



Contrat d'interface Web-service Tipi

Historique de la fiche

09/03/2022	V0	PEI/ISC et PNM1	Version initiale
------------	----	-----------------	------------------

Table des matières

1. ECHANGE DE DONNÉES VIA WEB-SERVICE.....	2
1.1 FOURNISSEUR ⇒ TIPI.....	2
1.1.1. Prérequis.....	2
1.1.2. Requête HTTP attendu par le web service.....	2
1.1.2.1. Corps de la requête HTTP.....	2
1.1.2.2. En-tête du message.....	3
1.1.3. Réponse du web service.....	4
1.1.4. Etapes de validation de la requête web-service.....	4
1.1.4.1.1. Vérification du DN.....	5
1.1.4.1.2. Validation du message SOAP.....	5
1.1.4.1.3. Vérification du fournisseur.....	5
1.1.4.1.4. Vérification du réseau d'accès.....	6
1.1.4.1.5. Vérification de la présence de la balise </feedType>.....	6
2. ANNEXES AU CONTRAT D'INTERFACE.....	7
2.1 : ANNEXE 1 – EXEMPLES DE FICHIERS XML.....	7
2.1.1 : Données événementielles au format Datex 2.....	7
2.1.2 : Données trafics au format Datex 2.....	7
2.1.3 : Données de conditions de conduites hivernales au format xml.....	7
2.2 : ANNEXE 2 – POINTS DE VIGILANCE.....	7
2.1.1 : Installation du certificat SSL sous environnement Windows.....	7

1. ECHANGE DE DONNÉES VIA WEB-SERVICE

1.1 Fournisseur ⇒ Tipi

1.1.1. Prérequis

Le web-service Tipi est exposé sur internet de manière sécurisée, il ne nécessite pas l'utilisation d'un VPN mais oblige la présentation d'un certificat de type SSL client, qualifié RGS. Pour information, voici une liste de vendeurs de certificats qualifiés RGS et reconnus par le Ministère :

- Certinomis : <http://www.certinomis.fr/>
- Chambersign : <http://www.chambersign.fr/>
- Dhimyotis : <https://www.certigna.fr/>
- OpenTrust : <https://www.ssl-europa.com>
- CertEurope : <https://commande.certeurope.fr/>

Lors de la connexion du fournisseur sur le web-service exposé par TIPI, il est nécessaire que le client web-service utilise un certificat dont le DN est reconnu par TIPI (spécifié dans la fiche fournisseur par une expression régulière ou à défaut « .* »). Du côté tipi, il n'y a rien à faire : le certificat est présenté à l'entrée du Ministère qui le vérifie et qui route ensuite le message en http vers Tipi.

Remarques :

1- les illustrations de traduction Datex 2 ci-dessous sont orientées pour des fichiers Datex 2 événementiels. Toutefois, ce protocole d'échange données par Webservice s'applique aussi aux données Datex 2 relatives aux Trafic (fichier QTV et trafic) ainsi qu'au fichier xml de Tipi relatif à l'envoi des conditions de conduites hivernales. Des exemples de fichiers attendus pour les trois catégories de données sont joints à ce document (cf. annexe1).

2- le passage du transfert de données en webservice Tipi par rapport au transfert de données par dépôts de fichiers sur FTP ne modifie en rien le corps du message en Datex2 des données d'information routière. Seule l'ajout d'un bloc Datex 2 pour traduire l'enveloppe SOAP est ajouté autour du message Datex 2.

3- Tipi expose sur ses plateformes école et production un webservice Datex2 sécurisé et accessible par internet (en méthode push)

- pour Tipi école, voici l'url : <https://tipi-ws.site-ecole.din.developpement-durable.gouv.fr/PushService?wsdl>
- pour Tipi production, voici l'url : <https://tipi-ws.din.developpement-durable.gouv.fr/PushService?wsdl>

1.1.2. Requête HTTP attendu par le web service

1.1.2.1. Corps de la requête HTTP

Le corps de la requête HTTP faite au web-service doit être un message conforme à la norme SOAP version 1.1.

Le namespace attendu pour les balises définies par la norme SOAP doit être:

<http://schemas.xmlsoap.org/soap/envelope/>

La requête doit être conforme à la WSDL *Push.wsdl*.

