ACube

Normes de développement Jasper

Version 1.1 du 22/02/2010

Etat : Validé

SUIVI DES MODIFICATIONS

Version	Rédaction	Description	Vérification	Date
1.0	A. Lesuffleur	Initialisation		27/08/09

DOCUMENTS DE REFERENCE

Version	Titre

SOMMAIRE

FIGURES

DOCUMENTS DE REFERENCE

Version	Titre

1 INTRODUCTION

Ce document présente les normes et méthodologies à appliquer dans les projets lors de l'utilisation de Jasper Report.

2 PRESENTATION DE JASPERREPORT

2.1 PRESENTATION GENERAL

Jasperreport est un Framework java open source permettant de générer des rapports d'états à partir d'une source de données. Les rapports peuvent être générés dans les formats suivant : xml, pdf, txt, xls, rtf, html et csv.

Jasperreport est le remplaçant de FOP pour LISE V3, il est à utiliser dans le même cadre que FOP.

Pour réaliser une génération d'un rapport, il faut :

- Réaliser le document jrxml qui est la description du rapport au format XML (JRXML),
- Définir la source de données : base de données, fichier XML, objets JAVABeans …
- Compiler le rapport jrxml pour obtenir le rapport compilé « .jasper »

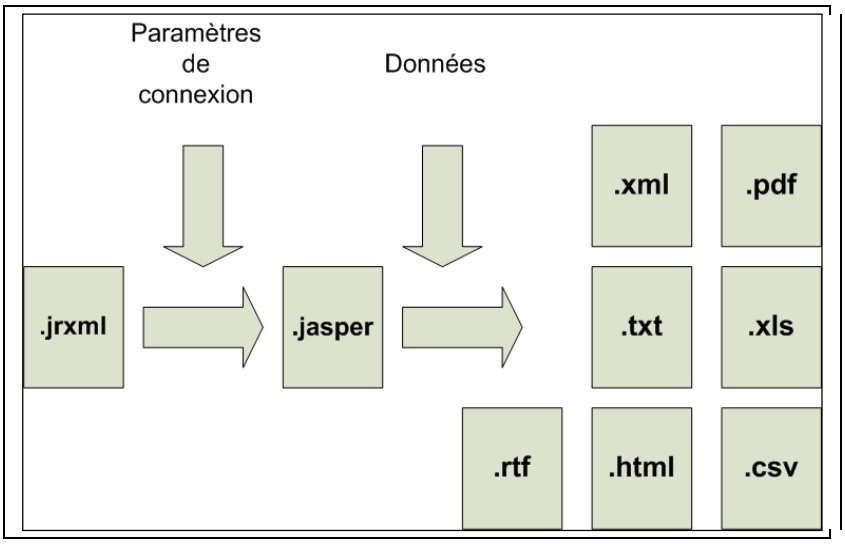

Figure 1 : Cycle de vie d'un rapport Jasper

3 NORMES JASPER

3.1 NOMMAGE ET LOCALISATION

Les sources des rapports (jrxml) sont à placer dans un package jasper au même niveau que le répertoire src contenant les sources java.

Les règles de nommages des rapports sont similaires aux règles java et les rapports doivent respectés le découpage fonctionnel java.

Par contre, en raison d'un dysfonctionnement jasper, les rapports compilés (fichier jasper) doivent se trouver dans un sous répertoire des sources java appelé src/acube/projet/export/jasper/report/<DOMAINE FONCTIONEL/…)

3.2 RAPPORTS, PERFORMANCE ET MEMOIRE

Les rapports de plus de 5 pages doivent être identifiés en amont de la réalisation et signalés aux équipes transverses afin de prévenir les cas de non performance ou de consommation mémoire excessive.

Les rapports volumineux avec graphiques doivent être employés avec précaution : ils sont potentiellement consommateur en CPU et en RAM.

3.3 LIVRAISON DES RAPPORTS

Le code source des rapports doit impérativement être livré avec le code source de l'application (fichier jrxml)

Les rapports doivent être compilées lors de la construction du livrable (jrxml -> jasper).

3.4 SCRIPLETS

L'utilisation et la création de scriplet pour un rapport doivent être soumises à l'autorisation préalable des équipes transverses (comme les taglibs jsp).

3.5 ELEMENTS PROHIBES

3.5.1 FRAGMENT DE CODE JAVA

L'utilisation de code pseudo-java ou java est possible dans jasper, notamment pour les structures conditionnelles.

L'utilisation de code pseudo-java ou java interdite dans les rapports, les règles de gestion et le formatage doivent être fait dans des V.O. ou classes utilitaires.

Pour illustration, voici un exemple de code java « à la jasper » :

```
( $F{matrRegNaEpx} != null
        ?
                ($F{matrPaysNaEpx} != null
                        ? $F{matrRegNaEpx} + "( " + $F{matrPaysNaEpx} + " )"
                        : $F{matrRegNaEpx})
        :  ( $F{matrArrondNaEpx} != null
                        ? ( $F{matrDeptNaEpx} != null
                                ? $F{matrArrondNaEpx}.toString() + "( " +
$F{matrDeptNaEpx}.toString() + " )"
```

```
                                : $F{matrArrondNaEpx})
                        : "")
)
```

3.5.2ACCES AUX DONNEES

Aucun accès à une source de données, SQL ou autre, ne doit avoir lieu dans un rapport jasper.

4 METHODOLOGIE

4.1 ARCHITECTURE PROJET

Classe à implémenter pour le maquettage du rapport:

- Un Javabean de présentation permet d'alimenter le rapport
- Une interface du service pour récupérer les Javabeans nécessaires à un rapport.
- Une classe de test qui implémente l'interface du service avec les méthodes statiques pour y accéder à partir d'Ireport

Classe à implémenter pour le rapport final:

- Un Javabean de présentation permet d'alimenter le rapport
- Une interface du service pour récupérer les Javabeans nécessaires à un rapport.
- Une classe de test qui implémente l'interface du service avec les méthodes statiques pour y accéder à partir d'Ireport
- Une classe service pour instancier les JavaBeans à partir des BO ou à partir de la base de données
- Une classe Action struts pour rendre disponible le fichier généré

4.2 LES OBJETS JAVABEANS

- Les formatages doivent être fait dans cet objet et non dans le rapport jasper.
- Il faut au minimum un JavaBean par sous-rapport
-

4.3 LES INTERFACES DE SERVICES

- Il faut au minimum une méthode par JavaBean qui retourne une liste de ce type dans le but de pouvoir générer unitairement les sous-rapports.
-

4.4 LES CLASSES DE TESTS

- Elle implémente l'interface service
-