

Documentation utilisateur pour les webservices RESWEL

Version partielle

HISTORIQUE DES VERSIONS

Version	Date de rédaction	Motif
0.1	25 février 2013	Version initiale
0.2	24 septembre 2013	Ajouts précisions sur utilisation Annuaire National
2.0	24 juin 2016	Ajouts indications pour utilisation en mode REST
2.1	26 octobre 2023	Accès annuaire uniquement en HTTPS, TLS1.0 et 1.1 désactivés

LISTE DES EVOLUTIONS APPORTEES DEPUIS LA VERSION 0.1

- **Version 0.2** :

Ajouts précisions sur annuaire national (serveur de secours)

- **Version 2.0** :

Ajouts indications pour utilisation en mode REST

- **Version 2.1** :

Accès à l'annuaire en HTTPS uniquement (TLS1.2 minimum)

SOMMAIRE

1.	Introduction	5
1.1.	But de ce document	5
1.2.	Vocabulaire	5
1.3.	Architecture générale	6
1.4.	Séquence d'appel à un webservice métier	7
2.	Appel à un webservice « RESWEL » en mode synchrone	8
	Version SOAP	8
	Description des balises	8
	Version REST	8
	Liste des codes Http utilisés	10
2.1.	Informations techniques nécessaires pour solliciter un webservice « RESWEL »	10
2.2.	Recherche de l'URL métier à contacter	11
2.3.	Authentification.....	12
2.4.	Appel au webservice métier	12
3.	Webservice Annuaire (WsAnnuaire)	13
3.1.	Opération – Recherche d'une URL.....	13
3.1.1.	Nomenclature de l'opération	13
3.1.2.	Définition des paramètres en entrée	13
	Description des balises	14
3.1.3.	Définition des paramètres en sortie.....	14
	Description des balises	15
	Exemple – Paramètres en entrée	15
	Exemple – Paramètres en sortie quand webservice métier SOAP uniquement.....	15
	Exemple – Paramètres en sortie quand webservice métier SOAP et REST	16
3.1.4.	Définition avec Swagger pour une utilisation en REST.....	17
	Exemple – Paramètres en entrée (POST)	17
	Exemple – Paramètres en sortie (Code Http=200)	18
3.2.	Opération – Recherche de la liste des webservices disponibles	18
3.2.1.	Nomenclature de l'opération	18
3.2.2.	Définition des paramètres en entrée	18
	Description des balises	19
3.2.3.	Définition des paramètres en sortie.....	20
	Description des balises	20
	Exemple « Recherche des webservices bovins » – Paramètres en entrée.....	20
	Exemple – Paramètres en sortie	21
3.2.4.	Définition avec Swagger pour une utilisation en REST.....	22
3.3.	Opération – Recherche des versions disponibles pour un webservice	23
3.3.1.	Nomenclature de l'opération	23
3.3.2.	Définition des paramètres en entrée	23
	Description des balises	23
3.3.3.	Définition des paramètres en sortie.....	24
	Description des balises	24
	Exemple « Recherche les versions disponibles pour IpBNotif » – Paramètres en entrée	24
	Exemple – Paramètres en sortie	25
3.3.4.	Définition avec Swagger pour une utilisation en REST.....	26
3.4.	Opération – Recherche les opérations d'une version d'un webservice	26
3.4.1.	Nomenclature de l'opération	26
3.4.2.	Définition des paramètres en entrée	27
	Description des balises	27
3.4.3.	Définition des paramètres en sortie.....	28
	Description des balises	28

Exemple « Recherche les opérations disponibles dans IpBNotif » – Paramètres en entrée	29
Exemple – Paramètres en sortie	29
3.4.4. Définition avec Swagger pour une utilisation en REST	30
4. Webservice Guichet (WsGuichet)	31
4.1. Opération – Authentification	31
4.1.1. Nomenclature de l'opération	31
4.1.2. Définition des paramètres en entrée	31
Description des balises	31
4.1.3. Définition des paramètres en sortie	32
Description des balises	32
Exemple – Paramètres en entrée	32
Exemple – Paramètres en sortie	33
4.1.4. Définition avec Swagger pour une utilisation en REST	33
5. ANNEXE A : Types complexes utilisés dans les opérations	34
5.1. typeReponse	34
5.2. typeAnomalie	34
5.3. typeInfosServiceMetierRegional	35
5.4. typeProfil	35
5.5. typeVersionPK	35
5.6. typIdentification	36
5.7. typeServices	36
5.8. typeService	37
5.9. typeServicePK	37
5.10. typeVersions	37
5.11. typeVersion	38
5.12. typeOperations	38
5.13. typeOperation	39
5.14. typeServiceVersion	39

1. Introduction

1.1. But de ce document

Le présent document est destiné aux éditeurs de logiciels, consommateurs des webservices conçus selon le plan qualité RESWEL de FIEA.

Il a pour but de :

- Décrire le protocole pour solliciter un webservice conçu selon le plan qualité RESWEL
- Décrire les fonctionnalités de l'annuaire (aide au routage)

1.2. Vocabulaire

Les termes définis ci-après seront utilisés dans ce document.

Webservice : un webservice est une application web (HTTP ou HTTPS) fournissant de 1 à n opérations.

MOA : Acronyme pour désigner la Maîtrise d'Ouvrage des webservices « métier ».

Opération : une opération est un traitement informatique capable de fournir un service à un consommateur. Le service rendu peut être une mise à jour d'un système d'information, une extraction de données, etc.

Exemples : Naissance, Inventaire, ...

Consommateur : un consommateur est une application informatique sollicitant une opération d'un webservice pour obtenir un service. Un consommateur peut être aussi bien un logiciel détenteur, qu'un applicatif web ou une application mobile, etc.

Annuaire : un annuaire de webservice est une liste décrivant tous les webservices et toutes les opérations mis à disposition des consommateurs.

WSDL : un WSDL est une définition normalisée décrivant les opérations fournies par un webservice SOAP. En d'autres termes, un WSDL décrit l'interface d'un webservice.

RESWEL : RESWEL est le nom du projet FIEA fournissant un socle technique aux différents ARSOE pour concevoir et héberger des webservices selon un plan qualité commun.

URL : une URL est une simple ligne de texte qui permet de retrouver une ressource (page web, webservice, image, ...) sur internet.

SOAP : SOAP est un protocole de communication en XML et standardisé par le W3C.

REST : REST est une architecture de webservice s'appuyant sur le protocole HTTP, les données étant généralement définies en JSON.

1.3. Architecture générale

Les webservices « RESWEL », conçus à partir d'une modélisation commune nationale, permettent de publier une interface identique quel que soit le site informatique ARSOE contacté par le consommateur.

Les webservices « RESWEL » peuvent exposer une signature en **SOAP** ou en **REST** ou les deux à la fois. Cela dépend des choix effectués par les Maitrises d'Ouvrage (MOA).

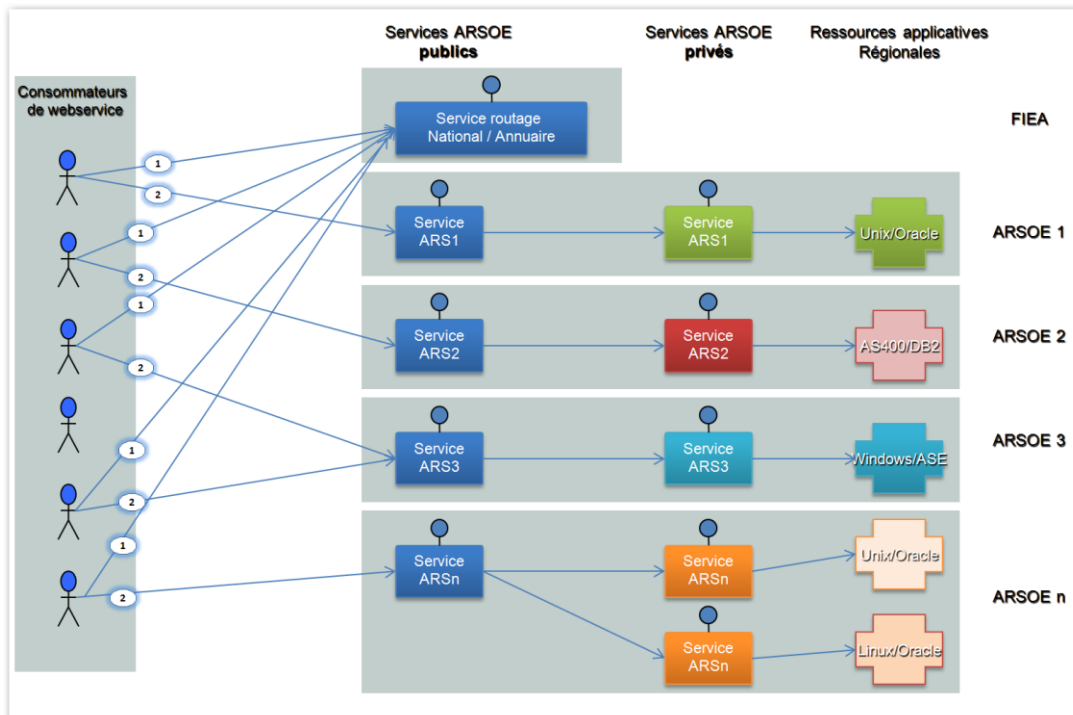
Quand ils sont conçus pour être utilisé en **SOAP**, ils sont modélisés selon la méthode « **Contract first** ». Cela permet ainsi de fournir un WSDL (plusieurs fichiers XML) aux consommateurs sans que le webservice ne soit encore implémenté.

Quand ils sont conçus pour être utilisé en **REST**, le contrat de service est décrit dans les spécifications et disponible avec des outils tels que **Swagger UI**. Les données sont définies en JSON.

Les webservices « RESWEL » sont principalement **synchrones**.

