Documentation Web service SOAP

Bienvenue dans la documentation des WebServices de FormPublisher.

1. Webservice

1.1. Concept

L'appel d'un webservice se produit par l'usage de la balise WebService dans une page du formulaire.

L'usage de la balise WebService va permettre à l'auteur de renseigner l'url conduisant à un WSDL et de sélectionner une opération que l'on souhaite réalisée.

Seules les webservices SOAP ayant en requête et en réponse une enveloppe SOAP et utilisant le protocole HTTP peuvent être accédés.

Cet accès doit se produire alors que les données nécessaires au webservice sont présentes dans le contexte de la session du formulaire.

Les données nécessaire à l'accès à un webservice sont celles décrites au sein d'un service FormPublisher.

Un **service** FormPublisher se décrit à l'aide de trois fichiers :

- Un fichier <id>.**properties**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties version="1.0">
<entry key="Method">POST</entry>
<entry key="Transport">http://ws.cdyne.com/ip2geo/ip2geo.asmx</entry>
<entry key="Accept-Encoding">gzip,deflate</entry>
<entry key="Content-Type">text/xml;charset=UTF-8</entry>
<entry key="SOAPAction">http://ws.cdyne.com/ResolveIP</entry>
<entry key="User-Agent">Jakarta Commons-HttpClient/3.1</entry>
<entry key="Host">ws.cdyne.com</entry>
<entry key="Content-Length"></entry>
</properties>
```

- un fichier <id>.**template**

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ws="http://ws.cdyne.com/">
<soapenv:Header/>
<soapenv:Body>
<ws:ResolveIP>
<ws:ipAddress><?data $( ipAddress )?></ws:ipAddress>
<ws:licenseKey><?data $( licenseKey )?></ws:licenseKey>
</ws:ResolveIP>
</soapenv:Body>
</soapenv:Envelope>
```

- un fichier <id>.**mapping**

  <?xml version="1.0" encoding="utf-8"?>
  <soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd=" http://www.w3.org/2001/XMLSchema ">
  <soap:Body>
  <ResolveIPResponse xmlns=" http://ws.cdyne.com/ ">
  <ResolveIPResult>
  <City> ***{city}*** *</City>*
  <StateProvince>C1</StateProvince>
  <Country> ***{localisation|pays|nom}*** *</Country>*
  <Organization />
  <Latitude>48.5833</Latitude>
  <Longitude>7.75</Longitude>
  <AreaCode>0</AreaCode>
  <TimeZone/>
  <HasDaylightSavings>false</HasDaylightSavings>
  <Certainty>90</Certainty>
  <RegionName />
  <CountryCode> ***{localisation|pays|code}*** *</CountryCode>*
  </ResolveIPResult>
  </ResolveIPResponse>
  </soap:Body>
  </soap:Envelope>

Ici les données nécessaires sont ipAddress et licensekey.

Le service va permettre à partir d'un <id> d'accéder à ces structures (**properties**, **template** et **mapping**) lors des traitements par la partie serveur du formulaire.

Dans le contexte d'un webservice, elles sont générées à l'aide de FormPublisher Studio mais peuvent également être créées manuellement si cela est nécessaire.

Les fichier **properties** et **template** sont instanciés avant d'être utilisés.

Autrement dit, les expressions JIL qu'ils pourraient contenir sont exécutées. Les contenus produits à l'issue de cette instanciation sont ensuite exploités pour appeler le webservice.

Le fichier **mapping** préfigure un exemple de réponse que le webservice pourrait produire.

Grâce à l'intégration de marqueur dans ce fichier, il est possible de déterminer les éléments de la réponse que l'on souhaite conserver (ils sont exprimés sous la forme *{a|b|...}*). Dans un premier temps, seuls les éléments non itératifs sont pris en compte.

