

OVIDENTIA

Ovidentia Web Services



Version Documentation	Version <i>OVIDENTIA</i> / Module	Date
1.0	5.8.6	
1.1	6.0.3	26/10/2006
1.2	6.0.3	30/10/2006
1.3	6.4.2	20/08/2007

Table des matières

1 - Introduction :	4
2 - Les Services Web d'Ovidentia :	4
2.1 - Activer les Services Web d'Ovidentia	4
2.2 - login	4
2.3 - babSoapOvml	5
2.4 - babSoapOvmlContent	5
2.5 - babSoapOvmlFile	5
3 - OvML et les Services Web	6
3.1 - Le langage OvML	6
3.1.1 - Les variables	6
3.1.2 - Les tableaux	6
3.1.3 - Les containers	8
3.1.3.1 - Appel d'un service web externe par OvML (méthode 1)	8
3.1.3.2 - Appel d'un service web externe par OvML (méthode 2)	9
4 - Exemples	10
4.1 - Appel d'un service web externe	10
4.2 - Authentification sur un site Ovidentia	11
4.3 - Exécution d'un container sur un site Ovidentia	12

1 - Introduction :

Ovidentia intègre une API destinée à faciliter l'utilisation des Services Web. Les administrateurs peuvent intégrer dans leurs sites des Services Web externes ou fournis par d'autres sites Ovidentia.

Ovidentia utilise les standards SOAP et WSDL permettant ainsi au webmaster ou au développeur d'utiliser son environnement et son langage favori tels que Perl, Java ou Visual Studio .NET, ...

- **SOAP** est un acronyme informatique signifiant Simple Object Access Protocol. C'est un protocole recommandé par le W3C dans la transmission de messages entre objets distants (Autorisation d'un objet à exécuter une méthode située sur une autre machine).
- **WSDL** est un acronyme informatique signifiant Web Services Description Language. Il s'agit d'une normalisation regroupant la description des éléments permettant de mettre en place l'accès à un service réseau (Services Web).

Ce document explique comment utiliser l'API SOAP d'Ovidentia ou les Services Web fournis par Ovidentia.

L'interrogation d'un service web externe, à partir d'un site d'Ovidentia, se fait via le langage OvML propre à Ovidentia.

Dans la suite, vous apprendrez comment échanger des informations entre différents sites d'Ovidentia. Par exemple, afficher en première page d'un site Ovidentia, les articles dernièrement publiés dans un autre site Ovidentia.

2 - Les Services Web d'Ovidentia :

Ces fonctions sont disponibles dans Ovidentia et accessibles pour des applications clientes externes.

2.1 - Activer les Services Web d'Ovidentia

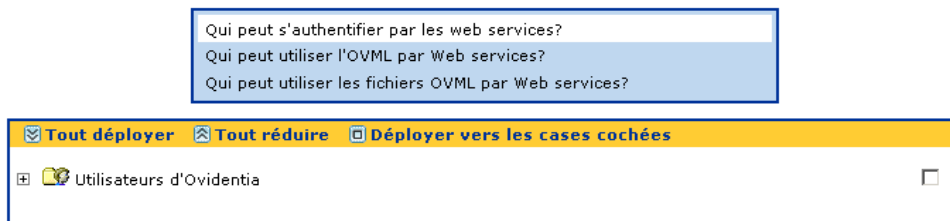
Avant toute utilisation, les services doivent être activés. Pour cela, une interface a été ajoutée dans les options du site (lien Sites en section Administration).

3 options sont disponibles :

- Qui peut s'authentifier par les Web services ?
- Qui peut utiliser l'OvML par Web services ?
- Qui peut utiliser les fichiers OVML par Web services ?


--> Le droit « Qui peut s'authentifier par les Web services ? » doit être activé afin d'utiliser l'OvML par Web Services et les fichiers OVML par Web services.

Web services: Ovidentia



<input type="checkbox"/> Qui peut s'authentifier par les web services?
<input type="checkbox"/> Qui peut utiliser l'OvML par Web services?
<input type="checkbox"/> Qui peut utiliser les fichiers OVML par Web services?

Tout déployer Tout réduire Déployer vers les cases cochées

<input checked="" type="checkbox"/>  Utilisateurs d'Ovidentia	<input type="checkbox"/>
--	--------------------------

2.2 - login

Ce service peut être utilisé pour s'authentifier sur un site Ovidentia.

Arguments:

- nickname: identifiant de l'utilisateur
- password: mot de passe de l'utilisateur

Réponse:

- id: id de la session PHP si l'authentification peut s'effectuer sinon 0 (zéro)
- error: libellé de l'erreur si l'authentification échoue.

A partir d'un site Ovidentia l'appel à ce service peut se faire via le container OCSOap (voir ci-dessous).