Le message Datex 2 doit être contenu au niveau de la balise <soapenv:Body/>.

Deux types de messages Datex 2 peuvent être reçus :

- un message de données : Ces messages sont utilisés par les fournisseurs pour alimenter TIPI en données ;

- un message keepalive : Ces messages permettent à TIPI de s'assurer que le fournisseur est toujours opérationnel.

Exemple pour le corps de la requête HTTP d'une message de données:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns="http://datex2.eu/schema/2/2_0">
  <soapenv:Header/>
  <soapenv:Body>
    <d2LogicalModel xmlns="http://datex2.eu/schema/2/2_0" modelBaseVersion="2">
      <exchange>
        <supplierIdentification>
          <country>fr</country>
          <nationalIdentifier>@NATIONAL-IDENTIFIER@</nationalIdentifier>
        </supplierIdentification>
        <subscription>
          <operatingMode>operatingMode1</operatingMode>
          <subscriptionStartTime>2017-02-17T19:47:31.906+02:00</subscriptionStartTime>
          <subscriptionState>active</subscriptionState>
          <updateMethod>@UPDATE-METHOD@</updateMethod>
          <target>
            <address>not used</address>
            <protocol>not used</protocol>
          </target>
        </subscription>
      </exchange>
      <payloadPublication xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="SituationPublication" lang="fre">
        <feedType>@FEEDTYPE@</feedType>
        .....
        .....
      </payloadPublication>
    </d2LogicalModel>
  </soapenv:Body>
</soapenv:Envelope>
```

Exemple pour le corps de la requête HTTP d'une message keepalive:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns="http://datex2.eu/schema/2/2_0">
  <soapenv:Header/>
  <soapenv:Body>
    <d2LogicalModel modelBaseVersion="2" xmlns="http://datex2.eu/schema/2/2_0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <exchange>
        <keepAlive>true</keepAlive>
        <supplierIdentification>
          <country>fr</country>
          <nationalIdentifier>fnrbbe_ws</nationalIdentifier>
        </supplierIdentification>
      </exchange>
    </d2LogicalModel>
  </soapenv:Body>
</soapenv:Envelope>
```

1.1.2.2. En-tête du message

Les en-têtes additionnelles à ceux du protocole HTTP nécessaires pour le bon traitement du message entrant sont:

X-origine.UtilisateurCertificatDN	DN présenté par le fournisseur
X-origine.UtilisateurProvenance	Réseau de provenance du message SOAP

1.1.3. Réponse du web service

Conformément à la spécification Datex 2 pour les web services (fichier Push.wsdl), le message en sortie du web-service est toujours un message Datex 2 d'acquittement.