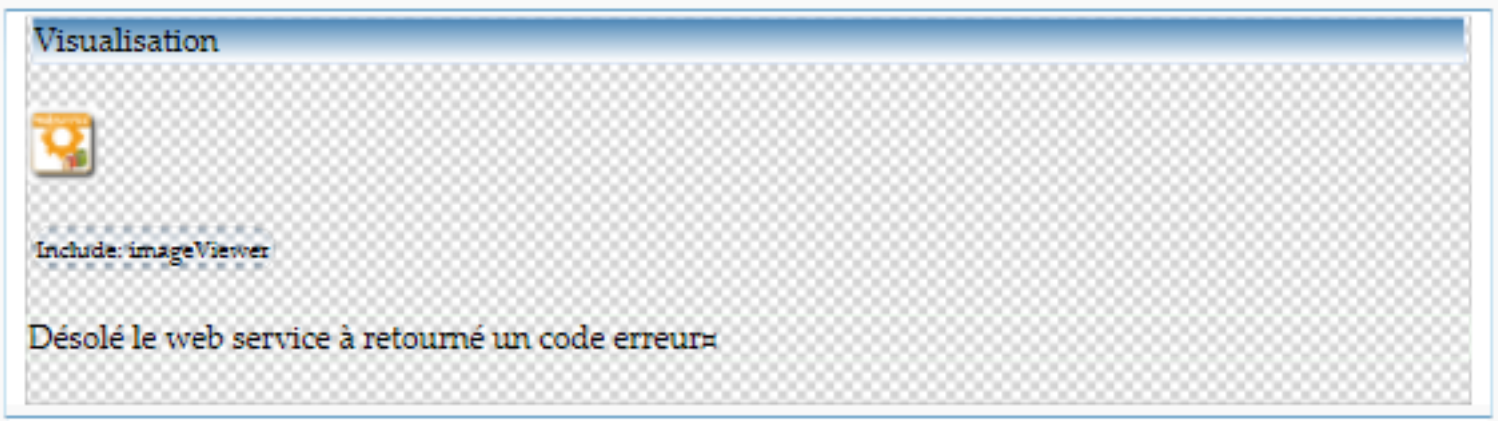
Les marqueurs ici présents sont les suivants *{city}, {localisation|pays|nom}* et *{localisation|pays|code*}.

L'usage du symbole *[]* permettra de déterminer une structure itérative par la suite.

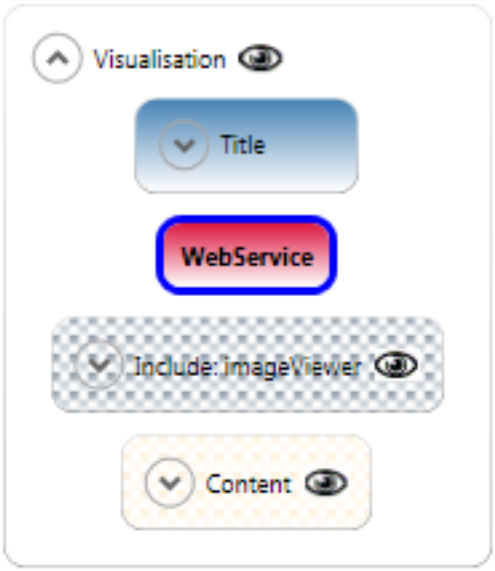
1.2. Mise en oeuvre

La mise en oeuvre des webservices dans FormPublisher s'opère par l'usage de la balise WebService.