Ils disposent d'un annuaire national permettant aux consommateurs de savoir quel est le site informatique à contacter pour obtenir un service et dans quelle technologie (SOAP / REST) les solliciter.

Voici la représentation générale du système :



Correspond à un webservice commun hébergé par chaque site informatique ARSOE

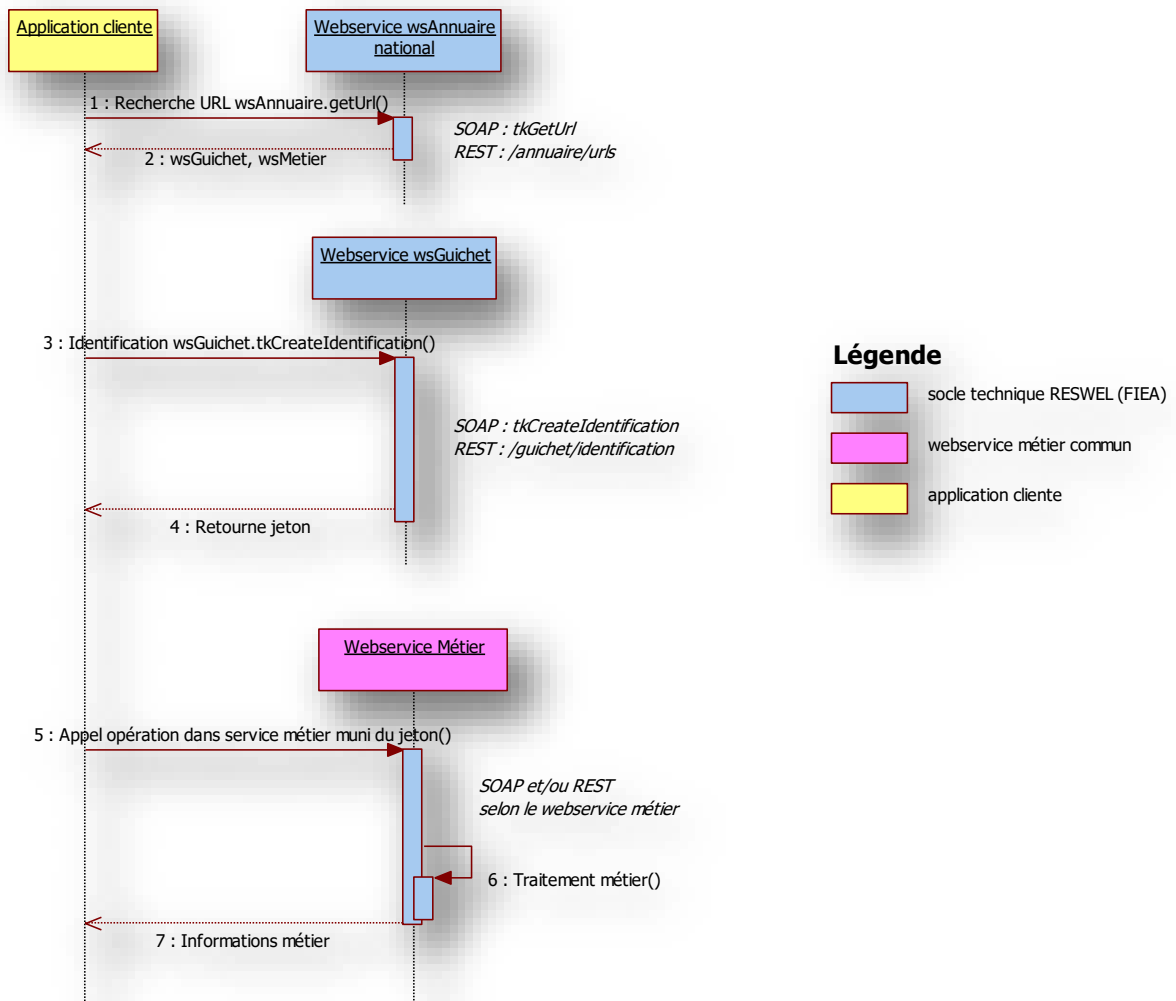


Correspond aux webservices régionaux, spécifiques à chaque ARSOE, **mais respectant l'interface nationale du webservice commun.**

1.4. Séquence d'appel à un webservice métier

La séquence d'appel à un webservice métier peut se résumer de la manière suivante :

1. Le consommateur contacte l'annuaire pour obtenir le webservice guichet et le webservice métier à utiliser
2. Il appelle ensuite le webservice technique pour l'authentification et l'obtention d'un jeton (HTTPS)
3. Il appelle le webservice métier muni du jeton
4. Le webservice métier régional contrôle la validité de la demande, effectue le traitement et renvoie la réponse



L'annuaire et le guichet RESWEL exposent les deux signatures **SOAP** et **REST**.

2. Appel à un webservice « RESWEL » en mode synchrone

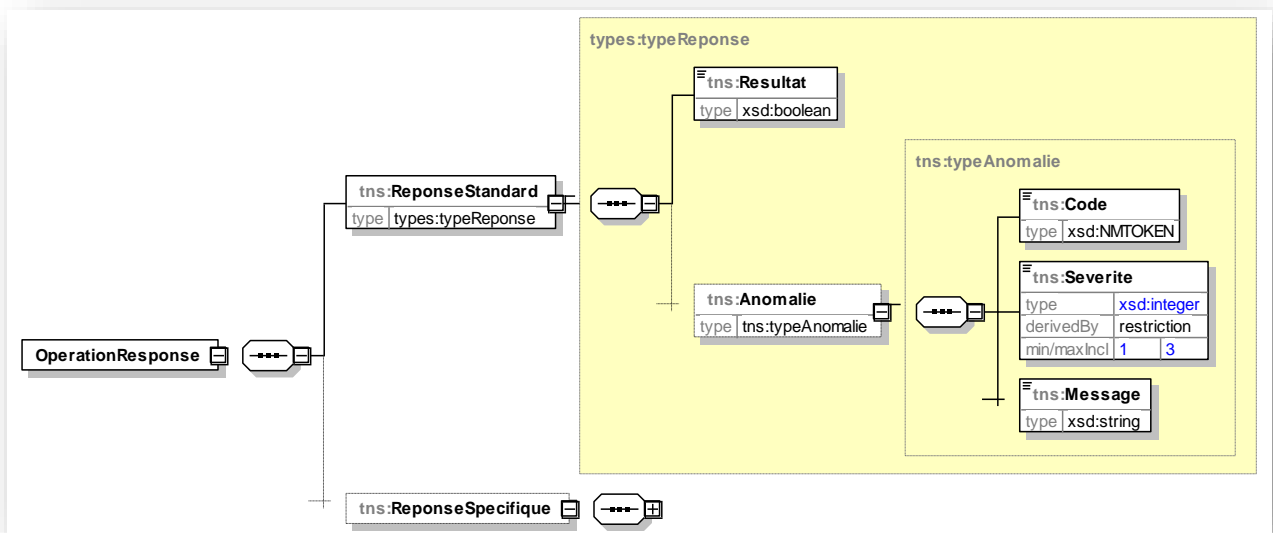
Pour utiliser un webservice « RESWEL » en mode synchrone, le consommateur devra suivre dans l'ordre les étapes suivantes :

- Connaître les informations techniques pour accéder au webservice
- Rechercher l'URL du webservice métier par l'intermédiaire de l'annuaire
- Authentification sur le site informatique hébergeant le webservice métier
- Utilisation du webservice métier

Toutes les opérations des webservices « RESWEL » (y compris les opérations de l'annuaire) ont une réponse normalisée.

Version SOAP

Les réponses SOAP sont de la forme suivante :



Description des balises

BALISE	SEQUENCE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
ReponseStandard	1	1	Type complexe typeReponse	Renseigne sur l'exécution de l'opération (OK, KO).
ReponseSpecifique	0	1	Type complexe	Type complexe spécifique à l'opération métier. Cette information n'est renseignée que si Resultat est OK.

Les types complexes sont documentés dans l'annexe A.

Version REST

En REST la demande est réalisée par une opération POST avec l'objet « Request » sérialisé en JSON sur la ressource associée à l'opération (ex: */[url de base]/annuaire/urls*).

Exemple de POST pour obtenir les URL liées au webservice EDNOTIF pour E540 :

```
{
  "ProfilDemandeur": {
    "Application": "RESWEL",
    "Entreprise": "E540"
  },
  "VersionPK": {
    "NomService": "IpBNotif",
    "CodeSiteService": "9",
    "NumeroVersion": "1.00",
    "CodeSiteVersion": "9"
  }
}
```

La réponse est gérée selon des paradigmes REST, c'est-à-dire que le résultat de l'opération est fourni sous forme de code Http. La valeur 200 correspond à un résultat OK. La liste des codes est détaillées ci-après.

- Dans le cas d'un code différent de 200, l'objet « Anomalie » sérialisé en JSON est renvoyé dans la réponse.

Exemple d'anomalie avec code Http=403 :

```
{
  "Code": "wtk003",
  "Severite": 1,
  "Message": "Le composant regional a refuse les informations d'authentification."
}
```

- Si le code Http=200, le contenu de l'objet « ReponseSpecifique » est retournée, sérialisé en JSON, directement dans la réponse.

Exemple de résultat :

```
{
  "UrlGuichet": "https://wstestg-directory.fiea.fr:8443/wsguichet/WsGuichet",
  "WsdlGuichet": "https://wstestg-directory.fiea.fr:8443/wsguichet/WsGuichet?wsdl",
  "UrlMetier": "https://wstestg-directory2.arsoe-nordest.com/wslpBNotif_100/wslpBNotif",
  "WsdlMetier": "https://wstestg-directory2.arsoe-nordest.com/wslpBNotif_100/wslpBNotif?wsdl"
}
```

Liste des codes Http utilisés

CODE HTTP	SEVERITE	LIBELLE ANOMALIE
200	1	Requête traitée avec succès
204	1	Requête traitée avec succès mais pas d'information à renvoyer
400	1	Requête invalide
403	1	Accès refusé
500	1	Erreur Interne

2.1. Informations techniques nécessaires pour solliciter un webservice « RESWEL »

L'URL de l'annuaire national des webservices « RESWEL » est diffusée par FIEA. **Il s'agit d'accès en HTTPS avec TLS1.2 minimum.** Les TLS 1.0 et 1.1 ne sont pas actifs.

