

Seed

WebService

## Documentation du Webservice Seed

Version 1.2 du 10/03/2015

## SOMMAIRE

1 INTRODUCTION.....	5
1.1 OBJET DU DOCUMENT.....	5
1.2 TERMINOLOGIE ET ABRÉVIATIONS.....	5
1.2.1 Sigles et abréviations.....	5
1.2.1.1 Fonctionnels.....	5
1.2.1.2 Techniques.....	5
2 COMMUNICATION AVEC LE WEB SERVICE.....	6
2.1 ARCHITECTURE GLOBALE.....	6
2.2 MÉTHODES DISPONIBLES.....	6
2.2.1 Méthode GetInformation.....	6
2.2.1.1 Objectif de la méthode.....	6
2.2.1.2 Paramètre de la méthode.....	6
2.2.1.3 Réponse de la méthode.....	6
2.2.1.4 Description du contenu du fichier XML de réponse.....	7
2.2.1.4.1 Erreur.....	7
2.2.1.4.2 Alerte.....	7
2.2.1.4.3 Opérateur.....	7
2.2.1.4.4 Entrepôt.....	8
2.2.1.4.5 Éléments commun aux opérateurs et entrepôts.....	8
3 FONCTIONNEMENT DU WEB SERVICE.....	10
3.1 WEBSERVICE FO.....	10
3.1.1 Méthodes exposées.....	10
3.1.1.1 getInformation.....	10
3.1.1.2 checkDataBase.....	10
3.2 WEBSERVICE BO.....	10
3.2.1 Méthodes appelées.....	10
3.2.1.1 getOperateur.....	11
3.2.1.2 getSurveillance.....	11
3.2.2 Alertes.....	11
3.2.2.1 Validation des données.....	11
3.2.2.1.1 TraderAuthorisation.....	11
3.2.2.1.2 TaxWareHouse.....	11
3.2.2.2 Opérateur invalide.....	11
3.2.3 Liste des procédures stockées utilisées.....	11
3.2.3.1 SEED.....	12
3.2.3.2 ROSA.....	12
3.3 CLIENT.....	12
4 SERVEUR JBOSS.....	12
5 ANNEXE.....	13
5.1 VALIDATION DES DONNÉES.....	13
5.1.1 CheckNumTVA.....	13
5.1.2 CheckDate.....	13
5.1.3 CheckCountry.....	13
5.1.4 CheckOperatorType.....	13
5.1.5 CheckCodeBureau.....	13
5.1.6 CheckProduits.....	13
5.1.7 CheckEntrepots.....	13
5.1.8 CheckAdresses.....	13

*SEED\_WebService\_1.2.odt*

5.2 EXEMPLES D'APPEL.....	13
5.2.1 GetInformation.....	13
5.2.2 CheckDataBase.....	14
5.3 EXEMPLE XML.....	14
5.3.1 GetInformation.....	14
5.3.2 checkDataBase.....	15

## 1 INTRODUCTION

### 1.1 OBJET DU DOCUMENT

Ce document contient l'ensemble des informations nécessaires à la compréhension du fonctionnement du web service SEED.

### 1.2 TERMINOLOGIE ET ABRÉVIATIONS

#### 1.2.1 SIGLES ET ABRÉVIATIONS

##### 1.2.1.1 FONCTIONNELS

	Description
ROSA	Référentiel des Opérateurs et de Suivi des Agréments
SEED	System of Exchange of Excise Data.

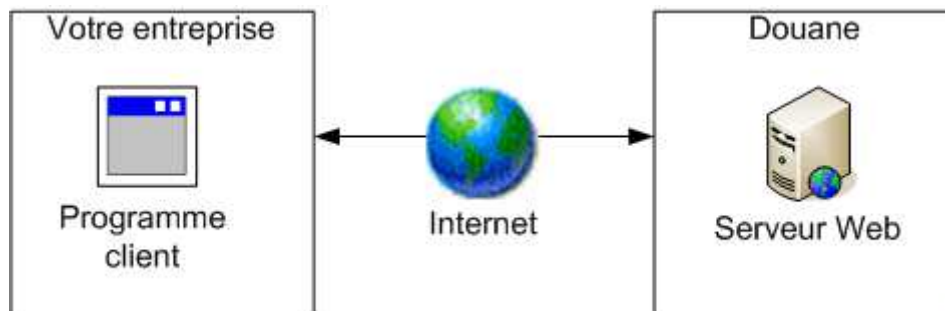
##### 1.2.1.2 TECHNIQUES

	Description
HTTP	HyperText Transfer Protocol
SOAP	Simple Object Access Protocol
WS	Web Service
WSDL	Web Service Description Language
XML	Extensible Markup Language
XSD	XML Schema