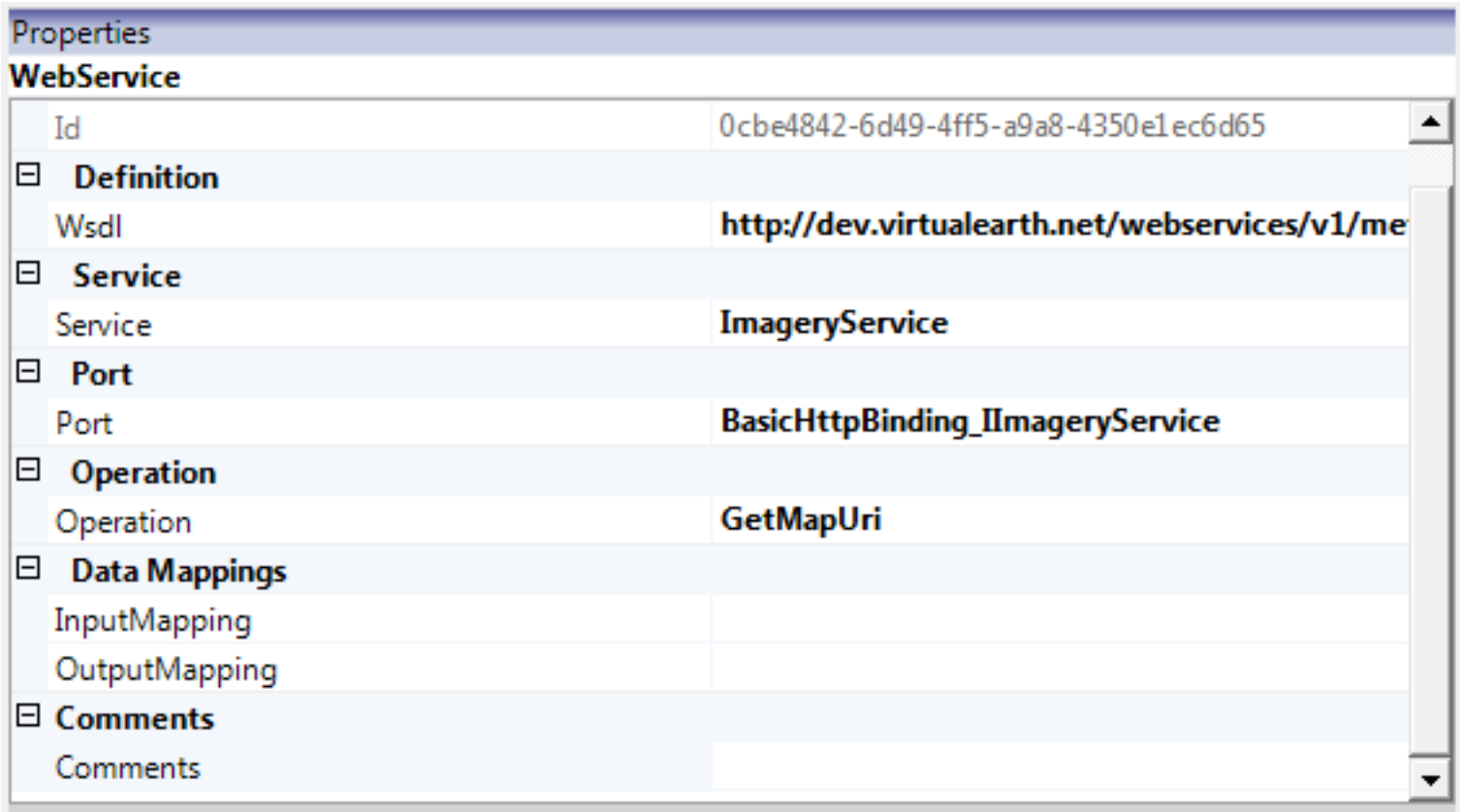
La balise WebService est rendue de la manière suivante en mode Edit :



La balise WebService est rendue de la manière suivante en mode Flow :



