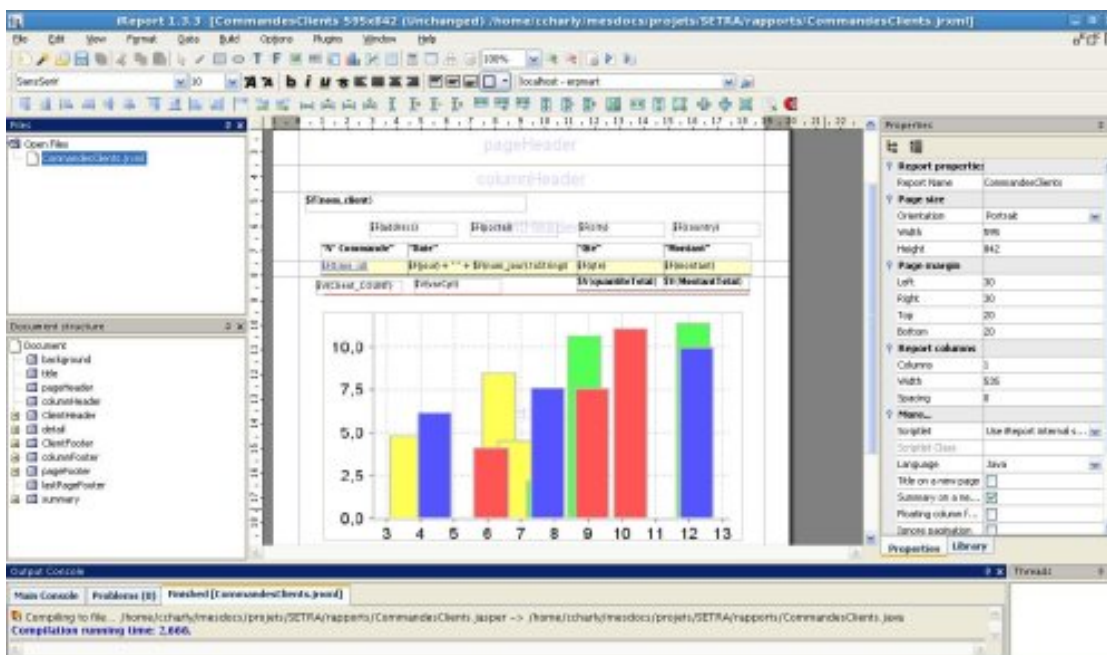


Illustration 6: iReport 1.3.3

Grâce à iReport vous allez pouvoir construire votre rapport :

- 1 définissez votre connexion à votre base de données ou autre source de données
- 2 créez un nouveau document
- 3 créez une requête avec l'assistant
- 4 Ajoutez les éléments que vous souhaitez à votre rapport
- 5 Enregistrez votre rapport
- 6 Exécutez le rapport.

III-B - écrire le script PHP qui appelle votre fichier JRXML

Dans l'exemple qui suit nous allons procéder à des instanciations des classes de l'API java de Jasper Reports en suivant les étapes généralement indiquées dans les exemples fournis par leur démo.

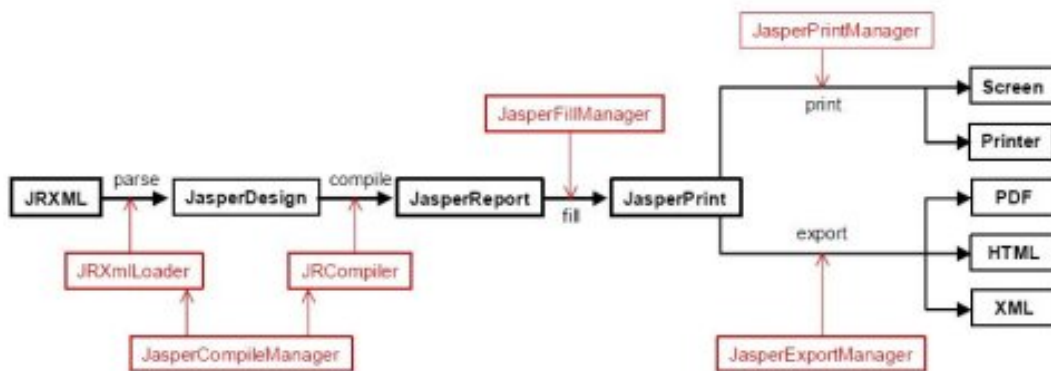


Illustration 7: API Jasper Reports : flux de génération

Créez un fichier que vous pouvez par exemple nommer `jasperreports.php` et ajoutez les lignes suivantes :