L'annuaire de l'environnement de **PRODUCTION** est disponible à l'adresse suivante :

<https://ws-directory.fiea.fr/wsannuaire/WsAnnuaire>

L'annuaire de l'environnement de **TEST** est disponible à l'adresse suivante :

<https://wstest-directory.fiea.fr/wsannuaire/WsAnnuaire>

Pour assurer une haute disponibilité, une redondance a été mise en place sur le service annuaire.

Ainsi dans le cas où le serveur annuaire primaire ne répond pas dans un temps adéquat, il est possible de solliciter un serveur secondaire. Pour faire cela, il est nécessaire de recueillir les différentes adresses IP concernant l'annuaire (ws-directory.fiea.fr pour la **production** et wstest-directory.fiea.fr pour les **tests**). Voici les fonctions à utiliser selon la plateforme de développement utilisée :

- Java InetAddress.getAllByName
- Visual Studio .Net Dns.GetHostEntry
- Windev DNSListeAdresse

Pour accéder à un webservice « RESWEL », il est nécessaire de connaître les informations suivantes : **nom du webservice, identifiant du webservice, code site¹ et version.**

Attention, **la casse de ces informations est importante.**

Il est nécessaire également au consommateur, pour renseigner son profil, de connaître le **code de l'entreprise** qu'il doit contacter et son **paramètre « zone »** s'il doit être utilisé. Ces deux informations sont fournies directement par l'entreprise au consommateur lors de la contractualisation de l'abonnement au service (convention). A cette occasion, l'entreprise communique également le login et le mot de passe au consommateur.

¹ Code site=9 indique qu'il s'agit d'un webservice commun national.

Voici un exemple des informations nécessaires à connaître pour l'accès au webservice EDNOTIF de l'EDE 54 par l'exploitation FR54001001 :

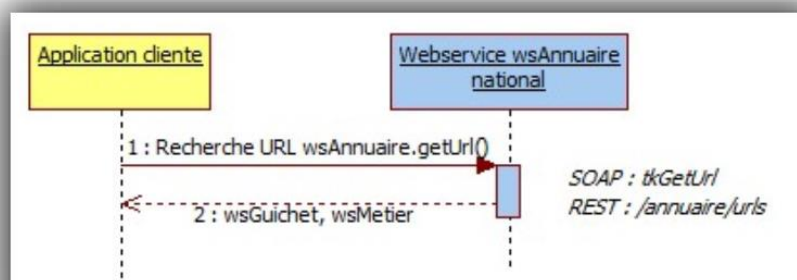
ANNUAIRE RESWEL	https://ws-directory.fiea.fr/wsannuaire/WsAnnuaire
NOM DU WEBSERVICE	wslpBNotif
IDENTIFIANT DU WEBSERVICE	lpBNotif
CODE SITE	9
VERSION	1.00
CODE ENTREPRISE	E540
ZONE	
LOGIN	54001001
MOT DE PASSE	****

2.2. Recherche de l'URL métier à contacter

Avec les informations collectées à l'étape précédente, le consommateur doit solliciter l'annuaire national « RESWEL » pour obtenir :

- L'URL à laquelle il devra s'authentifier. Cette URL est l' « **URL Guichet** ».
- L'URL à laquelle il pourra ensuite solliciter le webservice métier. Il s'agit de l' « **URL métier** ».

Pour cela il doit solliciter en SOAP ou en REST l'opération tkGetUrl décrite au §3, opération qui est publiée dans le WSDL de l'annuaire.

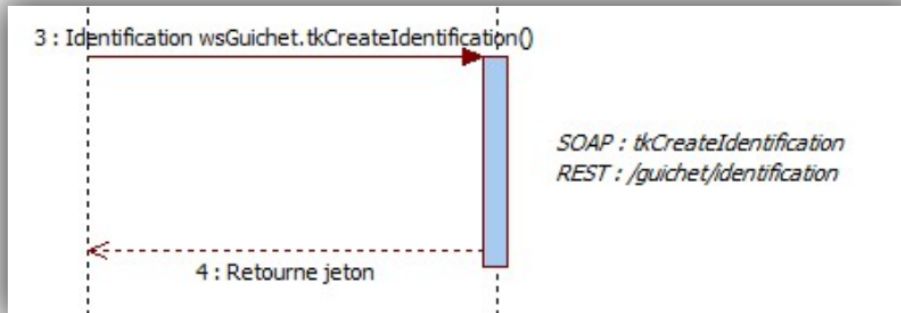


A l'issue, le consommateur dispose des URL « Guichet » et « Métier ». Il est informé si une version SOAP et/ou REST sont disponibles.

2.3. Authentification

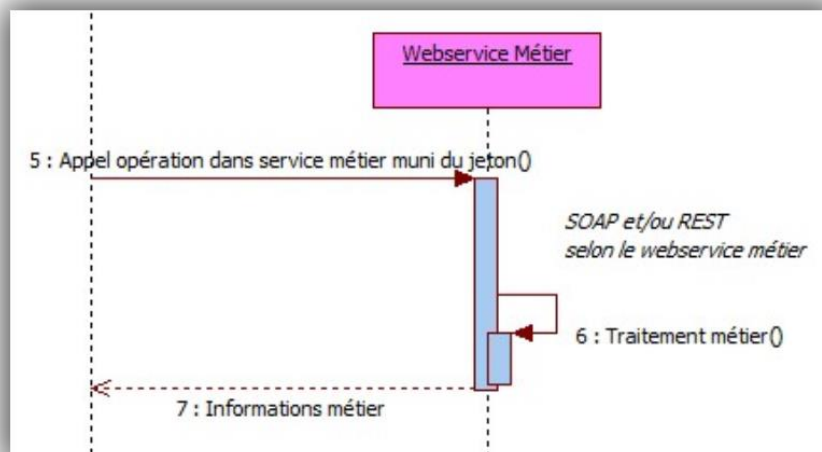
Le consommateur peut maintenant s'authentifier au près du webservice disponible à l'« URL Guichet » obtenue à l'étape précédente. Pour cela il doit solliciter l'opération tkCreateIdentification décrite au §4, opération qui est publiée dans le WSDL du guichet.

En retour favorable de l'authentification, le consommateur disposera d'un jeton d'authentification. Ce paramètre est nommé « JetonAuthentification » dans les différentes opérations « métier ».



2.4. Appel au webservice métier

Le consommateur peut solliciter maintenant l'« URL métier » en appelant les opérations disponibles et en précisant le jeton d'authentification comme paramètre dans le mode disponible (SOAP et/ou REST), en fonction du résultat obtenu lors de la recherche de URL du §2.2.



3. Webservice Annuaire (WsAnnuaire)

Les opérations de l'annuaire sont disponibles « informatiquement » dans le WSDL composé des fichiers suivants :

- WsAnnuaire.wsdl
- tk_types.xsd
- fiea_types.xsd

La description REST avec Swagger est disponible sur un annuaire RESWEL en interrogeant l'URL suivante : <https://site.informatique.fr/wsannuaire>

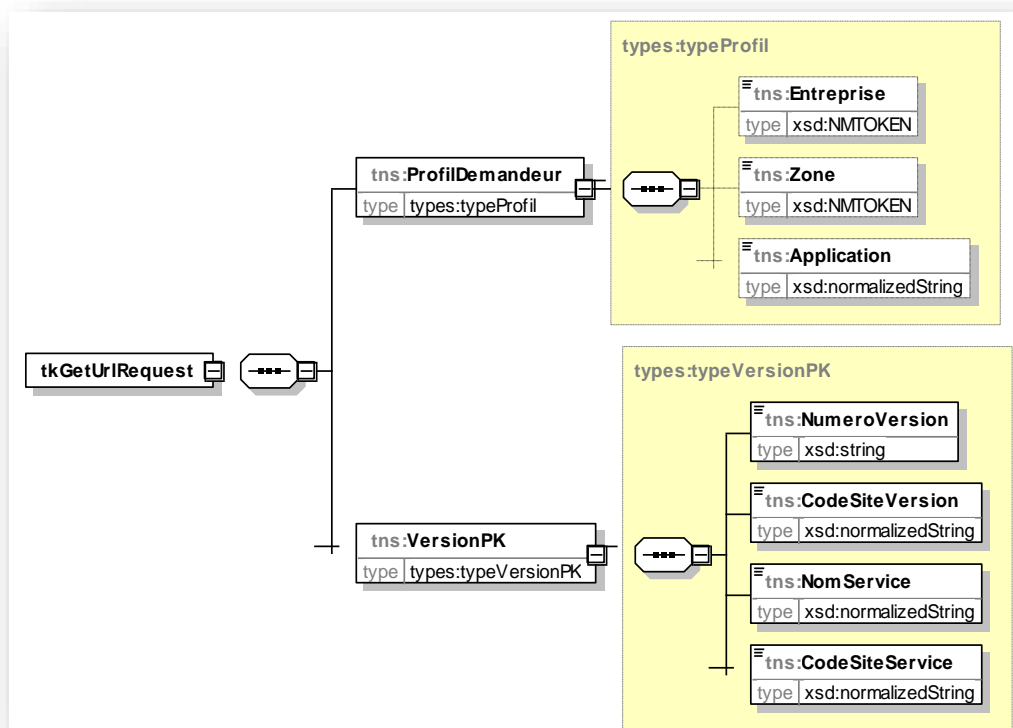
3.1. Opération – Recherche d'une URL

3.1.1. Nomenclature de l'opération

NOM DE L'OPERATION SOAP	tkGetUrl
RESSOURCE REST	/annuaire/urls

Cette opération permet de rechercher les URL « guichet » et « métier » à utiliser pour un webservice dans une version donnée et un profil donné.