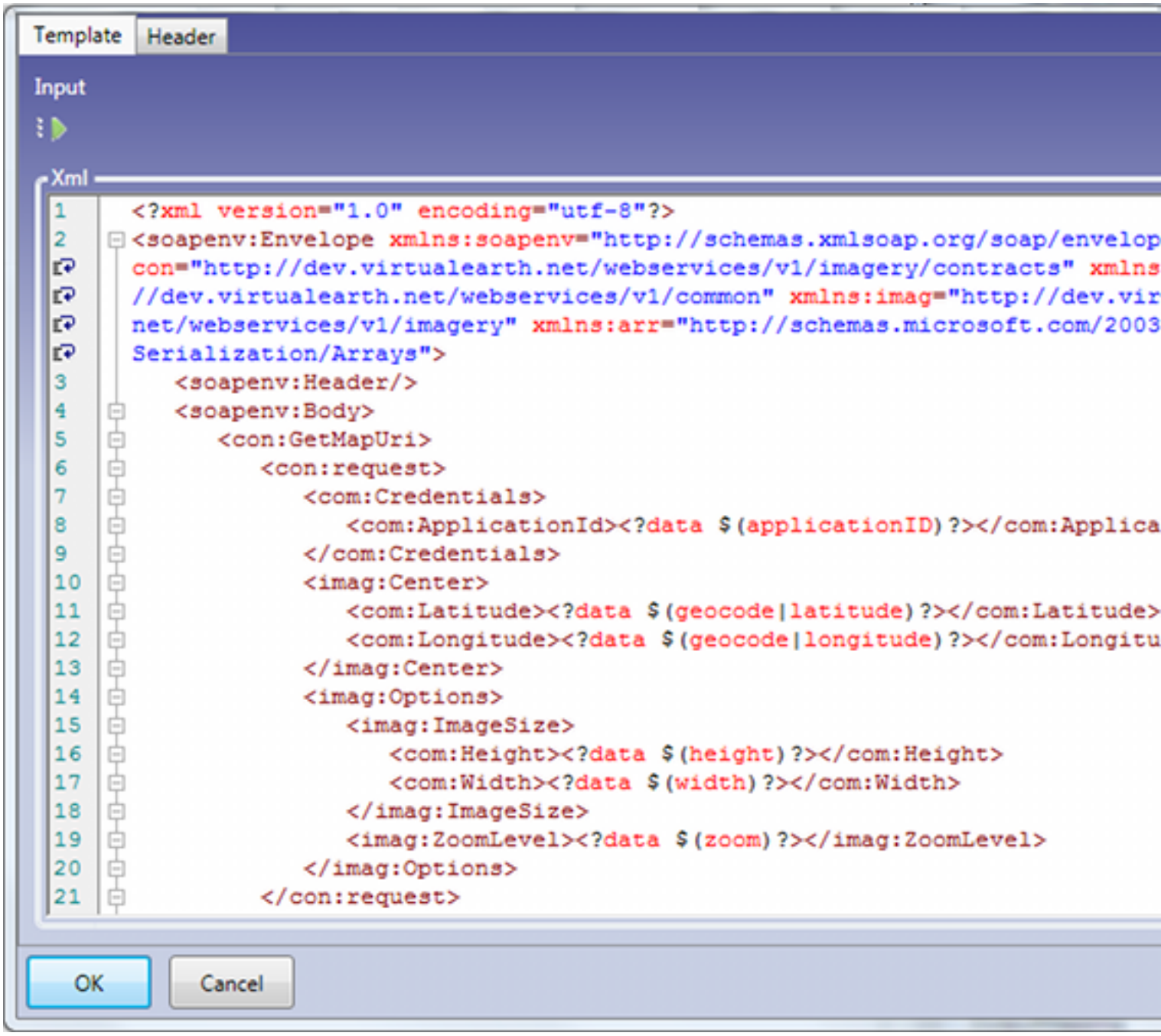
Enfin, les propriétés associées sont les suivantes :