## 2 COMMUNICATION AVEC LE WEB SERVICE

### 2.1 ARCHITECTURE GLOBALE

Ci-dessous un schéma global de la communication entre votre programme et le WS qui est situé sur un serveur Web de la Douane. La communication se fait dans les deux sens, car vous faites un appel au WS et celui-ci vous retourne l'information demandée. L'adresse d'appel au WS est : <https://pro.douane.gouv.fr/seedWS>



### 2.2 MÉTHODES DISPONIBLES

Une seule méthode est disponible, il s'agit de : `getInformation`. Cette méthode demande en paramètre une liste de numéros d'accises et retourne les données correspondant à ces numéros d'accises. Pour avoir plus d'information sur cette méthode, le WSDL correspondant : <https://pro.douane.gouv.fr/seedWS/SeedWS?wsdl>

#### Voir Annexe : 5.2 Exemples d'appel

#### 2.2.1 MÉTHODE `GETINFORMATION`

##### 2.2.1.1 OBJECTIF DE LA MÉTHODE

Cette méthode retourne les informations concernant un/des opérateur/s et/ou un/des entrepôt/s soumis à accises, par rapport à une liste de numéros d'accises envoyée en paramètre de la méthode.

##### 2.2.1.2 PARAMÈTRE DE LA MÉTHODE

Cette méthode ne dispose que d'un seul paramètre qui est une liste de numéros d'accises. Voici un exemple :

```
<numAccises>
  <numAccise>ATV0010206902</numAccise>
  <numAccise>BG005100R0009</numAccise>
  <numAccise>FR009039E0001</numAccise>
</numAccises>
```

Cette liste de numéros d'accises ne peut pas contenir plus de 100 nœuds `<< numAccise >>`. Le contenu d'un nœud `<< numAccise >>` doit répondre à l'expression régulière :

```
[A-Za-z]{2}[0-9A-Za-z]{11}
```

qui signifie que le numéro d'accise doit commencer par 2 lettres (majuscule ou minuscule) et être suivi de 11 caractères (lettre ou chiffre). Lors de l'appel à la méthode, le WS vérifiera que le paramètre est valide par rapport aux règles ci-dessus.

### 2.2.1.3 RÉPONSE DE LA MÉTHODE

Cette méthode retourne un flux XML en réponse. Voici un exemple de réponse :

#### Voir Annexe : 5.2 Exemple XML

### 2.2.1.4 DESCRIPTION DU CONTENU DU FICHER XML DE RÉPONSE

#### 2.2.1.4.1 ERREUR

Ce nœud contient une erreur qui peut être due à un mauvais appel ou un problème interne du Web Service.

Balise	Description
erreur	Nœud de base de l'erreur
code	Code de l'erreur
description	Description de l'erreur

#### 2.2.1.4.2 ALERTE

Ce nœud contient un avertissement sur un numéro d'accise demandé. Ce nœud n'empêche pas de recevoir les données des autres numéros d'accises demandés, sous réserve que ces derniers soient valides.

Balise	Description
alerte	Nœud de base de l'avertissement
exciseNumber	Numéro d'accise qui est à l'origine du problème
description	Description de l'alerte

Attribut	Balise	Description
type	alerte	Code de l'erreur : 1 : problème sur les données de l'opérateur ou de l'entrepôt 2 : Opérateur ou entrepôt non valide au moment de l'appel

#### 2.2.1.4.3 OPÉRATEUR

Ce nœud contient les informations d'un opérateur.