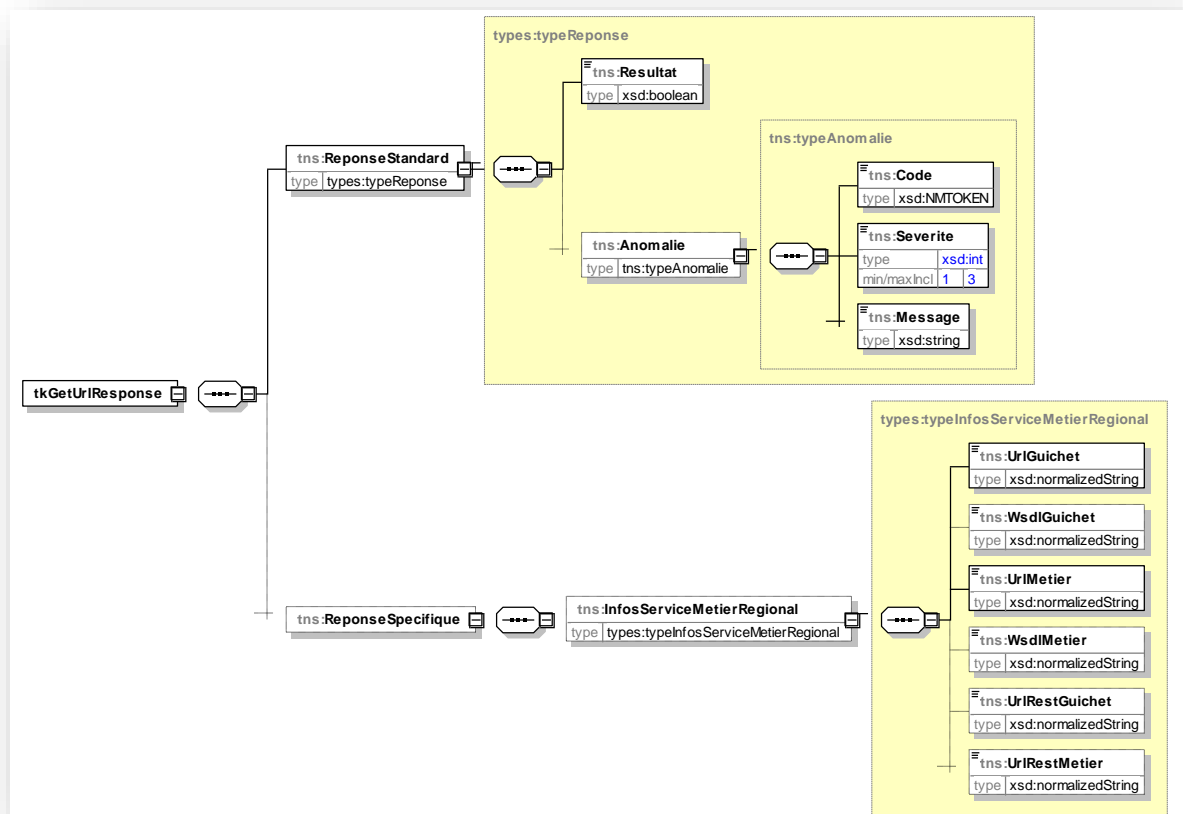
3.1.2. Définition des paramètres en entrée



Description des balises

BALISE	MULTIPLICITE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
ProfilDemandeur	1	1	Type complexe typeProfil	Profil du consommateur (entreprise à contacter)
VersionPk	1	1	Type complexe typeVersionPK	Identifiant d'un webservice métier

3.1.3. Définition des paramètres en sortie



Description des balises

BALISE	MULTIPLICITE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
ReponseStandard	1	1	Type complexe typeReponse	Renseigne sur l'exécution de l'opération (OK, KO).
ReponseSpecifique	0	1	Type complexe typeInfosServiceMetierRegional	Réponse métier si exécution est OK.

Exemple – Paramètres en entrée

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  <soapenv:Header/>
  <soapenv:Body>
    <tk:tkGetUrlRequest>
      <tk:ProfilDemandeur>
        <typ:Entreprise>E540</typ:Entreprise>
        <typ:Application>LOGICIEL DE NOTIFICATION</typ:Application>
      </tk:ProfilDemandeur>
      <tk:VersionPK>
        <typ:NumeroVersion>1.00</typ:NumeroVersion>
        <typ:CodeSiteVersion>9</typ:CodeSiteVersion>
        <typ:NomService>IpBNotif</typ:NomService>
        <typ:CodeSiteService>9</typ:CodeSiteService>
      </tk:VersionPK>
    </tk:tkGetUrlRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Exemple – Paramètres en sortie quand webservice métier SOAP uniquement

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/en
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:tkGetUrlResponse xmlns="http://www.fiea.org/types/" xmlns:ns2="http://www.fiea.org/tk/">
      <ns2:ReponseStandard>
        <Resultat>true</Resultat>
      </ns2:ReponseStandard>
      <ns2:ReponseSpecifique>
        <ns2:InfosServiceMetierRegional>
          <UrlGuichet>https://site.informatique.fr/wsguichet/WsGuichet</UrlGuichet>
          <WsdGuichet>https://site.informatique.fr/wsguichet/WsGuichet?wsdl</WsdGuichet>
          <UrlMetier>https://site.informatique.fr/wsIpBNotif_100/wsIpBNotif</UrlMetier>
          <WsdMetier>https://site.informatique.fr/wsIpBNotif_100/wsIpBNotif?wsdl</WsdMetier>
        </ns2:InfosServiceMetierRegional>
      </ns2:ReponseSpecifique>
    </ns2:tkGetUrlResponse>
  </S:Body>
</S:Envelope>
```

Exemple – Paramètres en sortie quand webservice métier SOAP et REST

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:tkGetUrlResponse xmlns="http://www.fiea.org/types/" xmlns:ns2="http://www.fiea.org/tk/">
      <ns2:ReponseStandard>
        <Resultat>true</Resultat>
      </ns2:ReponseStandard>
      <ns2:ReponseSpecifique>
        <ns2:InfosServiceMetierRegional>
          <UrlGuichet>https://site.informatique.fr/wsguichet/WsGuichet</UrlGuichet>
          <Wsd1Guichet>https://site.informatique.fr/wsguichet/WsGuichet?wsdl</Wsd1Guichet>
          <UrlMetier>https://site.informatique.fr/wsIpBNotif_100/wsIpBNotif</UrlMetier>
          <UrlRestGuichet>https://site.informatique.fr/wsguichet/rest</UrlRestGuichet>
          <UrlRestMetier>https://site.informatique.fr/wsIpBNotif/rest</UrlRestMetier>
        </ns2:InfosServiceMetierRegional>
      </ns2:ReponseSpecifique>
    </ns2:tkGetUrlResponse>
  </S:Body>
</S:Envelope>
```


3.1.4. Définition avec Swagger pour une utilisation en REST

POST /annuaire/urls Recherche la liste des urls disponibles

Response Class
Model | Model Schema

```
TypeInfosServiceMetierRegional {
  UriGuichet (string),
  WsdlGuichet (string, optional),
  UriMetier (string),
  WsdlMetier (string, optional),
  UriRestGuichet (string, optional),
  UriRestMetier (string, optional)
}
```

Response Content Type

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	<input type="text" value="(required)"/> <small>Parameter content type: <input type="text" value="application/json"/></small>	Parametres	body	Model Model Schema tkGetUrlRequest { ProfilDemandeur (TypeProfil), VersionPK (TypeVersionPK) } TypeProfil { Zone (string, optional), Application (string, optional), Entreprise (string, optional) } TypeVersionPK { NomService (string), CodeSiteService (string), NumeroVersion (string), CodeSiteVersion (string) }

Exemple – Paramètres en entrée (POST)

```
{
  "ProfilDemandeur": {
    "Application": "LOGICIEL DE NOTIFICATION",
    "Entreprise": "E540"
  },
  "VersionPK": {
    "NomService": "IpBNotif",
    "CodeSiteService": "9",
    "NumeroVersion": "1.00",
    "CodeSiteVersion": "9"
  }
}
```

Exemple – Paramètres en sortie (Code Http=200)

```

{
  "UrlGuichet": "https://site.informatique.fr/wsguichet/WsGuichet",
  "WsdlGuichet": "https://site.informatique.fr/wsguichet/WsGuichet?wsdl",
  "UrlMetier": "https://site.informatique.fr/wsIpBNotif_100/wsIpBNotif",
  "WsdlMetier": "https://site.informatique.fr/wsIpBNotif_100/wsIpBNotif?wsdl",
  "UrlRestGuichet": "https://site.informatique.fr/wsguichet/rest",
  "UrlRestMetier": "https://site.informatique.fr/wsIpBNotif/rest"
}

```

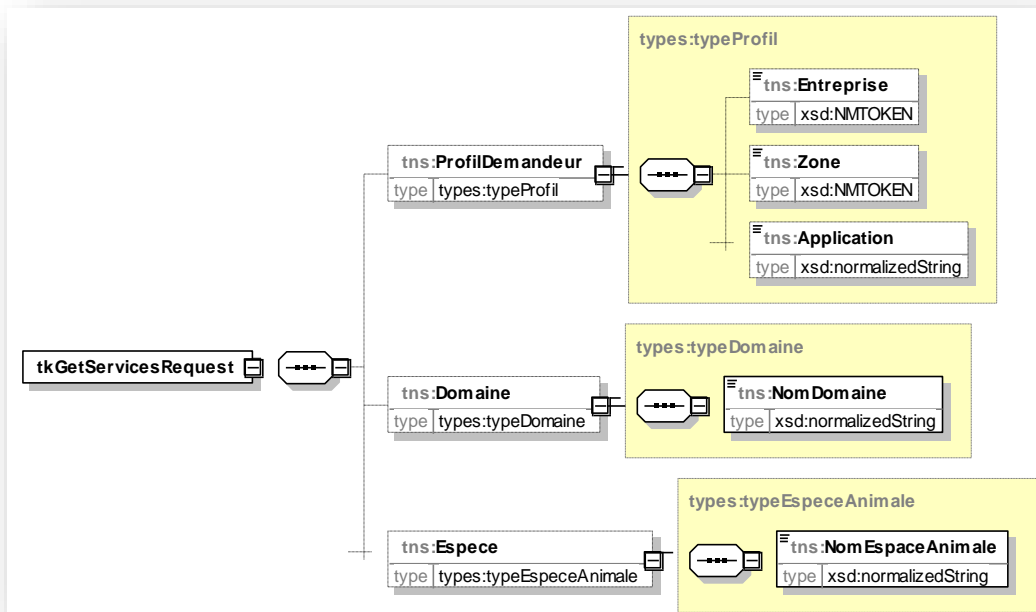
3.2. Opération – Recherche de la liste des webservices disponibles

3.2.1. Nomenclature de l'opération

NOM DE L'OPERATION SOAP	tkGetServices
RESSOURCE REST	/annuaire/services

Cette opération permet de rechercher la liste des webservices disponibles pour une espèce animale, un domaine applicatif et selon un profil.