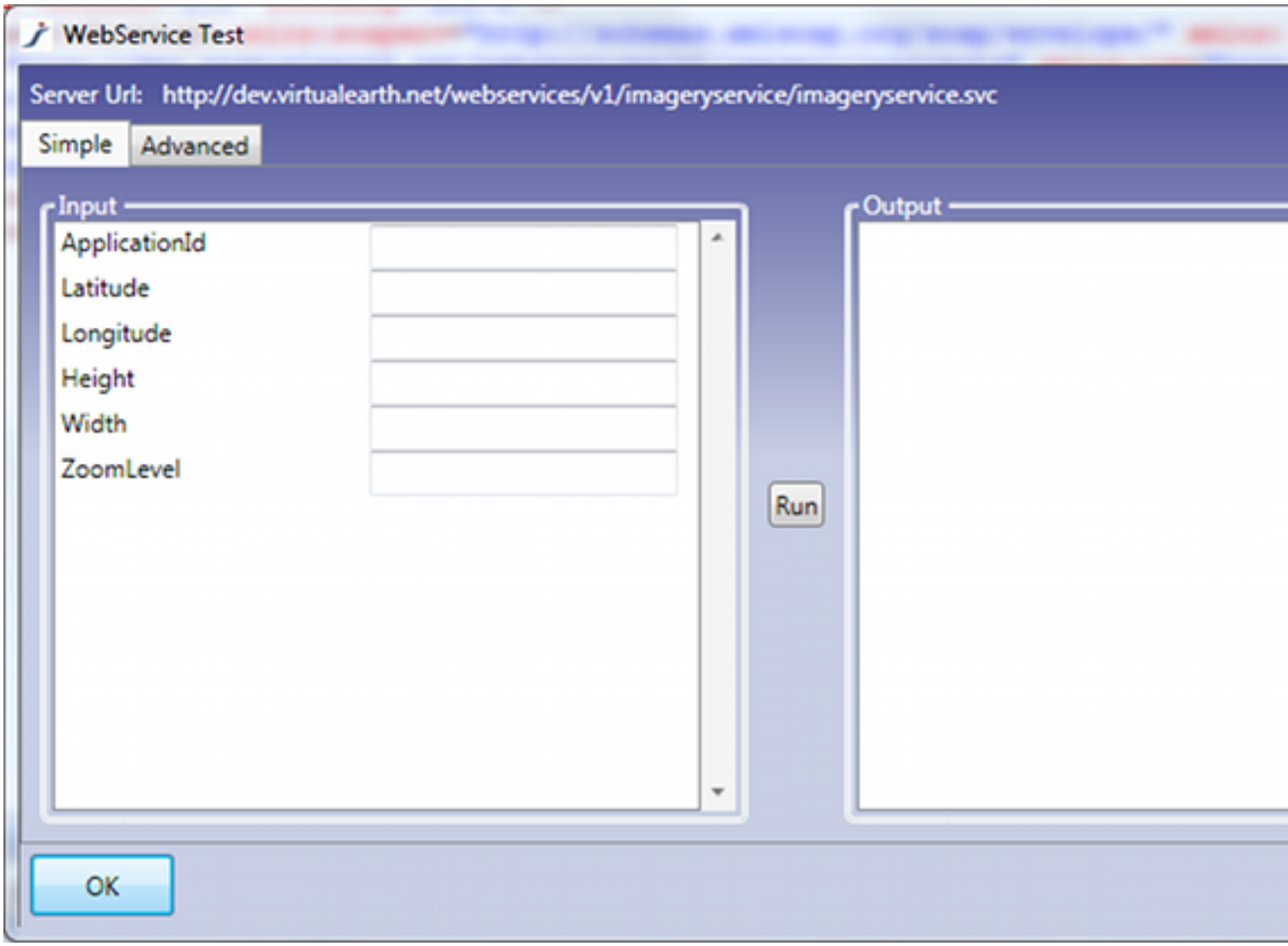
L'on retrouve ici les propriétés décrites précédemment, ainsi que InputMapping et OutputMapping qui permettent de renseigner **template** et **properties** pour le premier choix et **mapping** pour le second.

Ceci s'opère par le biais d'assistants.

L'assistant InputMapping :