Balise	Description
traderAuthorisation	Nœud de base de l'opérateur
traderExciseNumber	Numéro d'accise de l'opérateur
vatNumber	Numéro de TVA
authorisationBeginDate	Date de début d'activité de l'opérateur
authorisationEndDate	Date de fin d'activité de l'opérateur
operatorTypeCode	Type d'opérateur :

SEED\_WebService\_1.2.odt

	1 : EA 2 : OE 3 : RCO 4 : Autre
directDelivery	1 : peut faire de la livraison directe, 0 : ne peut pas
authorisedToSendWithoutConsignee	1 : Autorise a ne pas indiquer le destinataire au moment du départ du mouvement 0 : ne peut pas
referenceNumberOfExciseOffice	Code Europa du service qui gère cet opérateur
exciseProduct	Nœud contenant la liste des codes produits que gère l'opérateur
usingTaxWarehouse	Nœud contenant la liste des entrepôts de l'opérateur
addresses	Nœud contenant la liste des adresses de l'opérateur (différentes langues possibles)

2.2.1.4.4 ENTREPÔT

Ce nœud contient les données d'un entrepôt.

Balise	Description
taxWarehouse	Nœud de base de l'entrepôt
referenceOfTaxWarehouse	Numéro d'accise de l'entrepôt
validityBeginDate	Date de début d'activité de l'entrepôt
validityEndDate	Date de fin d'activité de l'entrepôt
referenceNumberOfExciseOffice	Code Europa du service qui gère cet entrepôt
exciseProduct	Nœud contenant la liste des codes produits que gère l'entrepôt
addresses	Nœud contenant la liste des adresses de l'entrepôt (différentes langues possibles)

2.2.1.4.5 ÉLÉMENTS COMMUN AUX OPÉRATEURS ET ENTREPÔTS

Contenu du nœud ExciseProduct

Balise	Description
exciseProductCode	Code du produit

Contenu du nœud usingTaxWarehouse

Balise	Description
--------	-------------

referenceOfTaxWarehouse	Numéro de l'entrepôt
-------------------------	----------------------

Contenu du nœud addresses

Balise	Description
address	Nœud contenant la liste des adresses (différentes langues possibles)

Contenu du nœud address

Balise	Description
Name	Nom de l'opérateur ou de l'entrepôt
StreetName	Nom de la rue de l'opérateur ou de l'entrepôt
StreetNumber	Numéro de l'opérateur ou de l'entrepôt dans la rue
PostCode	Code postal de l'opérateur ou de l'entrepôt
City	Nom de la ville de l'opérateur ou de l'entrepôt
CountryCode	Code du pays de l'opérateur ou de l'entrepôt
Other	Autres informations

Attribut	Balise	Description
lang	address	Code pays correspondant a la langue dont l'adresse est écrite



### 3 FONCTIONNEMENT DU WEB SERVICE

Le web service SEED s'articule autour de deux applications :

- WebService FO/BO
- Client (client utilisé pour tester la disponibilité du WebService FO)

#### 3.1 WEBSERVICE FO

Le web service front office utilise des ejb3. Il doit donc impérativement être installé sur un serveur ayant la capacité de les traiter (Jboss).

Le web service front office permet de se connecter au web service back office via un accès jnp.

##### 3.1.1 MÉTHODES EXPOSÉES

Le front office expose deux méthodes :

- UtilisateurResFo getInformation(final UtilisateurReqFo numAccise)
  - throws SeedBoException, RemoteException
- SurveillanceResFo checkDataBase()
  - throws SeedBoException, RemoteException

#### Voir l'annexe : 5.2 Exemples d'appel

##### 3.1.1.1 GETINFORMATION

La première renvoie une liste d'« Individus » qui est la modélisation d'une entité (Opérateur (TraderAuthorisation) ou entrepôt (TaxWareHouse)).

Appelé à cette adresse : **https://server:port/seedWS/SeedWS**

##### 3.1.1.2 CHECKDATABASE

La seconde permet de renvoyer un objet : « Surveillance » qui renseigne sur la disponibilité des bases de données SEED et ROSA ainsi que sur le temps d'exécution de la procédure stockée (dbo.EAOE\_SEED\_WS\_Surveillance).

Appelé à cette adresse : **https://server:port/seedWS/SurveillancesWS**

#### 3.2 WEBSERVICE BO

Le web service back office utilise des ejb3. Il doit donc impérativement être installé sur un serveur ayant la capacité de les traiter (Jboss).

##### 3.2.1 MÉTHODES APPELÉES

Le web service back office permet de se connecter aux bases de données SEED et ROSA.