Le message renvoyé au fournisseur est de la forme :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns="http://datex2.eu/schema/2/2_0">
  <soapenv:Header/>
  <soapenv:Body>
    <d2LogicalModel xmlns="http://datex2.eu/schema/2/2_0" modelBaseVersion="2">
      <exchange>
        <clientIdentification>TIPI</clientIdentification>
        <response>@RESPONSE@</response>
        <supplierIdentification>
          <country>fr</country>
          <nationalIdentifier>@NATIONAL-IDENTIFIER@</nationalIdentifier>
        </supplierIdentification>
      </exchange>
    </d2LogicalModel>
  </soapenv:Body>
</soapenv:Envelope>
```

La balise *nationalIdentifier* est valorisée avec la valeur de la balise *nationalIdentifier* récupérée dans le message d'entrée du web-service. Si la valeur n'a pas pu être lue dans le message d'origine, la valeur *unknown* est renseignée.

La balise *response* est valorisée à :

- **requestDenied** : En cas d'erreur de validation du message soumis ;
- **acknowledge** : En cas de message validé et donc pris en compte par le web-service.

1.1.4. Etapes de validation de la requête web-service

Lors de la réception d'une requête sur le web-service, différentes étapes de validation sont réalisées avant de réaliser un retour au client du web-service.

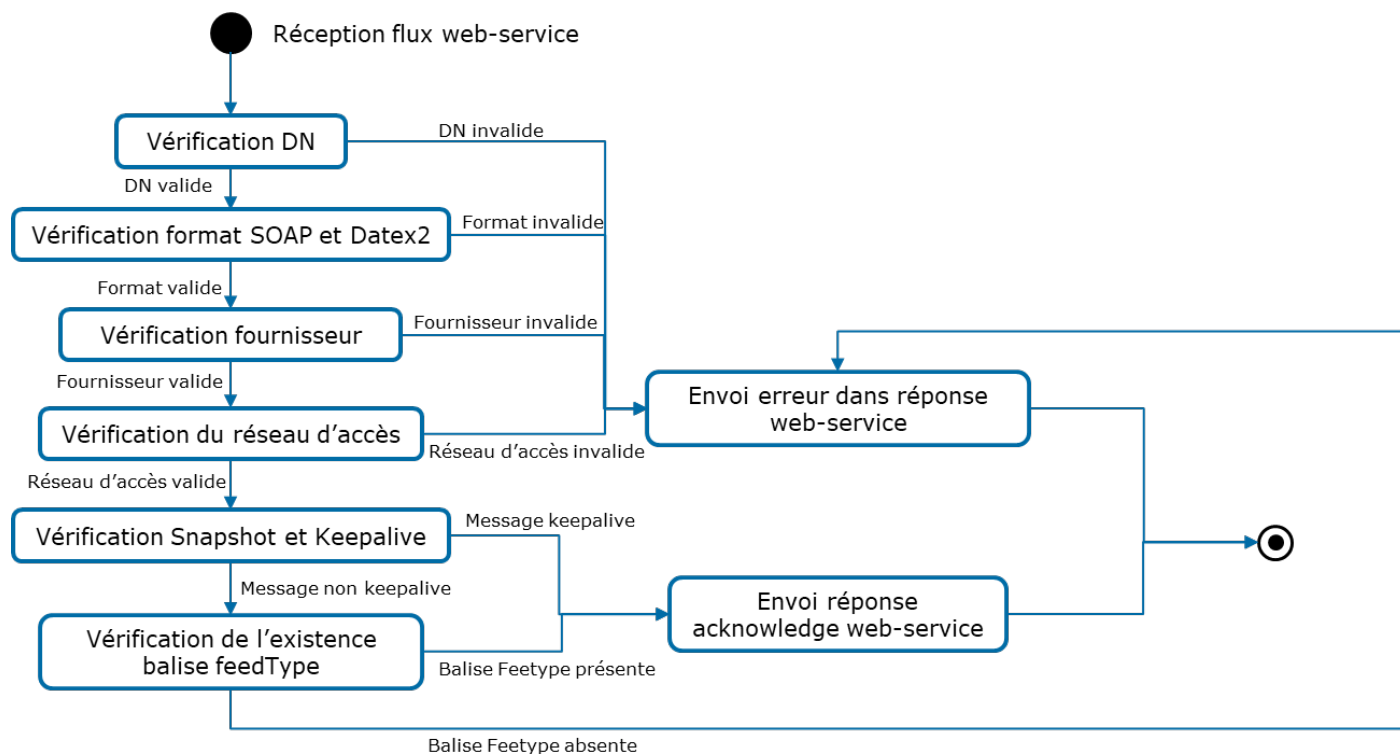


Figure 1: Etapes de validation du flux web-service reçu

1.1.4.1.1. Vérification du DN

La première étape qui est exécutée lors de la réception d'un message Datex 2, est le contrôle du DN transmis dans le certificat.

La recherche du DN n'est réalisé que pour les fournisseurs de type web service événementiel.

Le DN transmis par le fournisseur doit être sous le format :