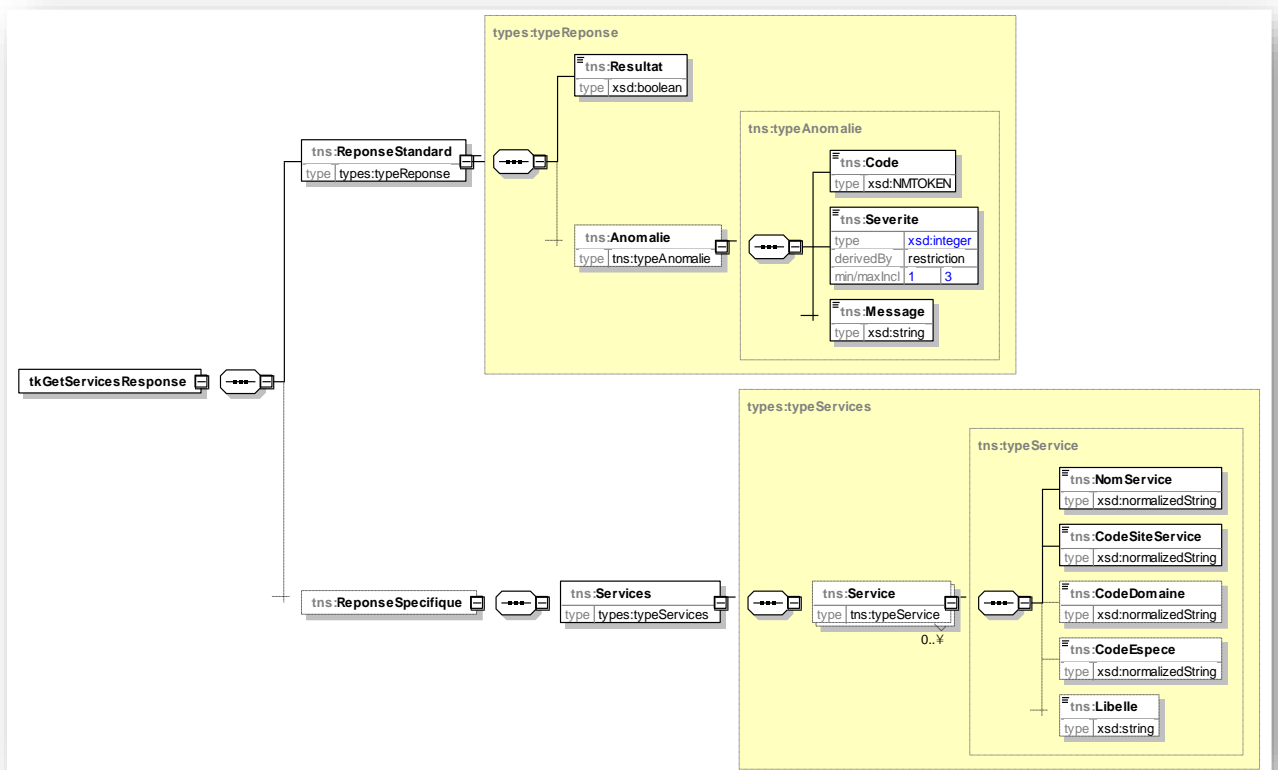
3.2.2. Définition des paramètres en entrée



Description des balises

SEQUENCE				
BALISE	MULTIPLICITE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
ProfilDemandeur	0	1	Type complexe typeProfil	Profil du consommateur (entreprise à contacter)
Domaine	0	1	Type simple typeDomaine	Domaine applicatif : Ip=IPG Cp=Certification des Parentés Bovines Ia=Insémination Cl=Contrôle Laitier Va=Vaches Allaitantes Os=UPRA Md=Multi-domaines (SIG, EDEL, ...) Tk=Opérations techniques ne concernant pas un domaine particulier
Espece	0	1	Type simple typeEspeceAnimale	Espèce animale : B=Espèce bovine Z=Petits ruminants (ovin/caprin) O=Espèce ovine seule C=Espèce caprine seule

3.2.3. Définition des paramètres en sortie



Description des balises

BALISE	SEQUENCE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
ReponseStandard	1	1	Type complexe typeReponse	Renseigne sur l'exécution de l'opération (OK, KO).
ReponseSpecifique	0	1	Type complexe typeServices	Réponse métier si exécution est OK.

Exemple « Recherche des webservices bovins » – Paramètres en entrée

```

<?xml version='1.0' encoding='UTF-8'>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <tk:tkGetServicesRequest>
      <tk:Espece>
        <typ:NomEspaceAnimale>B</typ:NomEspaceAnimale>
      </tk:Espece>
    </tk:tkGetServicesRequest>
  </soapenv:Body>
</soapenv:Envelope>

```

Exemple – Paramètres en sortie

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:tkGetServicesResponse xmlns="http://www.fiea.org/types/">
      <ns3:ReponseStandard>
        <Resultat>true</Resultat>
      </ns3:ReponseStandard>
      <ns3:ReponseSpecifique>
        <ns3:Services>
          <Service>
            <NomService>IpBNotif</NomService>
            <CodeSiteService>9</CodeSiteService>
            <CodeDomaine>Ip</CodeDomaine>
            <CodeEspece>B</CodeEspece>
            <Libelle>Notification IPG Bovine</Libelle>
          </Service>
          <Service>
            <NomService>MdBTest</NomService>
            <CodeSiteService>9</CodeSiteService>
            <CodeDomaine>Md</CodeDomaine>
            <CodeEspece>B</CodeEspece>
            <Libelle>Webservice de test</Libelle>
          </Service>
        </ns3:Services>
      </ns3:ReponseSpecifique>
    </ns3:tkGetServicesResponse>
  </S:Body>
</S:Envelope>
```

3.2.4. Définition avec Swagger pour une utilisation en REST

POST /annuaire/services Recherche la liste des services disponibles

Response Class

Model | Model Schema

```

TypeServices {
  Service (array[TypeService], optional)
}
TypeService {
  Libelle (string, optional),
  NomService (string),
  CodeSiteService (string),
  CodeDomaine (string, optional),
  CodeEspece (string, optional)
}

```

Response Content Type: application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	(required)	Parametres	body	Model Model Schema

Parameter content type: application/json