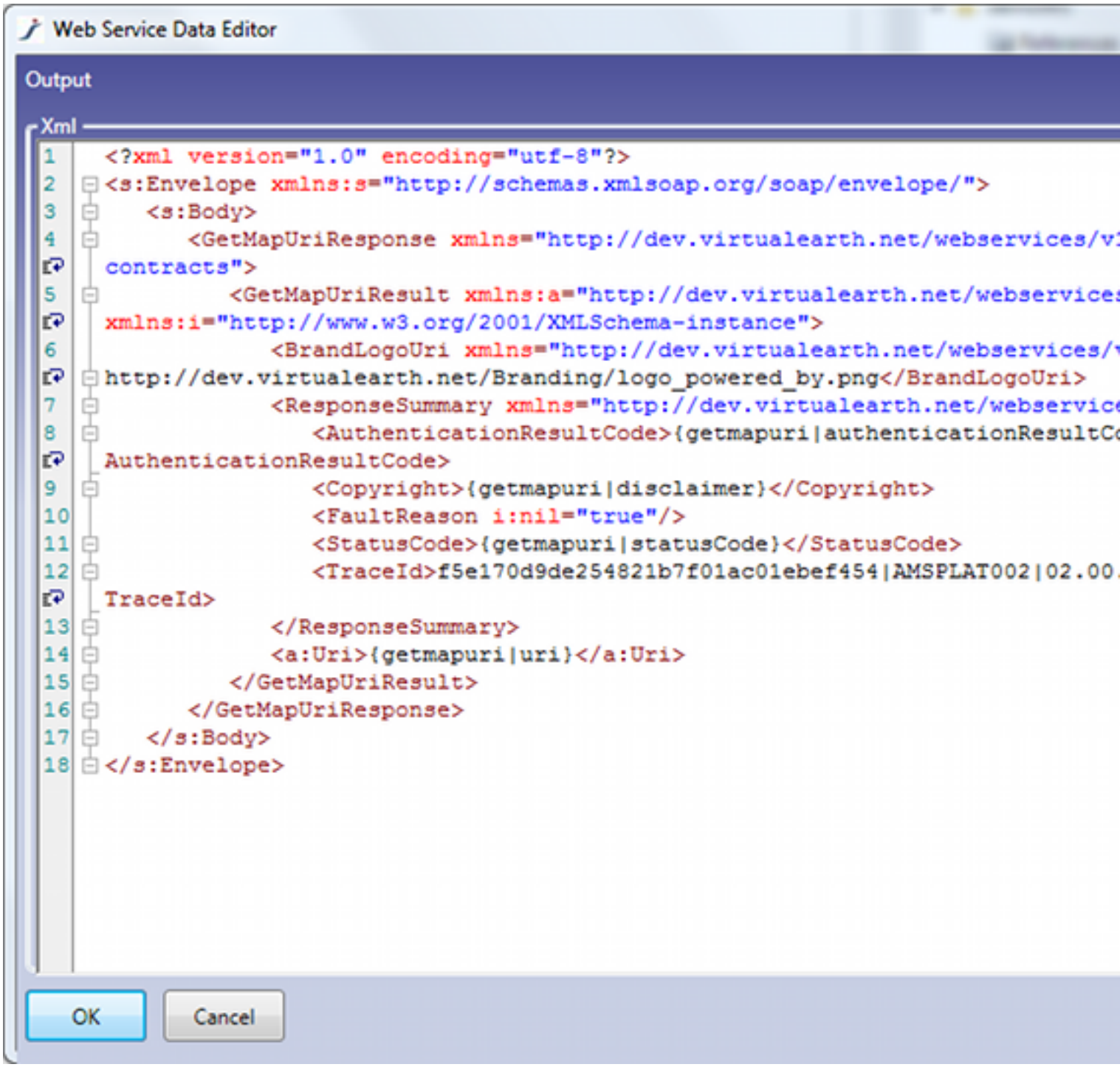
Son environnement de simulation, démarré par la flèche verte (en haut à gauche) :



On remarquera que l'outil de simulation indique les données à renseigner avant d'effectuer l'appel. Le mode Advanced affichera directement les enveloppes soap envoyées et reçues lors de l'échange aller/retour avec le WebService de type [1] **Document Literal Wrapped**.

L'assistant OutputMapping :

[1] Il existe plusieurs catégories de WebService de style RPC ou Document.
FormPublisher supporte uniquement le style Document Literal wrapped avec un échange aller/retour, c'est à dire émission d'un message SOAP par POST HTTP et réponse d'un message SOAP.



Celle-ci introduit un comportement spécifique

L'usage de WebService introduit un comportement spécifique à adopter lors de cas extrêmes comme la non réponse par le WebService (NoAnswer) et l'absence de données extraites de la réponse (NoData).