```
/C=FR/ST=France/L=Saint-Medard-en-Jalles/O=CPII/OU=DOSO-ED/CN=ICITES-serveur1/
emailAddress=emaill1
```

Si le DN transmis ne correspond à aucun fournisseur paramétré dans TIPI, alors le fichier est rejeté.

1.1.4.1.2. Validation du message SOAP

La deuxième étape est la validation du message SOAP entrant. Il doit contenir une enveloppe SOAP 1.1 valide et un message Datex 2 valide.

Si le message SOAP n'est pas valide, le message est rejeté.

1.1.4.1.3. Vérification du fournisseur

Une fois le message validé, on vérifie que le fournisseur renseigné dans le message Datex 2 correspond à un des fournisseurs ayant eu le bon DN.

Si aucun des fournisseurs ayant le bon DN ne correspond à celui renseigné dans le message Datex 2, le message reçu est rejeté.

L'objectif est d'assurer une cohérence entre le(s) fournisseur(s) identifié(s) grâce au DN et le fournisseur renseigné dans le message Datex 2 étant donné que plusieurs fournisseurs ont pu être retourné à l'étape de vérification du DN car plusieurs peuvent utiliser le même DN.

1.1.4.1.4.Vérification du réseau d'accès

Le réseau de provenance des données est récupéré dans l'en-tête HTTP du message SOAP. Une vérification est réalisée afin de savoir si le fournisseur est autorisé à utiliser le réseau (cette configuration est interne à TIPI).

Si le fournisseur passe par un réseau non autorisé, le message est rejeté.

1.1.4.1.5.Vérification de la présence de la balise </feedType>

Dans le message Datex 2, si le contenu de la balise *updateMethod* est différent de « *snapshot* » et que le message n'est pas « *keepAlive* », le *feedType* est obligatoire.

Si le *feedType* est absent, le message reçu est rejeté.

2.1 : Annexe 1 – exemples de fichiers xml

2.1.1 : Données évènementielles au format Datex 2

Le message ci-joint (**soap_putdatex2_fichier_EVE.xml**) se veut simplement montrer un exemple concret de la traduction attendue par Tipi de l'enveloppe SOAP. Il ne correspond en aucun cas à une situation réaliste.

2.1.2 : Données trafics au format Datex 2

Le message ci-joint (**soap_putdatex2_fichier_QTV.xml**, **soap_putdatex2_fichier_trafic.xml**) se veut simplement montrer un exemple concret de la traduction attendue par Tipi de l'enveloppe SOAP. Il ne correspond en aucun cas à une situation réaliste.

2.1.3 : Données de conditions de conduites hivernales au format xml

Le message ci-joint (**soap_putfichierVH.xml**) se veut simplement montrer un exemple concret de la traduction attendue par Tipi de l'enveloppe SOAP. Il ne correspond en aucun cas à une situation réaliste.

2.2 : Annexe 2 – points de vigilance

2.1.1 : Installation du certificat SSL sous environnement Windows

Le protocole SSL est basé sur le cryptage des données à l'aide d'une clé privée et d'une clé publique.

Clé privée : Une clé privée est créée en convertissant une portion de texte générée automatiquement en un fichier clé à l'aide d'un algorithme mathématique, ce qui lui donne une valeur unique. Ce fichier clé est utilisé pour générer un CSR, et ensuite pour créer un certificat SSL. Dans ce processus de CSR, la clé publique est également créée. La clé privée doit impérativement rester secrète. Cette clé permet de décrypter les données cryptées et de crypter les messages signés par un certificat.

Clé publique : La clé publique est créée lors de la génération d'un CSR et peut être distribuée au public. Par exemple, une clé publique est utilisée pour crypter des informations que seul le propriétaire de la clé privée est autorisé à recevoir. La combinaison unique de la clé publique et de la clé privée peut alors décrypter ces données. Une clé publique peut également être utilisée pour vérifier qu'un message a été envoyé par le propriétaire de la clé privée.

Pour que le transfert des données du fournisseur par Webservice Tipi puisse se faire correctement, il est nécessaire que la clé privée du certificat SSL soit configurée sur le serveur du système d'information du fournisseur.

Sous environnement windows, il existe deux méthodes pour installer le certificat SSL : double-cliquer sur le fichier.pfx ou en l'installant par importation via la console mmc.exe sans oublier l'étape indispensable de la configuration de la clé privée.

Il est recommandé d'installer le certificat SSL par importation via la console mmc.exe car le double-cliquage sur le fichier .pfx n'intègre pas la configuration de la clé privée.

Le lien internet ci-contre illustre les pièges à éviter et les bonnes pratiques à suivre pour installer le certificat SSL : <https://www.it-swarm-fr.com/fr/c%23/system.security.cryptography.cryptographicexception-le-jeu-de-cles-nexiste-pas/1067739411/>