```

tkGetServicesRequest {
  ProfilDemandeur (TypeProfil, optional),
  Domaine (TypeDomaine, optional),
  Espece (TypeEspeceAnimale, optional)
}
TypeProfil {
  Zone (string, optional),
  Application (string, optional),
  Entreprise (string, optional)
}
TypeDomaine {
  NomDomaine (string)
}
TypeEspeceAnimale {
  NomEspeceAnimale (string)
}

```

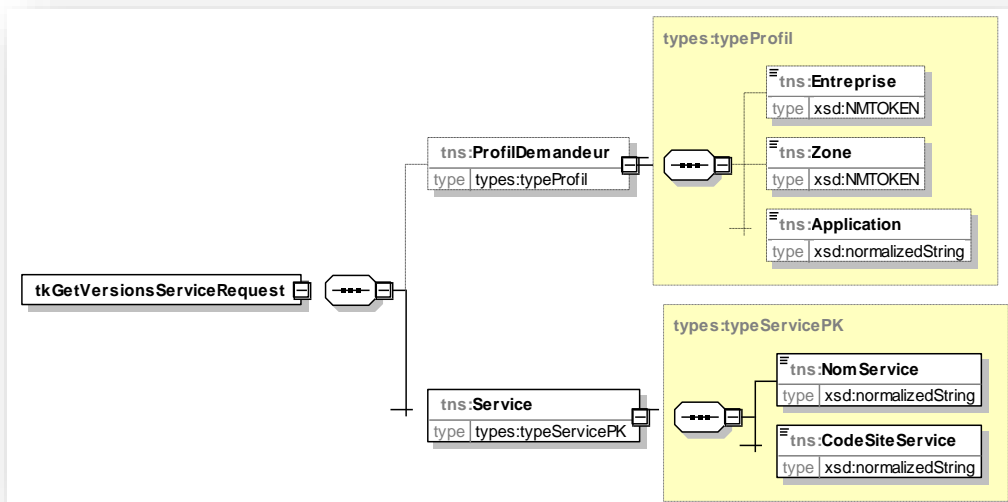
3.3. Opération – Recherche des versions disponibles pour un webservice

3.3.1. Nomenclature de l'opération

NOM DE L'OPERATION SOAP	tkVersionsService
RESSOURCE REST	/annuaire/versionsService

Cette opération permet de rechercher les différentes versions disponibles pour un webservice et un profil donnés.

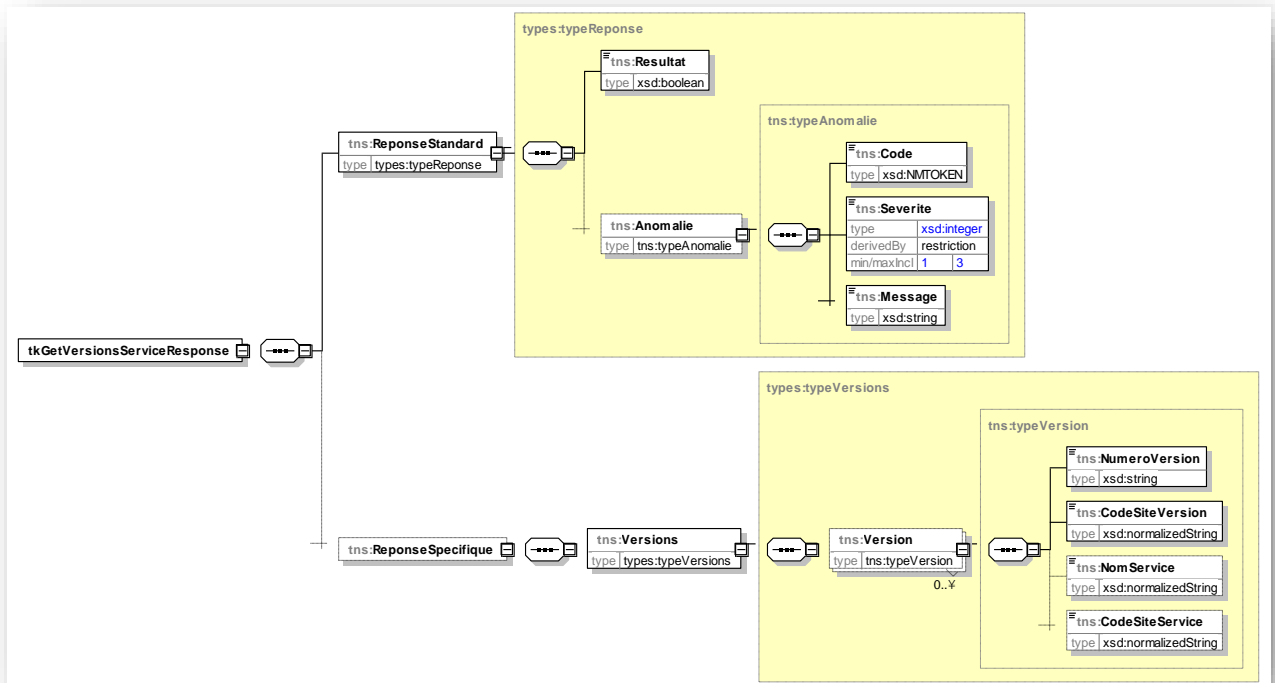
3.3.2. Définition des paramètres en entrée



Description des balises

BALISE	SEQUENCE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
ProfilDemandeur	1	1	Type complexe typeProfil	Profil du consommateur (entreprise à contacter)
Service	1	1	Type complexe typeServicePK	Code d'un webservice métier

3.3.3. Définition des paramètres en sortie



Description des balises

BALISE	SEQUENCE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
ReponseStandard	1	1	Type complexe typeReponse	Renseigne sur l'exécution de l'opération (OK, KO).
ReponseSpecifique	0	1	Type complexe typeVersions	Réponse métier si exécution est OK.

Exemple « Recherche les versions disponibles pour IpBNotif » – Paramètres en entrée

```
<?xml version='1.0' encoding='UTF-8'>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <tk:tkGetVersionsServiceRequest>
      <tk:Service>
        <typ:NomService>IpBNotif</typ:NomService>
        <typ:CodeSiteService>9</typ:CodeSiteService>
      </tk:Service>
    </tk:tkGetVersionsServiceRequest>
  </soapenv:Body>
</soapenv:Envelope>
```


Exemple – Paramètres en sortie

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:tkGetVersionsServiceResponse xmlns="http://www.fiea.org/types/"
      <ns3:ReponseStandard>
        <Resultat>true</Resultat>
      </ns3:ReponseStandard>
      <ns3:ReponseSpecifique>
        <ns3:Versions>
          <Version>
            <NumeroVersion>1.0.0</NumeroVersion>
            <CodeSiteVersion>9</CodeSiteVersion>
            <NomService>IpBNotif</NomService>
            <CodeSiteService>9</CodeSiteService>
          </Version>
          <Version>
            <NumeroVersion>1.1.0</NumeroVersion>
            <CodeSiteVersion>9</CodeSiteVersion>
            <NomService>IpBNotif</NomService>
            <CodeSiteService>9</CodeSiteService>
          </Version>
        </ns3:Versions>
      </ns3:ReponseSpecifique>
    </ns3:tkGetVersionsServiceResponse>
  </S:Body>
</S:Envelope>
```

3.3.4. Définition avec Swagger pour une utilisation en REST

POST /annuaire/versionsService Recherche la liste des versions pour un service donne

Response Class
Model | Model Schema

```

TypeVersions {
  Version (array[TypeVersion], optional)
}
TypeVersion {
  NomService (string, optional),
  CodeSiteService (string, optional),
  NumeroVersion (string),
  CodeSiteVersion (string)
}

```

Response Content Type

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	(required)	Parametres	body	Model Model Schema tkGetVersionsServiceRequest { Service (TypeServicePK), ProfilDemandeur (TypeProfil, optional) } TypeServicePK { NomService (string), CodeSiteService (string) } TypeProfil { Zone (string, optional), Application (string, optional), Entreprise (string, optional) }

Parameter content type:

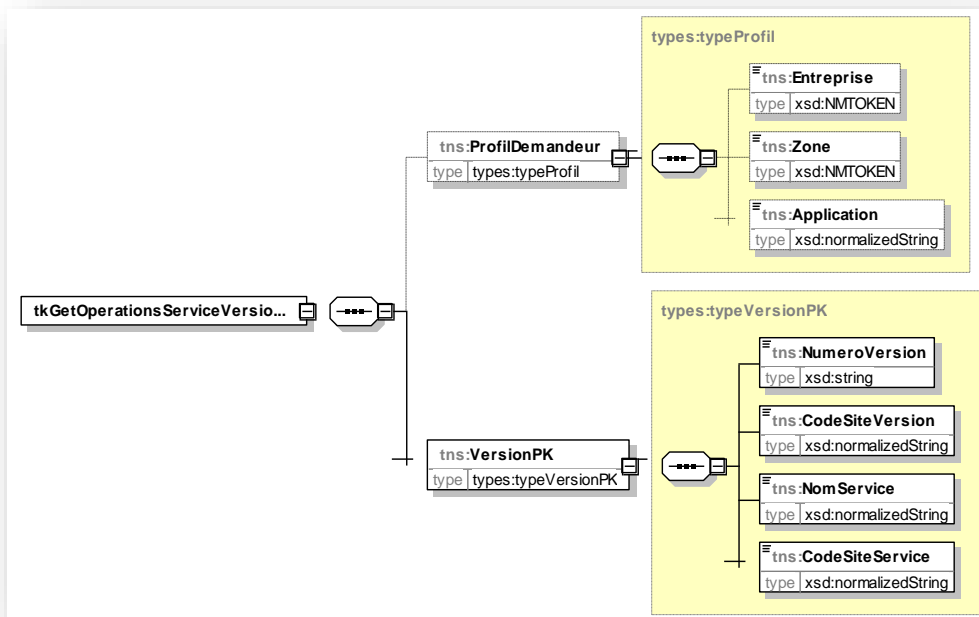
3.4. Opération – Recherche les opérations d'une version d'un webservice

3.4.1. Nomenclature de l'opération

NOM DE L'OPERATION SOAP	tkGetOperationsServiceVersion
RESSOURCE REST	/annuaire/operationsServiceVersion

Cette opération permet de rechercher les opérations disponibles dans la version d'un webservice et d'un profil donné.