JasperReports depuis un script php

```

<?php
$reportsPath = "/home/ccharly/publichtml/utills/reports/";
$reportFileName = "CommandesClients1";
$jasperReportsLib = "/home/ccharly/publichtml/utills/jasperlib";

if(extension_loaded('java')) {

// lecture du répertoire où sont rangés les librairies utiles à JasperReports
$handle = @opendir($jasperReportsLib);

// ajout de tous les fichier jar au chemin de classe (Class Path)
while(($new_item = readdir($handle))!==false) {

    $java_library_path .= 'file:'. $jasperReportsLib . '/' . $new_item . ';';
}

try {
// chargement des librairies au classpath
java_require($java_library_path);

// création de la connexion JDBC
$conn = new Java("org.altic.jasperReports.JdbcConnection");
// driver
$conn->setDriver("com.mysql.jdbc.Driver");
// url de connexion
$conn->setConnectionString("jdbc:mysql://localhost/erpmart");
// utilisateur
$conn->setUser("root");
// mot de passe
$conn->setPassword(null);

// Compilation du fichier JRXML en fichier Jasper
$sJcm = new JavaClass("net.sf.jasperreports.engine.JasperCompileManager");
$report = $sJcm->compileReport($reportsPath . $reportFileName . ".jrxml");

// Remplir le modèle avec les données
$sJfm = new JavaClass("net.sf.jasperreports.engine.JasperFillManager");
$print = $sJfm->fillReport(
    $report,
    new Java("java.util.HashMap"),
    $conn->getConnection()
);
}
  
```

JasperReports depuis un script php

```
// Export du fichier au format pdf
$$Jem = new JavaClass("net.sf.jasperreports.engine.JasperExportManager");
$$Jem->exportReportToPdfFile($print, $reportsPath . $reportFileName.".pdf");

if (file_exists($reportsPath . $reportFileName.".pdf")){
    header('Content-disposition: attachment; filename="'. $reportFileName.'.pdf');
    header('Content-Type: application/pdf');
    header('Content-Transfer-Encoding: binary');
    header('Content-Length: ' . @filesize($reportsPath . $reportFileName.".pdf"));
    header('Pragma: no-cache');
    header('Cache-Control: must-revalidate, post-check=0, pre-check=0');
    header('Expires: 0');
    set_time_limit(0);
    @readfile($reportsPath . $reportFileName.".pdf") or die("problem occurs.");
}

} catch (JavaException $ex) {
    $trace = new Java("java.io.ByteArrayOutputStream");
    $ex->printStackTrace(new Java("java.io.PrintStream", $trace));
    print "java stack trace: $trace\n";
}
}

?>
```

Vous devez adapter le fichier en modifiant les variables suivantes :

\$reportsPath : chemin où sont rangés vos rapports au format jrxml

\$reportFileName : nom du fichier à compiler et exporter en pdf (remarque : ici seule la racine du nom du fichier est nécessaire)

\$jasperReportsLib : répertoire des bibliothèques nécessaires à l'utilisation de JasperReports. Ce répertoire contient par exemple les mêmes bibliothèques que ceux contenu dans le répertoire lib d'iReport.

La commande `java_require` charge dans le classpath toutes ces bibliothèques. Veillez à ce que le répertoire désigné par `$jasperReportsLib` soit accessible à l'utilisateur qui fait tourner apache ainsi que les fichiers qu'il contient.

Ajoutez le fichier `alticJasper.jar` il contient un petit utilitaire pour créer une connexion JDBC.

classe de la connexion JDBC

```
package org.altic.jasperReports;

import java.io.FileInputStream;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

/**
 * @author ccharly
 *
 */
public class JdbcConnection {
```

classe de la connexion JDBC

```
private String driver; // "oracle.jdbc.driver.OracleDriver";
private String connectString; // "jdbc:oracle:thin:@YOUR_ORACLE_HOST:1521:YOUR_SID";
private String user; // "YOUR_ORACLE_USER_NAME";
private String password; // "YOUR_ORACLE_PASSWORD";

public JdbcConnection(){
    //loadPropertiesConnection();
}

public JdbcConnection(String driver, String connectString, String user, String password){
    this.driver = driver;
    this.connectString = connectString;
    this.user = user;
    this.password = password;
}

    public void loadPropertiesConnection(){

this.driver = "";
this.connectString = "";
this.user = "";
this.password = "";

try {
    Properties props = new Properties();
    InputStream resourceAsStream = getClass().getResourceAsStream("/connection.properties");
    props.load(resourceAsStream);
    this.setDriver(props.getProperty("driver"));
    this.setConnectString(props.getProperty("connectString"));
    this.setUser(props.getProperty("user"));
    this.setPassword(props.getProperty("password"));
} catch (Exception e){
    e.printStackTrace();
}
}

    public Connection getConnection()
{
try {
    //Change these settings according to your local configuration
    Class.forName(this.getDriver());
    Connection conn = DriverManager.getConnection(
        this.getConnectString(),
        this.getUser(),
        this.getPassword());
    return conn;
} catch (ClassNotFoundException e){
    e.printStackTrace();
} catch (SQLException e){
    e.printStackTrace();
}
return null;
}

    public String toString(){
return " Driver : " + this.getDriver() + " | " +
    " ConnectString : " + this.getConnectString() + " | " +
    " User : " + this.getUser() + " | " +
    " Password : " + this.getPassword();
}

public String getConnectString() {
return connectString;
}
```