Il expose au front office trois méthodes :

- ArrayList<Individu> getOperateur(String numAccise)
  - throws SeedBoException, RemoteException
- ArrayList<Individu> getOperateur(ArrayList<String> numAccises)
  - throws SeedBoException, RemoteException
- SurveillanceRes getSurveillance()
  - throws SeedBoException, RemoteException

### 3.2.1.1 GETOPERATEUR

Les deux premières méthodes renvoient une liste d'Individus qui est la modélisation d'une entité (Opérateur (TraderAuthorisation) ou entrepôt (TaxWareHouse)).

Lors de la recherche d'une accise, le web service se connecte à la base ROSA si le numéro d'accise commence par : « FR » sinon il se connecte à la base SEED.

### 3.2.1.2 GETSURVEILLANCE

La troisième méthode permet de renvoyer un objet : « Surveillance » qui renseigne sur la disponibilité des bases de données SEED et ROSA ainsi que sur le temps d'exécution d'une procédure stockée (dbo.EAOE\_SEED\_WS\_Surveillance).

Les informations de temps remontés sont comptés en milliseconde.

**Voir annexe : 5.2.2 checkDataBase**

## 3.2.2 ALERTES

### 3.2.2.1 VALIDATION DES DONNÉES

Lors de l'utilisation de la méthode : getOperateur

Une vérification des données est réalisée par le Web Service. Si les données ne sont pas valides, le Web Service retourne le **code erreur 1** :

**Problème sur les données de l'opérateur ou de l'entrepôt**

Les vérifications suivantes sont réalisées :

**Voir annexe : 5.2.1 GetInformation**

#### 3.2.2.1.1 TRADERAUTHORISATION

Liste des contrôles réalisés dans le cas d'un TraderAuthorisation (Voir annexe pour plus de détails sur les méthodes de contrôle) :

checkNumTVA, checkDate, checkCountry, checkOperatorType, checkCodeBureau,  
checkProduits, checkEntrepots, checkAdresses

#### 3.2.2.1.2 TAXWAREHOUSE

Liste des contrôles réalisés dans le cas d'un TaxWareHouse (Voir annexe pour plus de détails sur les méthodes de contrôle) :

checkDate, checkCountry, checkCodeBureau, checkProduits, checkAdresses

### 3.2.2.2 OPÉRATEUR INVALIDE

Lors de l'utilisation de la méthode : getOperateur

Le web service BO recherche le numéro d'accise, soit dans ROSA, soit dans SEED. Si le numéro d'accise commence par FR ou autre. Si l'accise n'existe pas alors le Web Service retourne le **code erreur 2** :

**Opérateur ou entrepôt non valide au moment de l'appel**

## 3.2.3 LISTE DES PROCÉDURES STOCKÉES UTILISÉES

### 3.3 CLIENT

Le client se connecte au web service pour vérifier que celui-ci fonctionne correctement.

### 4 SERVEUR JBOSS

Les échanges de flux entre le client et le WebService FO se font par défaut sur le port 8080 pour les requêtes http et sur le port 1099 pour les requêtes jnp entre le WebService FO et le WebService BO.

## 5 ANNEXE

### 5.1 VALIDATION DES DONNÉES

#### 5.1.1 CHECKNUMTVA

La valeur n'est pas obligatoire.

Si elle est présente, elle doit respecter la regex suivante : `.{1,35}`

#### 5.1.2 CHECKDATE

La date de début est obligatoire, la date de fin n'est pas obligatoire.

Si elle est présente, la date de début doit être inférieur à la date de fin.

Si elle est présente, la date de fin doit être supérieure à la date de début.

#### 5.1.3 CHECKCOUNTRY

La valeur est obligatoire, ne peut être null en base.

Le pays doit être l'un des pays référencé : `bg|at|be|cy|cz|de|dk|ee|el|es|fi|fr|gb|hu|ie|it|lt|lu|lv|mt|nl|pl|pt|ro|se|si|sk`

#### 5.1.4 CHECKOPERATORTYPE

La valeur est obligatoire, ne peut être null en base.

Le type de l'opérateur ne peut être que 1 ou 2

#### 5.1.5 CHECKCODEBUREAU

La valeur est obligatoire, ne peut être null en base.

Le code bureau doit respecter la regex suivante : `[A-Z]{2}[a-zA-Z0-9]{6}`

#### 5.1.6 CHECKPRODUITS

La valeur n'est pas obligatoire.