2.3 - babSoapOvml

Ce service peut être utilisé pour interroger des containers OvML sur un site Ovidentia distant.

Arguments:

- container: nom du container sans le préfixe OC (Exemples : RecentArticles ou Files)
- args: attributs du container sous la forme d'un tableau multidimensionnel de tableaux associatifs:
args[n] = array('name'=>param1, 'value'=>valeur de param1)

Réponse:

- chaîne contenant le résultat du container.

A partir d'un site Ovidentia l'appel à ce service peut se faire via le container OCSOap (voir ci-dessous).

2.4 - babSoapOvmlContent

Ce service peut être utilisé pour exécuter un contenu OvML sur un site Ovidentia distant.

Arguments:

- content: chaîne contenant le code OvML
- args: un tableau multidimensionnel de tableaux associatifs:
args[n] = array('name'=>param1, 'value'=>valeur de param1)

Réponses:

- Chaîne contenant le résultat de l'interprétation du code OvML.

A partir d'un site Ovidentia l'appel à ce service peut se faire via le container OCSOap (voir ci-dessous).

2.5 - babSoapOvmlFile

Ce service peut être utilisé pour exécuter un fichier OvML sur un site Ovidentia distant.

Arguments:

- file: nom du fichier OvML et son chemin (Exemple : mescodes/exemple1.ovml)
- args: un tableau multidimensionnel de tableaux associatifs:
args[n] = array('name'=>param1, value=>valeur de param1)

Réponses:

- Chaîne contenant le résultat du fichier OvML.

A partir d'un site Ovidentia l'appel à ce service peut se faire via le container OCSOap (voir ci-dessous).

3 - OvML et les Services Web

3.1 - Le langage OvML

Afin de faciliter l'utilisation des Services Web, l'OvML a été amélioré pour permettre la manipulation des tableaux et des types de données.

3.1.1 - Les variables

Le type d'une variable peut être précisé ou forcé en utilisant la syntaxe du « cast » :

Exemple :

```
<OFPutVar name="myvar" value="(int)10">
```

Liste des types :

<i>Cast</i>	<i>Type</i>
int ou integer	Entiers
float, double, real	Nombres décimaux (un point comme séparateur)
bool, boolean	Booléens (true, false)
var, variable	Force Ovidentia à considérer la chaîne comme une variable existante (voir exemple ci-dessous)

Exemple :

```
<OFPutVar name="varA" value="(int)10">  
<OFPutVar name="varB" value="varA">  
<OFPutVar name="varC" value="(var)varA">  
Valeur de varA = <OVvarA><br>  
Valeur de varB = <OVvarB><br>  
Valeur de varC = <OVvarC><br>
```

Ceci affichera :

```
Valeur de varA = 10  
Valeur de varB = varA  
Valeur de varC = 10
```

3.1.2 - Les tableaux

OFPutArray:

Permet de créer un tableau en indiquant manuellement ces entrées et leurs valeurs.

```
<OFPutArray name="montableau" entree1="valeur1" entree2="valeur2" ...>
```

Ceci créera un tableau associatif nommé « montableau » où « entree1 » et « entree2 » sont les index et « valeur1 » et « valeur2 » leurs valeurs respectives.

Pour afficher une valeur du tableau, il faut utiliser cette syntaxe :

```
<OVmontableau[entree1]>
```

Dans cet exemple, on affichera « valeur1 ».

OFPutSoapArray:

Permet de créer un tableau directement utilisable par Soap.

```
<OFPutSoapArray name="montableau" entree1="valeur1" entree2="valeur2" ....>
```

OCOvmlArray:

Ce container permet de lister les entrées d'un tableau.

```
<OCOvmlArray name="montableau2" value="montableau">...</OCOvmlArray>
```

Attributs:

name:

- nom du tableau qui va manipuler l'autre tableau. Ici « montableau2 » manipule « montableau »

value:

- variable Ovml contenant le tableau

Variables:

OV.[nom du tableau].Key	Nom de l'entrée
OV.[nom du tableau].Value	Valeur de l'entrée

Exemple :

```
<OFPutVar name="valeur1" value="(float)300.65">
<OFPutArray name="montableau" A="(int)1" B="(int)2" C="(int)3" D="(var)valeur1">

<OCOvmlArray name="montableau2" value="montableau">
  <OVmontableau2Key>=<OVmontableau2Value><br>
</OCOvmlArray>
```

Ceci affichera :

```
A=1
B=2
C=3
D=300.65
```

OCOvmlArrayFields:

Ce container permet de lister les entrées d'un tableau d'une autre façon que le container OCOvmlArray.

```
<OCOvmlArrayFields name="montableau2" value="montableau">...</OCOvmlArrayFields>
```

Attributs:

name:

- nom du tableau qui va manipuler l'autre tableau. Ici « montableau2 » manipule « montableau »

value:

- variable Ovml contenant le tableau

Variables:

OVxxxxx	xxxxx étant l'entrée dans le tableau
---------	--------------------------------------

Exemple :

```
<OFPutVar name="valeur1" value="(float)300.65">
<OFPutArray name="montableau" A="(int)1" B="(int)2" C="(int)3" D="(var)valeur1">

<OCVmlArrayFields name="montableau2" value="montableau">
  <OVA><br>
  <OVB><br>
  <OVC><br>
  <OVD><br>
</OCVmlArrayFields>
```

Ceci affichera :

```
1
2
3
300.65
```

3.1.3 - Les containers

3.1.3.1 - Appel d'un service web externe par OvML (méthode 1)

Le container OCSOap permet d'interroger un service web externe : un autre site Ovidentia ou un fournisseur. Remarque : il est possible d'interroger un service Web externe si la connexion Web passe par un proxy (authentifié ou pas).

```
<OCSoap apiserver="" apicall="" apinamespace="" debug="" proxyhost=""
proxyport="" proxyusername="" proxypassword="" ...>...</OCSoap>
```

Attributs:

apiserver:

- Url du service web.

apicall:

- La méthode ou le service appelé

apinamespace:

Optionnel

- Le namespace (espace de nommage)

debug:

Optionnel

- Si égal à 1, Ovidentia génère les variables de debug: babSoapError, babSoapResponse et babSoapRequest que l'on peut afficher en utilisant la syntaxe <OvbabSoap...>

proxyhost:

Optionnel

- Nom de domaine ou adresse IP du proxy
- Si on n'utilise pas de proxy, ce paramètre ne doit pas être renseigné

proxyport:

Optionnel

- Numéro de port du proxy
- Si on n'utilise pas de proxy, ce paramètre ne doit pas être renseigné

proxyusername:

Optionnel

- Identifiant du proxy si le proxy demande une authentification
- Si on n'utilise pas de proxy authentifié, ce paramètre ne doit pas être renseigné

proxypassword:

Optionnel

- Mot de passe du proxy si le proxy demande une authentification
- Si on n'utilise pas de proxy authentifié, ce paramètre ne doit pas être renseigné

Si l'appel au service réussit, le résultat sera fourni dans la variable SoapResult.

Le résultat peut être un tableau et dans ce cas il faut, en s'appuyant sur la documentation du fournisseur de service, utiliser les containers OvML <OCOvmlArray> et <OCOvmlArrayFields> pour le manipuler.

Appel d'un service web Ovidentia :

L'attribut apiserver doit prendre la valeur : <http://monsite/index.php?tg=omlsoap>

L'attribut apicall correspond au service appelé : login, babSoapOvml...

Appel d'un service web Ovidentia en tant qu'utilisateur authentifié :

Il est possible d'utiliser un service Ovidentia en étant authentifié en ajoutant un paramètre :

<http://monsite/index.php?tg=omlsoap&WSSESSIONID=masession>

L'authentification sur le site cible se réalise avec le service « login ».

L'appel du service renvoie un identifiant de session qui pourra être réutilisé par la suite avec les services Ovidentia comme babSoapOvml (voir paramètre WSSESSIONID).

3.1.3.2 - Appel d'un service web externe par OvML (méthode 2)

Afin de faciliter l'utilisation des Web services OVML d'un site Ovidentia à un autre, le container OCOvmlSoap a été mis en place. La syntaxe d'utilisation équivaut à celle des containers OVML existants.

Le container OCOvmlSoap permet d'interroger par service web un autre site Ovidentia.

Remarque : il est possible d'interroger un service Web externe si la connexion Web passe par un proxy (authentifié ou pas).

```
<OCOvmlSoap apiserver="" container="" proxyhost="" proxyport="" proxyusername=""  
proxypassword="" ...>...</OCOvmlSoap>
```

Attributs:

apiserver:

- Url du service web.

apicall:

- La méthode ou le service appelé

debug:

Optionnel

- Si égal à 1, Ovidentia génère les variables de debug: babSoapError, babSoapResponse et babSoapRequest que l'on peut afficher en utilisant la syntaxe <OVbabSoap...>

proxyhost:

Optionnel

- Nom de domaine ou adresse IP du proxy
- Si on n'utilise pas de proxy, ce paramètre ne doit pas être renseigné

proxyport:

Optionnel

- Numéro de port du proxy
- Si on n'utilise pas de proxy, ce paramètre ne doit pas être renseigné

proxyusername:

Optionnel

- Identifiant du proxy si le proxy demande une authentification
- Si on n'utilise pas de proxy authentifié, ce paramètre ne doit pas être renseigné

proxypassword:

Optionnel

- Mot de passe du proxy si le proxy demande une authentification
- Si on n'utilise pas de proxy authentifié, ce paramètre ne doit pas être renseigné

Si l'appel au service réussit, le résultat sera fourni dans la variable SoapResult.

