

Référentiel Coordonnées JAVA

Architecture et solutions techniques

Thomas Duchatelle (thomas.duchatelle-ext@yrnet.com)

Yves Rocher

February 1, 2013

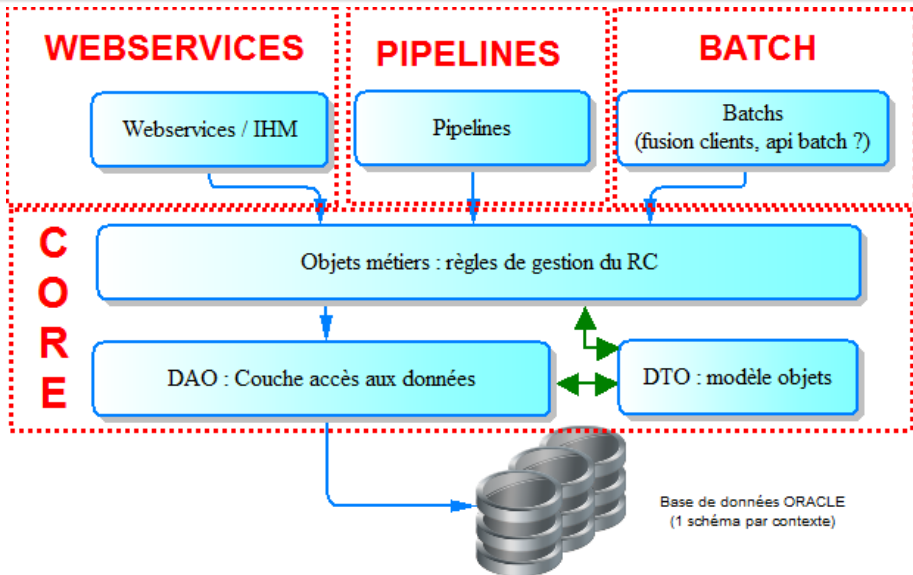
- 1 Architecture générale
- 2 Coeur applicatif
- 3 Webservices et IHM
- 4 Pipelines

Sommaire

- 1 Architecture générale
 - Couches applicatives et modules
 - Outils utilisés
 - Compilation et déploiement
- 2 Coeur applicatif
- 3 Webservices et IHM
- 4 Pipelines

Référentiel Coordonnées

1 application, 4 modules



Modules du référentiel coordonnées

Core Coeur de l'application (jar) :

Modules du référentiel coordonnées

Core Coeur de l'application (jar) :

- Modèle objets : représentation *client / coordonnées*

Modules du référentiel coordonnées

Core Coeur de l'application (jar) :

- Modèle objets : représentation *client / coordonnées*
- couche d'accès au données : structure de la base, gestion des schémas

Modules du référentiel coordonnées

Core Coeur de l'application (jar) :

- Modèle objets : représentation *client / coordonnées*
- couche d'accès au données : structure de la base, gestion des schémas
- règles de gestion : calcul des consentements, règles de mises à jour, dé-duplication

Modules du référentiel coordonnées

Core Coeur de l'application (jar) :

- Modèle objets : représentation *client / coordonnées*
- couche d'accès au données : structure de la base, gestion des schémas
- règles de gestion : calcul des consentements, règles de mises à jour, dé-duplication

Webservices Application Java EE (war) exposant des webservices

Modules du référentiel coordonnées

Core Coeur de l'application (jar) :

- Modèle objets : représentation *client / coordonnées*
- couche d'accès au données : structure de la base, gestion des schémas
- règles de gestion : calcul des consentements, règles de mises à jour, dé-duplication

Webservices Application Java EE (war) exposant des webservices

- correspondance *modèle objet interne* \leftrightarrow *contrat WSDL*

Modules du référentiel coordonnées

Core Coeur de l'application (jar) :

- Modèle objets : représentation *client / coordonnées*
- couche d'accès au données : structure de la base, gestion des schémas
- règles de gestion : calcul des consentements, règles de mises à jour, dé-duplication

Webservices Application Java EE (war) exposant des webservices

- correspondance *modèle objet interne* \leftrightarrow *contrat WSDL*
- Interface WEB : suivi des pipelines

Modules du référentiel coordonnées

Core Cœur de l'application (jar) :

- Modèle objets : représentation *client / coordonnées*
- couche d'accès au données : structure de la base, gestion des schémas
- règles de gestion : calcul des consentements, règles de mises à jour, dé-duplication

Webservices Application Java EE (war) exposant des webservices

- correspondance *modèle objet interne* ↔ *contrat WSDL*
- Interface WEB : suivi des pipelines

Pipelines Intégration de données par fichiers CSV

Modules du référentiel coordonnées

Core Coeur de l'application (jar) :

- Modèle objets : représentation *client* / *coordonnées*
- couche d'accès au données : structure de la base, gestion des schémas
- règles de gestion : calcul des consentements, règles de mises à jour, dé-duplication

Webservices Application Java EE (war) exposant des webservices

- correspondance *modèle objet interne* ↔ *contrat WSDL*
- Interface WEB : suivi des pipelines

Pipelines Intégration de données par fichiers CSV

- Démarrage/Arrêt d'un *démon*

Modules du référentiel coordonnées

Core Coeur de l'application (jar) :

- Modèle objets : représentation *client / coordonnées*
- couche d'accès au données : structure de la base, gestion des schémas
- règles de gestion : calcul des consentements, règles de mises à jour, dé-duplication

Webservices Application Java EE (war) exposant des webservices

- correspondance *modèle objet interne* \leftrightarrow *contrat WSDL*
- Interface WEB : suivi des pipelines

Pipelines Intégration de données par fichiers CSV

- Démarrage/Arrêt d'un *démon*
- Détection de l'arrivée de fichiers

Modules du référentiel coordonnées

Core Coeur de l'application (jar) :