Si elle est présente, les produits doivent respecter la regex suivante : `[A-Z][0-9]{3}`

Si elle est présente, les catégories produits doivent respecter la regex suivante : `[A-Z]{1}`

#### 5.1.7 CHECKENTREPOTS

Il doit y avoir au minimum un entrepôt.

Pour chaque entrepôt le numéro d'accise doit respecter la regex suivante :

`[A-Za-z]{2}[0-9A-Za-z]{11}`

#### 5.1.8 CHECKADRESSES

La valeur n'est pas obligatoire.

Si elle est présente, aucun des champs de l'adresse ne peut être vide ou Null

Et le code pays est validé par la fonction : `CheckCountry`

### 5.2 EXEMPLES D'APPEL

#### 5.2.1 GETINFORMATION

Exemple de requête d'appel SOAP au web service :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ws="http://ws.seed.douane.finances.gouv.fr/">
  <soapenv:Header/>
  <soapenv:Body>
    <ws:getInformation>
      <numAccises>
        <numAccise>FR0000000001</numAccise>
      </numAccises>
    </ws:getInformation>
  </soapenv:Body>
</soapenv:Envelope>
```

## 5.2.2 CHECKDATABASE

Exemple de requête d'appel SOAP au web service :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ws="http://ws.seed.douane.finances.gouv.fr/">
  <soapenv:Header/>
  <soapenv:Body>
    <ws:checkDataBase/>
  </soapenv:Body>
</soapenv:Envelope>
```

## 5.3 EXEMPLE XML

### 5.3.1 GETINFORMATION

Exemple de XML retourné par le web service SEED : « problème de données », « numéro invalide » et « données valide ».

```
<root>
  <uid>aa8426fc-1dd0-42e3-865e-5a5872517fad</uid>
  <alertes type="1">
    <exciseNumber>GBWK935270300</exciseNumber>
    <description>Problème de données</description>
  </alertes>
  <alertes type="2">
    <exciseNumber>FR015000E1239</exciseNumber>
    <description>Numéro invalide</description>
  </alertes>
  <taxWarehouse>
    <referenceOfTaxWarehouse>FR00000000003</referenceOfTaxWarehouse>
    <validityBeginDate>2003-02-01</validityBeginDate>
    <validityEndDate>2010-12-15</validityEndDate>
    <country>FR</country>
    <referenceNumberOfExciseOffice>FR001260</referenceNumberOfExciseOffice>
    <exciseProduct>
      <exciseProductCode>E200</exciseProductCode>
      <exciseProductCode>E300</exciseProductCode>
      <exciseProductCode>E410</exciseProductCode>
      <exciseProductCode>E420</exciseProductCode>
      <exciseProductCode>E430</exciseProductCode>
      <exciseProductCode>E440</exciseProductCode>
      <exciseProductCode>E450</exciseProductCode>
      <exciseProductCode>E460</exciseProductCode>
      <exciseProductCode>E470</exciseProductCode>
      <exciseProductCode>E480</exciseProductCode>
      <exciseProductCode>E490</exciseProductCode>
      <exciseProductCode>E500</exciseProductCode>
      <exciseProductCode>E600</exciseProductCode>
      <exciseProductCode>E700</exciseProductCode>
      <exciseProductCode>E800</exciseProductCode>
      <exciseProductCode>E910</exciseProductCode>
      <exciseProductCode>E920</exciseProductCode>
      <exciseProductCode>E930</exciseProductCode>
    </exciseProduct>
  </taxWarehouse>
  <addresses>
    <address lang="fr">
      <name>RAFFINERIE DE DUNKERQUE</name>
      <streetName>Route DE L'OUVRAGE OUEST</streetName>
      <streetNumber>
      <postCode>59140</postCode>
      <city>DUNKERQUE</city>
    </address>
  </addresses>
</root>
```

SEED\_WebService\_1.2.odt

---

```
<countryCode>FR</countryCode>
</address>
</addresses>
</taxWarehouse>
</root>
```

### 5.3.2 CHECKDATABASE

Exemple de XML retourné par le web service SEED :

```
<root>
  <accesBO>true</accesBO>
  <rosa>true</rosa>
  <seed>true</seed>
  <timeRosa>6.0</timeRosa>
  <timeSeed>30.0</timeSeed>
</root>
```