classe de la connexion JDBC

```
}  
  
public void setConnectionString(String connectionString) {  
    this.connectString = connectionString;  
}  
  
public String getDriver() {  
    return driver;  
}  
  
public void setDriver(String driver) {  
    this.driver = driver;  
}  
  
public String getPassword() {  
    return password;  
}  
  
public void setPassword(String password) {  
    this.password = password;  
}  
  
public String getUser() {  
    return user;  
}  
  
public void setUser(String user) {  
    this.user = user;  
}  
  
}
```

Vous devez donc modifier les paramètres de connexions vers votre base de données. Pensez aussi à ajouter la librairie du connecteur JDBC vers votre base de données.

Indiquez le driver JDBC de votre base de données, par exemple pour MySql

configuration d'une connexion JDBC depuis PHP - 2

```
$Conn->setDriver("com.mysql.jdbc.Driver");
```

Indiquez l'url de connexion

configuration d'une connexion JDBC depuis PHP - 2

```
$Conn->setConnectionString("jdbc:mysql://{Nom du S}:{port}/{Nom de la Base de données}");
```

Indiquez le nom de l'utilisateur

configuration d'une connexion JDBC depuis PHP - 3

```
$Conn->setUser({utilisateur});
```

Indiquez le mot de passe

configuration d'une connexion JDBC depuis PHP - 4

```
$Conn->setPassword({Mot de passe});
```

Une fois ces paramétrages terminés, vous êtes en mesure de générer vos rapports JasperReports depuis PHP. La compilation n'est sans doute pas utile si vous publiez directement les fichiers « .jasper » que vous pouvez récupérer depuis iReport.

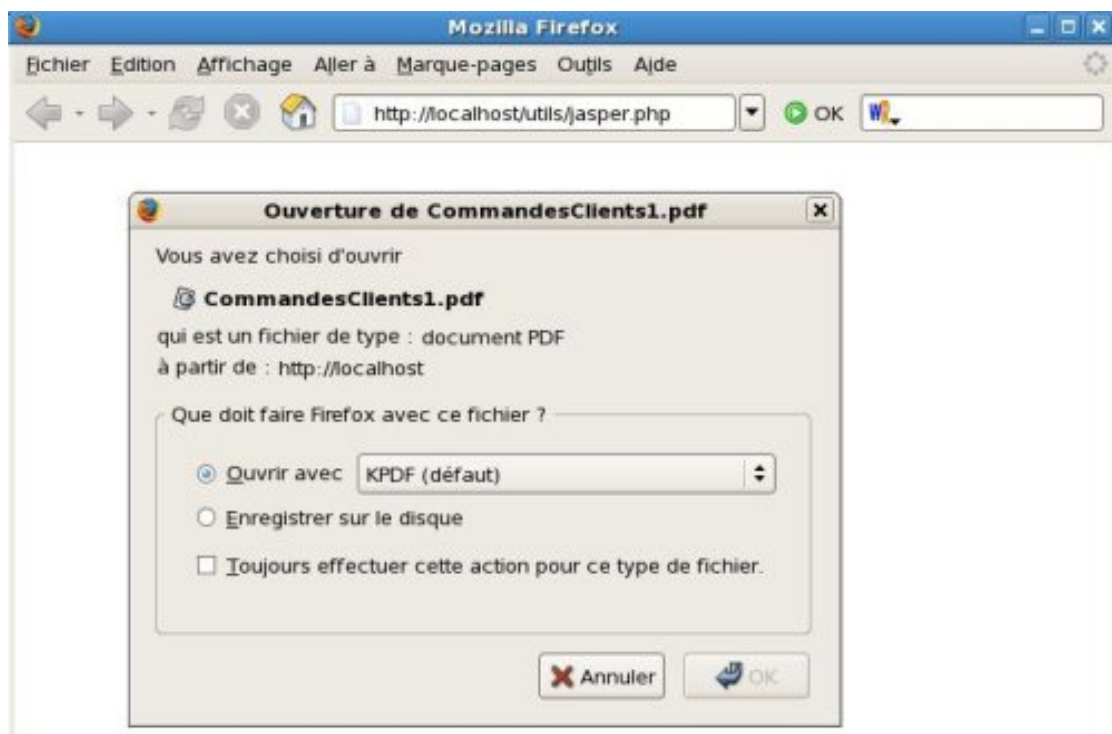


Illustration 8: téléchargement du fichier généré

IV - Conclusion

Ce petit exemple montre comment il est ainsi possible d'exploiter JasperReports via une application PHP. Cette intégration apporte à PHP un outil de reporting performant et particulièrement productif grâce à iReport. Les concepteurs de rapports se retrouve dans un environnement convivial et productif.