Le résultat peut être un tableau et dans ce cas il faut, en s'appuyant sur la documentation du fournisseur de service, utiliser les containers OvML <OCOvmlArray> et <OCOvmlArrayFields> pour le manipuler.

Exemple :

```
<OCOvmlSoap apiserver="http://monsite/index.php?tg=omlsoap"
container="RecentArticles" last="10">
    <OVArticleTitle><br>
</OCOvmlSoap>
```

4 - Exemples

4.1 - Appel d'un service web externe

- Appel de l'API de recherche Google avec le mot ovidentia

```
<OCSoap apiserver="http://api.google.com/search/beta2" apicall="doGoogleSearch"
apinamespace="urn:GoogleSearch" key="xxxxxxxxxxxxxxxxxxxx" q="ovidentia"
start="(int)0" maxResults="(int)10", filter="(bool)0" restrict=""
safeSearch="(bool)0" lr="" ie="" oe="">
    <OCOvmlArray name="resultElements" value="SoapResult[resultElements]">
        <OCOvmlArrayFields name="results" value="resultElementsValue">
            <a href="<OVURL>"><OVtitle></a><br>
```

```
        </OCOvmlArrayFields>
    </OCOvmlArray>
</OCSoap>
```

Remarque : remplacez la clé xxxxxxxxxxxxxxxx par la vôtre.
Le numéro de licence peut s'obtenir depuis cette url :
<http://api.google.com/createkey>
--> Google accepte jusqu'à 1000 requêtes par jour par numéro de licence.
Plus de détails sur le site Google :
<http://code.google.com/apis/soapsearch/index.html>

- Interrogation d'une base de données sur les compagnies aériennes

```
<OCSoap apiserver="http://www.SoapClient.com/xml/SQLDataSoap.WSDL"
apicall="ProcessSRL" apinamespace="" SRLFile="/xml/EDGAR.SRI"
RequestName="edgar" key="AOL">
    <OVSoapResult>
</OCSoap>
```

- Taux d'échange entre différentes monnaies

```
<OCSoap apiserver="http://services.xmethods.net:80/soap" apicall="getRate"
apinamespace="urn:xmethods-CurrencyExchange" country1="FRANCE"
country2="ALGERIA">
    <OVSoapResult>
</OCSoap>
```

4.2 - Authentification sur un site Ovidentia

Cet exemple montre comment à partir d'un site Ovidentia on peut s'authentifier sur un autre site Ovidentia.

```
<OCSoap apiserver="http://autresite/index.php?tg=omlsoap" apicall="login"
nickname="xxxxxxx" password="xxxxxxx">
    <OCOvmlArrayFields name="result" value="SoapResult">
        <OFSetCookie name="wssessionid" value="<Ovid">">
        <OFPutVar name="wssessionid" value="<Ovid">">
        <OFPutVar name="wserror" value="<OvError">">
    </OCOvmlArrayFields>
    Identifiant de session = <OVwssessionid><br>
    erreur = <OVwserror><br>
</OCSoap>
```

L'identifiant de session est sauvegardé en utilisant la fonction `OFSetCookie` afin de le réutiliser dans les futurs appels SOAP et utiliser ainsi le contexte de l'utilisateur authentifié (voir paramètre `WSSESSIONID`):

```
<OCOvmlSoap
apiserver="http://autresite/index.php?tg=omlsoap&WSSESSIONID=<OVwssessionid>"
container="RecentArticles" last="10">
    <OVArticleTitle><br>
</OCOvmlSoap>
```

4.3 - Exécution d'un container sur un site Ovidentia

Tous les containers Ovidentia sont des Services Web (si les services web sont activés) et peuvent donc être appelés en utilisant le container ovml OCSOap :

Dans cet exemple, on demande une interprétation au site distant du container OCRecentArticles qui donne comme résultat une liste des derniers articles publiés.

```
<OFPutArray name="args" last="10">
<OCSOap apiserver="http://monsie/index.php?tg=omlsoap" apicall="babSoapOvml"
container="RecentArticles" args="(var)args">
  <OCOvmlArray name="wsresult" value="SoapResult" id1>
    <OCOvmlArray name="entry" value="wsresult Value" id2>
      <OCOvmlArray name="item" value="entryValue" id3>
        <OCIfEqual expr1="<OVitemValue>" expr2="ArticleTitle">
          <OVitem[value]><br>
        </OCIfEqual>
      </OCOvmlArray id3>
    </OCOvmlArray id2>
  </OCOvmlArray id1>
</OCSOap>
```