



# REUSABLE BUSINESS OBJECTS (RBO)

## 1. Description

Dans le cadre de la gestion des processus métier des services publics fédéraux et des institutions de la sécurité sociale, de nombreux services informatiques métier et applicatifs font appel à des données communes.

Pour garder la cohérence des données et mieux gérer l'interopérabilité entre les services métier, applicatifs et techniques, une librairie de composants réutilisables à faible granularité a été élaborée. Cette librairie d'objets porte le nom de « Reusable Business Objects » (RBO).

L'objectif des RBO est de gérer les opérations de formatage, de validation du format et du contenu des données par rapport à toute une série de règles prédéfinies. Ces RBO présentent un intérêt important pour améliorer la productivité des équipes de développement et décharger les applications de toutes ces opérations.

Actuellement, il existe trois catégories de RBO :

- La première catégorie comprend des RBO qui traitent les données relatives à la sécurité sociale. Il s'agit notamment des données telles que le NISS, le NOSS ou le numéro d'entreprise KBO ;
- La deuxième catégorie des RBO permet de manipuler les données de type date ;
- La troisième catégorie comprend des RBO divers et variés permettant notamment de traiter d'autres types de données tels que le numéro de TVA, le numéro du compte bancaire ou le numéro de ticket.

Les tableaux suivants présentent la liste des RBO actuellement disponibles dans ces trois différentes catégories. Le nom de chaque objet RBO représente la donnée (métier ou technique) qu'il peut traiter. A titre d'exemple, le RBO intitulé « Numéro de compte bancaire » permet de valider et de formater toutes les données représentant un numéro de compte bancaire.

#### RBO « Sécurité Sociale »

- Catégorie employeur
- Catégorie ONSS / APL d'un employeur
- Catégorie ONSS d'un employeur
- Code d'activité principal
- Code d'importance d'un employeur
- Code travailleur
- Degré d'incapacité
- Numéro d'identification INAMI d'un hôpital
- Numéro d'identification INAMI d'un praticien
- Numéro d'occupation
- Numéro de carte d'identité sociale (NISS)
- Numéro de commission paritaire
- Numéro de matricule ONSS
- Numéro de matricule ONSS / APL
- Numéro de matricule « M7 »
- Numéro de secrétariat social
- Numéro Dimona
- Numéro unique d'entreprise
- Régime administratif
- Situation administrative
- Type de cotisation
- Numéro d'établissement

#### RBO de type Date

- Année/Trimestre
- Date
- Date de naissance
- Heure

#### RBO « Divers »

- Code pays
- E-mail
- Numéro du compte bancaire (Belgique, France, Luxembourg, Pays-Bas, Irlande et Royaume-Uni)
- Numéro de ticket
- Numéro de TVA (relatif aux 25 membres de l'UE)
- Plaque adresse (rue, numéro, code postal, localité, pays)
- Sexe

## 2. Disponibilité

En production.

## 3. Conditions d'utilisation du composant réutilisable

Le composant est utilisable par d'autres services publics belges : oui.

Les modalités d'utilisation doivent être convenues à l'occasion d'un projet d'installation. Pour ce faire, les services publics intéressés sont priés de s'adresser à la personne de contact.

## 4. Procédures de demande d'utilisation

Adressez-vous à la personne de contact. Nous prenons l'initiative pour répondre à vos questions et/ou proposer une réunion d'étude.

## 5. Support (en mode service)

Les modalités de support pour les services en production seront communiquées au cours de la mise en production.

## 6. Informations fonctionnelles

Vous trouverez ci-dessous un aperçu des fonctionnalités qu'offre ce composant.

### 6.1. Description input/output du composant réutilisable

Le tableau suivant présente les données nécessaires pour l'invocation des objets RBO (input) et le résultat fourni (output).

Input	Output
La valeur (formatée ou non) d'une des données mentionnées.	La validité ou non de la valeur de la donnée réceptionnée.
Le format de la donnée. Cette deuxième information est optionnelle.	La validité ou non du format de la donnée.
	La validité ou non du chiffre de contrôle.
	La donnée formatée dans un autre format que celui de l'input transmis.
	La valeur de la donnée non formatée.

### 6.2. Description des possibilités d'intégration / d'interfaçage du composant réutilisable

Les RBO sont des objets embarqués dans les applications sous forme de librairie Java. Cette librairie doit être copiée et intégrée dans l'application (Web ou standalone) pour être utilisée. Il ne s'agit donc pas d'un composant distribué pouvant être invoqué à distance via les Web Services, les méthodes RMI (Remote Method Invocation) ou autres mécanismes d'invocation à distance.

La description du contrat de chaque RBO est définie à l'aide d'une interface Java. Nous entendons par contrat la liste des opérations (ou méthodes Java) disponibles et leur signature (liste des paramètres et leur type, et le résultat fourni et son type).

Enfin, soulignons qu'il existe des adaptateurs permettant aux frameworks Hibernate<sup>1</sup> et GDF<sup>2</sup> de manipuler les objets RBO de façon transparente.

<sup>1</sup> <http://www.hibernate.org>.

<sup>2</sup> GDF : Generic Data Framework : composant développé par Smals.

### **6.3. Description des éléments de volumétrie qui ont été pris en compte lors du développement de ce composant**

Comme les objets RBO sont embarqués dans l'application cliente, il n'y a pas de limite en termes de volumétrie. Ces objets peuvent être instanciés un nombre illimité de fois. La capacité de la mémoire RAM est la seule contrainte dont il faut tenir compte.

### **6.4. Description des autres éléments pertinents**

Les règles de formatage, de validation et de transformation sont codées en dur (hardcodés) dans les classes Java de façon prédéfinie. Ces règles ne changent généralement pas, raison pour laquelle elles n'ont pas été externalisées du code source. Cela étant, il est tout à fait possible que l'utilisateur indique ses propres règles de formatage.

## **7. Informations techniques**

Les objets RBO ont été développés sur la base des spécifications Java (version 1.4.). Ils sont disponibles sous forme de fichier Jar et peuvent être déployés sur toutes les plates-formes compatibles avec Java et J2EE, dont notamment les serveurs d'applications BEA Weblogic et IBM WebSphere.

La description des différentes méthodes fournies par les RBO et des données d'input et d'output est disponible dans la documentation HTML générée par l'outil Javadoc. Cette documentation peut être envoyée en même temps que les objets RBO.