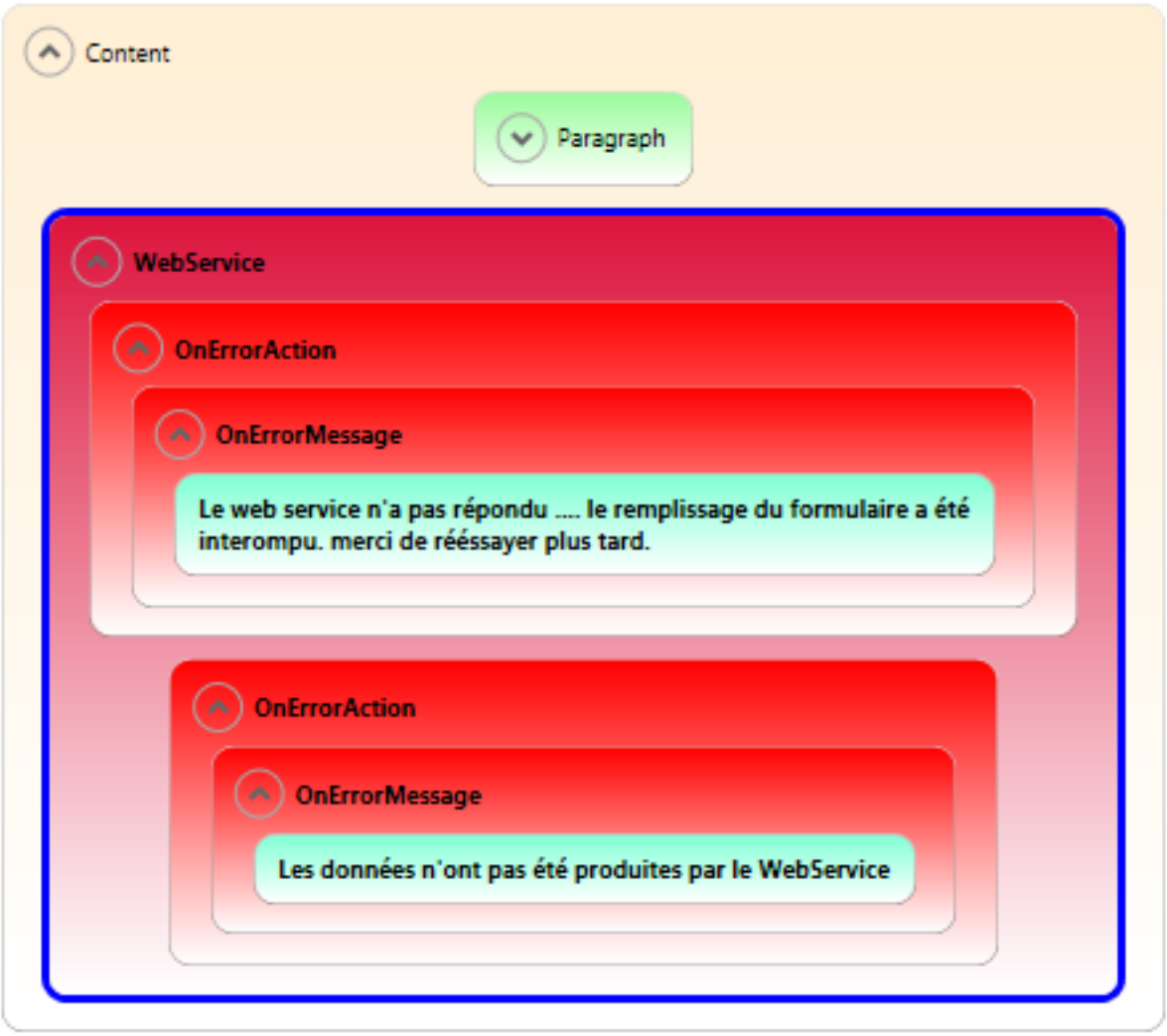
Par défaut, le comportement est le suivant :

- NoAnswer : cela va générer une erreur de code ERR042 qui sera propagée jusqu'à la page error.xml. La session sera close et les données de l'utilisateur seront perdues.
- NoData : est ignoré par FormPublisher.

Il y a donc quatre niveaux de réponse aux cas extrêmes :

1. Fatal : session close
2. Error : session conservée mais il est conseillé à l'utilisateur d'arrêter de travailler et de sauvegarder ses données
3. Warning : session conservée mais l'utilisateur est averti du dysfonctionnement.
4. Ignore : aucune information n'est indiquée.

L'auteur peut influer sur le comportement par défaut en utilisant OnErrorAction sur la balise WebService associée :



Ici on a associé deux OnErrorAction à ce WebService.

La balise OnErrorMessage permet d'indiquer le message qui sera affiché à l'utilisateur. Si elle est manquante, le message technique associé au cas extrême rencontré sera affichée.

Le cas extrême et le niveau de réponse sont renseignés sur la balise OnErrorAction par le biais de la fenêtre Propeties associée.