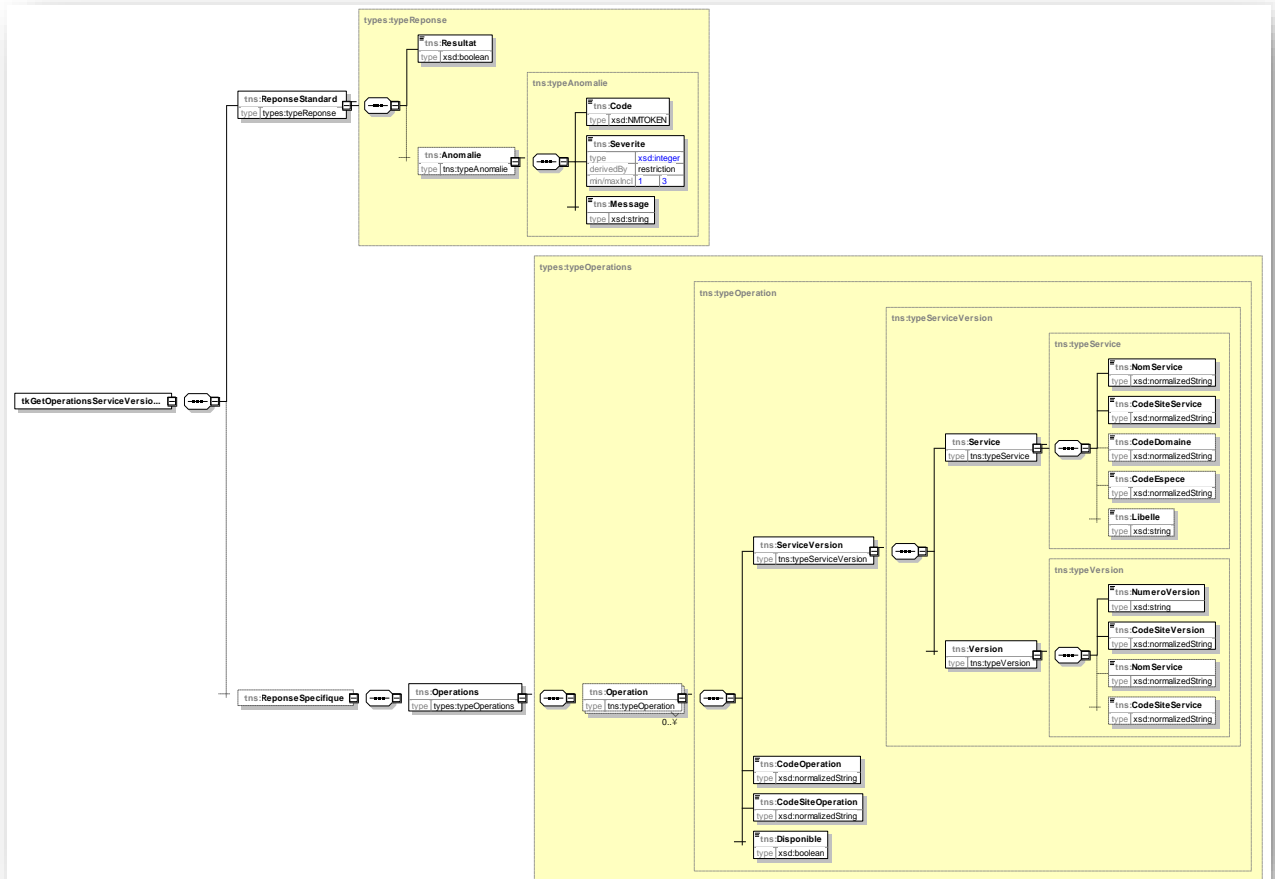
3.4.2. Définition des paramètres en entrée



Description des balises

SEQUENCE				
BALISE	MULTIPLICITE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
ProfilDemandeur	0	1	Type complexe typeProfil	Profil du consommateur (entreprise à contacter)
VersionPk	1	1	Type complexe typeVersionPK	Identifiant d'un webservice métier

3.4.3. Définition des paramètres en sortie



Description des balises

BALISE	SEQUENCE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
ReponseStandard	1	1	Type complexe typeReponse	Renseigne sur l'exécution de l'opération (OK, KO).
ReponseSpecifique	0	1	Type complexe typeOperations	Réponse métier si exécution est OK.

Exemple « Recherche les opérations disponibles dans IpBNotif » – Paramètres en entrée

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  <soapenv:Header/
  <soapenv:Body>
    <tk:tkGetOperationsServiceVersionRequest>
      <tk:ProfilDemandeur>
        <typ:Entreprise>E540</typ:Entreprise>
      </tk:ProfilDemandeur>
      <tk:VersionPK>
        <typ:NumeroVersion>1.0.0</typ:NumeroVersion>
        <typ:CodeSiteVersion>9</typ:CodeSiteVersion>
        <typ:NomService>IpBNotif</typ:NomService>
        <typ:CodeSiteService>9</typ:CodeSiteService>
      </tk:VersionPK>
    </tk:tkGetOperationsServiceVersionRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Exemple – Paramètres en sortie

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:tkGetOperationsServiceVersionResponse xmlns="http://www.fiea.org/types/"
      <ns3:ReponseStandard>
        <Resultat>true</Resultat>
      </ns3:ReponseStandard>
      <ns3:ReponseSpecifique>
        <ns3:Operations>
          <Operation>
            <ServiceVersion>
              <Service>
                <NomService>IpBNotif</NomService>
                <CodeSiteService>9</CodeSiteService>
              </Service>
              <Version>
                <NumeroVersion>1.0.0</NumeroVersion>
                <CodeSiteVersion>9</CodeSiteVersion>
                <NomService>IpBNotif</NomService>
                <CodeSiteService>9</CodeSiteService>
              </Version>
            </ServiceVersion>
            <CodeOperation>IpBCreateEntree</CodeOperation>
            <CodeSiteOperation>9</CodeSiteOperation>
            <Disponible>true</Disponible>
          </Operation>
          <Operation>
            <ServiceVersion>
              <Service>
                <NomService>IpBNotif</NomService>
                <CodeSiteService>9</CodeSiteService>
              </Service>
              <Version>
                <NumeroVersion>1.0.0</NumeroVersion>
                <CodeSiteVersion>9</CodeSiteVersion>
                <NomService>IpBNotif</NomService>
                <CodeSiteService>9</CodeSiteService>
              </Version>
            </ServiceVersion>
            <CodeOperation>IpBGetInventaire</CodeOperation>
            <CodeSiteOperation>9</CodeSiteOperation>
            <Disponible>true</Disponible>
          </Operation>
        </ns3:Operations>
      </ns3:ReponseSpecifique>
    </ns3:tkGetOperationsServiceVersionResponse>
  </S:Body>
</S:Envelope>
```

3.4.4. Définition avec Swagger pour une utilisation en REST

POST /annuaire/operationsServiceVersion Recherche la liste des operations pour un service et une version donnees

Response Class

Model | Model Schema

```

TypeOperations {
  Operation (array[TypeOperation], optional)
}
TypeOperation {
  CodeSiteOperation (string),
  CodeOperation (string),
  ServiceVersion (TypeServiceVersion),
  Disponible (boolean, optional)
}
TypeServiceVersion {
  Version (TypeVersion),
  Service (TypeService)
}
TypeVersion {
  NomService (string, optional),
  CodeSiteService (string, optional),
  NumeroVersion (string),
  CodeSiteVersion (string)
}
TypeService {
  Libelle (string, optional),
  NomService (string),
  CodeSiteService (string),
  CodeDomaine (string, optional),
  CodeEspece (string, optional)
}

```

Response Content Type: application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	(required)	Parametres	body	Model Model Schema tkGetOperationsServiceVersionRequest { ProfilDemandeur (TypeProfil, optional), VersionPK (TypeVersionPK) } TypeProfil { Zone (string, optional), Application (string, optional), Entreprise (string, optional) } TypeVersionPK { NomService (string), CodeSiteService (string), NumeroVersion (string), CodeSiteVersion (string) }

Parameter content type: application/json

4. Webservice Guichet (WsGuichet)

L'opération du guichet est disponible « informatiquement » dans le WSDL composé des fichiers suivants :

- WsGuichet.wsdl
- tk_types.xsd
- fiea_types.xsd

La description REST avec Swagger est disponible sur un guichet RESWEL en interrogeant l'URL suivante : <https://site.informatique.fr/wsguichet>

Ce webservice n'est disponible qu'en **protocole sécurisé (HTTPS)**.

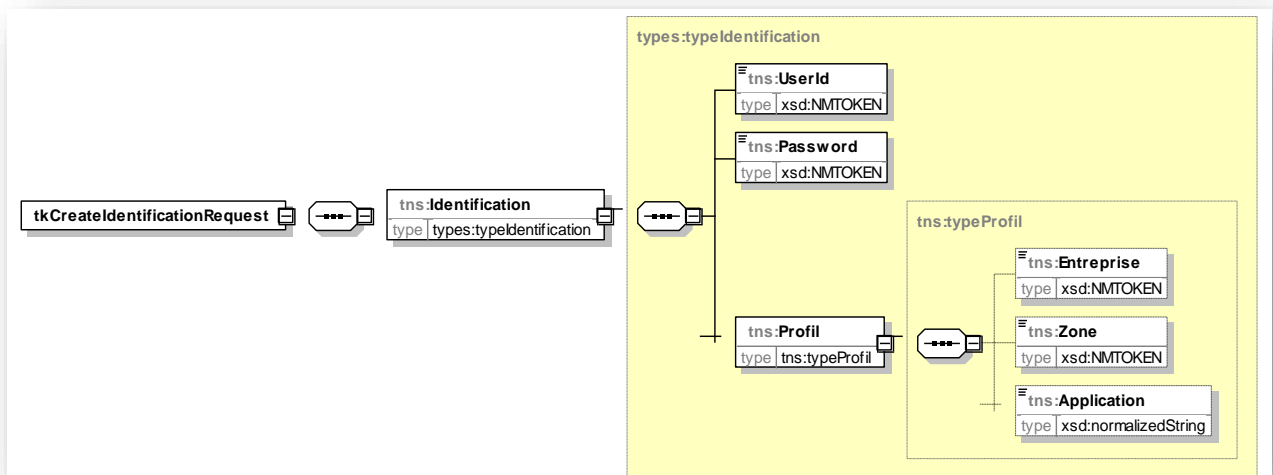
4.1. Opération – Authentification

4.1.1. Nomenclature de l'opération

NOM DE L'OPERATION SOAP	tkCreateIdentification
RESSOURCE REST	/guichet/identification

Cette opération permet au consommateur d'être authentifié. Si l'authentification réussie, le webservice délivre un jeton au consommateur qu'il devra passer en paramètre à chaque opération « métier ».

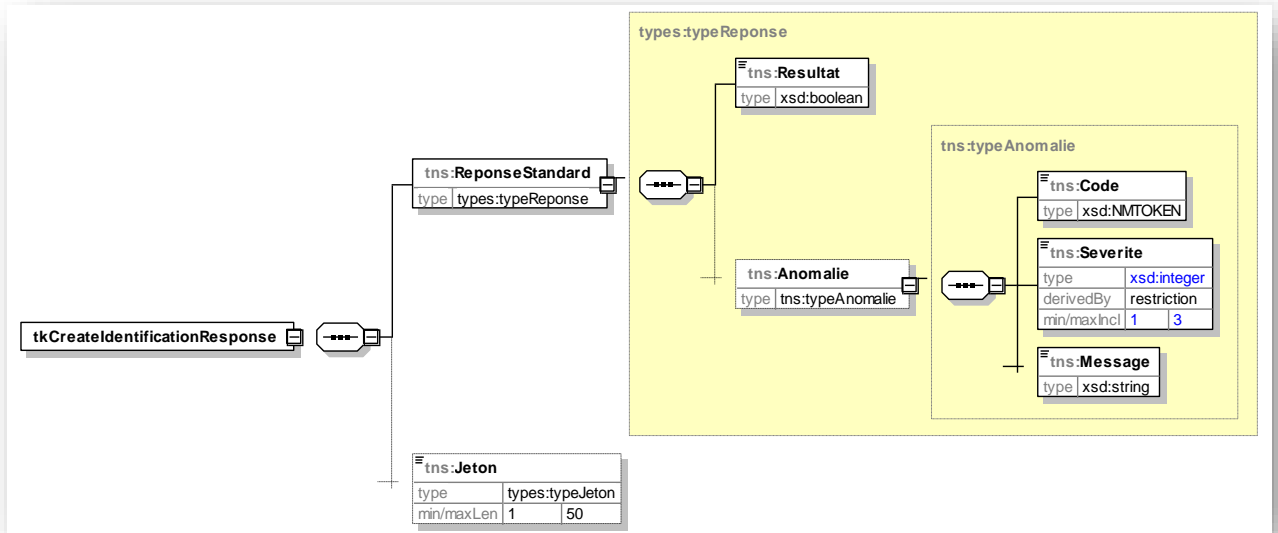
4.1.2. Définition des paramètres en entrée



Description des balises

BALISE	SEQUENCE		TYPE DE DONNEES	COMMENTAIRE
	MULTIPLICITE MIN	MULTIPLICITE MAX		
Identification	1	1	Type complexe typeIdentification	Login/mot de passe utilisateur avec son profil

4.1.3. Définition des paramètres en sortie



Description des balises

SEQUENCE				
BALISE	MULTIPLICITE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
ReponseStandard	1	1	Type complexe typeReponse	Renseigne sur l'exécution de l'opération (OK, KO).
Jeton	0	1	String(50)	Jeton d'authentification obtenu si Resultat=OK

Exemple – Paramètres en entrée

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  <soapenv:Header/>
  <soapenv:Body>
    <tk:tkCreateIdentificationRequest>
      <tk:Identification>
        <typ:UserId>54001001</typ:UserId>
        <typ:Password>****</typ:Password>
        <typ:Profil>
          <typ:Entreprise>E540</typ:Entreprise>
          <typ:Application>LOGICIEL DE NOTIFICATION</typ:Application>
        </typ:Profil>
      </tk:Identification>
    </tk:tkCreateIdentificationRequest>
  </soapenv:Body>
</soapenv:Envelope>
  