Pour augmenter à ce titre la productivité il serait intéressant d'encapsuler JasperReports afin d'en masquer la complexité pour ne pas rebuter les utilisateurs de PHP déjà habitués à une syntaxe relativement simple. Prenons par exemple la connexion vers la base de données il faudrait une ligne de code qui serait :

simplifier connexion JDBC - 1

```
$conn->getMySQLConnection({nom du Serveur}, {Port}, {Base de données}, {Utilisateur}, {Mot de passe});
```

ou bien

simplifier connexion JDBC - 2

```
$conn->getConnection({Type de Base},{nom du Serveur}, {Port}, {Base de données}, {Utilisateur}, {Mot de passe});
```

où {Type de Base} vaudrait : ORACLE, MYSQL, POSTGRESQL

L'encapsulation éviterait aussi aux développeurs PHP d'avoir à manipuler les appels directs aux classes java, ce qui en simplifierait la lecture du code.

Le projet Dynamic Jasper offre une simplification de l'API JasperReports. Il serait intéressant de se baser sur ce projet pour développer une librairie de scripts PHP pour JasperReports.

Pour aller plus loin

PHP / Java Bridge  <http://php-java-bridge.sourceforge.net/pjb>

Jasper Reports and PHP C'est un très bon article qui apporte déjà une certaine abstraction de la complexité de Jasper Report en PHP. Il faut je pense aller encore plus loin. Mais c'est un très bon début.

 <http://www.rjohnson.id.au/wordpress/2007/02/04/jasper-reports-and-php/>

Projet Jasper Reports  <http://jasperforge.org/sf/projects/jasperreports>

Projet iReport  <http://jasperforge.org/sf/projects/ireport>

PHP et BEA Weblogic  <http://dev2dev.bea.com/pub/a/2007/02/php-java-bridge.html>


Les cours sur les générateurs d'états - sur Developpez.com  <http://etats.developpez.com>

Intégration de FOP et PHP / Java Bridge  <http://wiki.apache.org/xmlgraphics-fop/HowTo/PHPJavaBridge>

Dynamic Jasper  <http://dynamicjasper.sourceforge.net/>

Auteur

Charly Clairmont

Après un passage en startup de la "nouvelle économie", ensuite en SSII, il fonde  **Altic** avec Marc Sallieres.

ALTIC, société spécialisée dans l'informatique de gestion en logiciel libre. Depuis 3 ans, elle a aussi développé une forte expertise dans les solutions d'informatique décisionnelle en logiciel libre. Les différents projets menés dans le monde de l'entreprise mais aussi des grandes administrations ont apporté aux consultant d'ALTIC une expérience pragmatique de les solutions de reporting open source du marché (Jasper, Birt, Agata, ...).

ALTIC accompagne ses clients directement avec le support de ses partenaires qui sont pour la plupart les éditeurs open source du marché de l'informatique décisionnelle : Talend, Engineering, eXo Platform, JasperSoft...

ALTIC propose d'ailleurs des formations autour de ces solutions JasperReports / iReport Niveau 1 et Niveau 2 (Monitorat), Birt Niveau 1, mais aussi Mondrian Niveau 1 et enfin la plate-forme décisionnelle et collaborative SpagoBI.

1 : Livre Blanc "PHP en entreprise" -

<http://php.developpez.com/etudes/livre-blanc-php-entreprise>

2 : <http://google.fr/trends?q=java,php,python,.net&ctab=0&geo=all&date=all&sort=0>

3 : <http://www.agata.org.br/>

4 :

http://groups.google.fr/group/comp.lang.php/browse_thread/thread/b9e178b19c135cca/ba59aa035b6e4b8f

5 : http://www.zend.com/fr/products/zend_platform/what_s_new

6 : <http://www.eclipse.org/birt/phoenix/> - <http://birtworld.blogspot.com>

7 : <http://php-java-bridge.sourceforge.net/pjb/index.php>

8 : <http://www.clever-age.com/veille/blog/retour-d-experiences-sur-php-java-bridge.html>