- Modèle objets : représentation *client / coordonnées*
- couche d'accès au données : structure de la base, gestion des schémas
- règles de gestion : calcul des consentements, règles de mises à jour, dé-duplication

Webservices Application Java EE (war) exposant des webservices

- correspondance *modèle objet interne* ↔ *contrat WSDL*
- Interface WEB : suivi des pipelines

Pipelines Intégration de données par fichiers CSV

- Démarrage/Arrêt d'un *démon*
- Détection de l'arrivée de fichiers
- EAI et lecture de fichier

Modules du référentiel coordonnées

Core Coeur de l'application (jar) :

- Modèle objets : représentation *client / coordonnées*
- couche d'accès au données : structure de la base, gestion des schémas
- règles de gestion : calcul des consentements, règles de mises à jour, dé-duplication

Webservices Application Java EE (war) exposant des webservices

- correspondance *modèle objet interne* ↔ *contrat WSDL*
- Interface WEB : suivi des pipelines

Pipelines Intégration de données par fichiers CSV

- Démarrage/Arrêt d'un *démon*
- Détection de l'arrivée de fichiers
- EAI et lecture de fichier
- Parallélisation des traitements

Modules du référentiel coordonnées

Core Coeur de l'application (jar) :

- Modèle objets : représentation *client* / *coordonnées*
- couche d'accès au données : structure de la base, gestion des schémas
- règles de gestion : calcul des consentements, règles de mises à jour, dé-duplication

Webservices Application Java EE (war) exposant des webservices

- correspondance *modèle objet interne* ↔ *contrat WSDL*
- Interface WEB : suivi des pipelines

Pipelines Intégration de données par fichiers CSV

- Démarrage/Arrêt d'un *démon*
- Détection de l'arrivée de fichiers
- EAI et lecture de fichier
- Parallélisation des traitements

Batch Exports, traitements types fusion, statistiques

Organisation du code

Maven

Le référentiel coordonnées utilise *Maven* pour :

- la gestion des dépendances
- la compilation
- le déploiement des pipelines et batch
- configuration du poste de travail : *Eclipse*
(1 projet maven = 1 projet Eclipse)
- tests en local des webservices et IHM

Organisation du code

Maven

Le référentiel coordonnées utilise *Maven* pour :

- la gestion des dépendances
- la compilation
- le déploiement des pipelines et batch
- configuration du poste de travail : *Eclipse*
(1 projet maven = 1 projet Eclipse)
- tests en local des webservices et IHM

Projets Maven

Chaque modules du RC est représenté par un sous-projet Maven.

Projets et sous-projets *Maven*

Le référentiel coordonnées est répartie sur 6 projets *Maven* :

- **addressrepository** : *version, déclarations des dépendances*

Projets et sous-projets *Maven*

Le référentiel coordonnées est répartie sur 6 projets *Maven* :

- **addressrepository** : *version, déclarations des dépendances*
 - **address-core** : *coeur de l'application*
package : `net.yvesrocher.services.address.core`

Projets et sous-projets *Maven*

Le référentiel coordonnées est répartie sur 6 projets *Maven* :

- **addressrepository** : *version, déclarations des dépendances*
 - **address-core** : *coeur de l'application*
package : `net.yvesrocher.services.address.core`
 - **address-pipelines** : *Jar exécutable (démon)*
package : `net.yvesrocher.services.address.pipelines`

Projets et sous-projets *Maven*

Le référentiel coordonnées est répartie sur 6 projets *Maven* :

- **addressrepository** : *version, déclarations des dépendances*
 - **address-core** : *coeur de l'application*
package : `net.yvesrocher.services.address.core`
 - **address-pipelines** : *Jar exécutable (démon)*
package : `net.yvesrocher.services.address.pipelines`
 - **address-webservices** : *War du webservices*
package : `net.yvesrocher.services.address.webservices`

Projets et sous-projets *Maven*

Le référentiel coordonnées est répartie sur 6 projets *Maven* :

- **addressrepository** : *version, déclarations des dépendances*
 - **address-core** : *coeur de l'application*
package : `net.yvesrocher.services.address.core`
 - **address-pipelines** : *Jar exécutable (démon)*
package : `net.yvesrocher.services.address.pipelines`
 - **address-webservices** : *War du webservices*
package : `net.yvesrocher.services.address.webservices`
 - **address-ear** : *Package le war en un EAR*

Projets et sous-projets *Maven*

Le référentiel coordonnées est répartie sur 6 projets *Maven* :

- **addressrepository** : *version, déclarations des dépendances*
 - **address-core** : *coeur de l'application*
package : `net.yvesrocher.services.address.core`
 - **address-pipelines** : *Jar exécutable (démon)*
package : `net.yvesrocher.services.address.pipelines`
 - **address-webservices** : *War du webservices*
package : `net.yvesrocher.services.address.webservices`
 - **address-ear** : *Package le war en un EAR*
 - **address-batch** : *Batches utilisant le coeur V3, jar exécutable*
package : `net.yvesrocher.services.address.batch`

Outils et frameworks utilisés

Spring Inversion de contrôle, gestion des contextes, transactions
BDD

Hibernate *Mapping Relationnel Objet*

Sommaire

- 1 Architecture générale
- 2 Coeur applicatif
 - Couche d'accès au données
 - Plan et principales briques logicielles
 - API de tests unitaires
- 3 Webservices et IHM
- 4 Pipelines

Sommaire

- 1 Architecture générale
- 2 Coeur applicatif
- 3 Webservices et IHM**
- 4 Pipelines

Sommaire

- 1 Architecture générale
- 2 Coeur applicatif
- 3 Webservices et IHM
- 4 Pipelines**