```


Exemple – Paramètres en sortie

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:tkCreateIdentificationResponse xmlns="http://www.fiea.org/types/"
      <ns3:ReponseStandard>
        <Resultat>true</Resultat>
      </ns3:ReponseStandard>
      <ns3:Jeton>f515d688-6af1-4bc2-9b68-2f1c8bf59f14</ns3:Jeton>
    </ns3:tkCreateIdentificationResponse>
  </S:Body>
</S:Envelope>
```

4.1.4. Définition avec Swagger pour une utilisation en REST

POST /guichet/identification Créer un jeton

Implementation Notes
Operation d'authentification. Cree un jeton d'identification.

Response Class
Model | Model Schema

```
CreateIdentificationResponse {
  jeton (string, optional)
}
```

Response Content Type

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	(required)	Informations d'authentification	body	Model Model Schema tkCreateIdentificationRequest { Identification (TypeIdentification) } TypeIdentification { Password (string), UserId (string), Profil (TypeProfil) } TypeProfil { Zone (string, optional), Application (string, optional), Entreprise (string, optional) }

Parameter content type:

5. ANNEXE A : Types complexes utilisés dans les opérations

5.1. typeReponse

Il s'agit du type complexe correspondant à la réponse standard de chaque opération pour connaître le résultat (OK, KO avec message d'erreur).

SEQUENCE				
BALISE	MULTIPLICITE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
Resultat	1	1	Boolean	Indique si l'opération s'est déroulée sans problème. Quand resultat=True, une anomalie peut être retournée , mais elle est forcément de sévérité 2 ou 3 . Quand resultat=False, une anomalie est retournée , mais elle est forcément de sévérité 1 .
Anomalie	0	1	Type complexe typeAnomalie	Anomalie fournie en réponse à l'opération

5.2. typeAnomalie

Il s'agit du type complexe correspondant à la description d'une anomalie.

SEQUENCE				
BALISE	MULTIPLICITE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
Code	1	1	NMTOKEN	Code de l'anomalie
Severite	1	1	Int (1 à 3)	Indique le niveau de sévérité de l'anomalie : <ul style="list-style-type: none"> • 1 Erreur bloquante • 2 Erreur non bloquante nécessitant une correction • 3 Message d'information
Message	1	1	String	Message détaillant la raison de l'anomalie

5.3. typeInfosServiceMetierRegional

Il s'agit du type complexe correspondant à la réponse métier de tkGetUrl.

SEQUENCE				
BALISE	MULTIPLICITE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
UrlGuichet	1	1	String	URL du webservice guichet SOAP où l'authentification doit être faite
WsdIGuichet	0	1	String	URL du WSDL du webservice guichet SOAP où l'authentification doit être faite
UrlMetier	1	1	String	URL webservice métier SOAP
WsdIMetier	0	1	String	URL du WSDL du webservice métier SOAP
UrlRestGuichet	0	1	String	URL du webservice guichet REST où l'authentification doit être faite
UrlRestMetier	0	1	String	URL webservice métier REST

5.4. typeProfil

Il s'agit du type complexe correspondant au profil du consommateur.

SEQUENCE				
BALISE	MULTIPLICITE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
Entreprise	0	1	NMTOKEN	Code de l'entreprise avec laquelle le consommateur a contractualisé le service
Zone	0	1	NMTOKEN	Zone géographique d'intervention de l'entreprise
Application	0	1	String	Nom du logiciel du consommateur

5.5. typeVersionPK

Il s'agit du type complexe correspondant à l'identifiant d'un webservice dans une version.

SEQUENCE				
BALISE	MULTIPLICITE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
NumeroVersion	1	1	String	Numéro de version du webservice
CodeSiteVersion	1	1	String	Code site de la version (9 pour les webservices communs nationaux)
NomService	1	1	String	Identifiant du webservice
CodeSiteService	1	1	String	Code site du webservice (9 pour les webservices communs nationaux)

5.6. typeIdentification

Il s'agit du type complexe correspondant à la description de l'identification d'un consommateur.

BALISE	MULTIPLICITE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
Userld	1	1	NMTOKEN	Login du consommateur
Password	1	1	NMTOKEN	Mot de passe
Profil	1	1	Type complexe typeProfil	Profil du consommateur

5.7. typeServices

Il s'agit du type complexe correspondant à une liste de services.

BALISE	MULTIPLICITE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
Service	0	∞	Type complexe typeService	Liste des services disponibles

5.8. typeService

Il s'agit du type complexe correspondant à la description d'un service.

SEQUENCE				
BALISE	MULTIPLICITE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
NomService	1	1	String	Identifiant du webservice
CodeSiteService	1	1	String	Code site du webservice (9 pour les webservices communs nationaux)
CodeDomaine	0	1	String	Domaine applicatif : Ip=IPG Cp=Certification des Parentés Bovines Ia=Insémination Cl=Contrôle Laitier Va=Vaches Allaitantes Os=UPRA Md=Multi-domaines (SIG, EDEL, ...) Tk=Opérations techniques ne concernant pas un domaine particulier
CodeEspece	0	1	String	Espèce animale : B=Espèce bovine Z=Petits ruminants (ovin/caprin) O=Espèce ovine seule C=Espèce caprine seule
Libelle	0	1	String	Commentaire décrivant le webservice

5.9. typeServicePK

Il s'agit du type complexe correspondant à l'identifiant d'un webservice.

SEQUENCE				
BALISE	MULTIPLICITE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
NomService	1	1	String	Identifiant du webservice
CodeSiteService	1	1	String	Code site du webservice (9 pour les webservices communs nationaux)

5.10. typeVersions

Il s'agit du type complexe correspondant à une liste de versions pour un webservice.

SEQUENCE				
BALISE	MULTIPLICITE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
Version	0	∞	Type complexe typeVersion	Liste des versions disponibles

5.11. typeVersion

Il s'agit du type complexe correspondant à la description d'une version.

SEQUENCE				
BALISE	MULTIPLICITE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
NumeroVersion	1	1	String	Numéro de version du webservice
CodeSiteVersion	1	1	String	Code site de la version (9 pour les webservices communs nationaux)
NomService	0	1	String	Identifiant du webservice
CodeSiteService	0	1	String	Code site du webservice (9 pour les webservices communs nationaux)

5.12. typeOperations

Il s'agit du type complexe correspondant à une liste des opérations disponibles dans un webservice.

SEQUENCE				
BALISE	MULTIPLICITE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
Operation	0	∞	Type complexe typeOperation	Liste des opérations disponibles

5.13. *typeOperation*

Il s'agit du type complexe correspondant à la description d'une opération de webservice.

SEQUENCE				
BALISE	MULTIPLICITE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
ServiceVersion	1	1	Type complexe type ServiceVersion	Numéro de version du webservice
CodeOperation	1	1	String	Code de l'opération
CodeSiteOperation	1	1	String	Code site de l'opération (9 pour les webservices communs nationaux)
Disponible	1	1	Boolean	Indique si l'opération est disponible

5.14. *typeServiceVersion*

Il s'agit du type complexe correspondant à la description d'une version.

SEQUENCE				
BALISE	MULTIPLICITE		TYPE DE DONNEES	COMMENTAIRE
	MIN	MAX		
Service	1	1	Type complexe typeService	Information sur le webservice
Version	1	1	Type complexe typeVersion	Information sur la version