

SPECIFICATIONS TECHNIQUES EXTERNES

ACCES AU SERVICE MAILEVA

WEB SERVICE D'ENVOI ET DE SUIVI

Version : 1.0.9

Date : 18/01/2017

DOCUMENT CONFIDENTIEL

Sommaire

Sommaire.....	2
OBJET	3
DESCRIPTION DES WEBSERVICES.....	4
Limitations.....	4
Accès aux webservices	4
WEBSERVICE CONNECTOR	5
Description	5
Méthodes du webservice ConnectorWebService.....	5
La méthode « <i>submit</i> ».....	5
Paramètres du webservice ConnectorWebService	5
L'élément « <i>Campaign</i> » (campagne).....	5
L'élément « <i>Request</i> » (envoi)	6
L'élément « <i>DocumentData</i> » (documents)	6
L'élément « <i>Document</i> » (document).....	6
L'élément « <i>Content</i> » (contenu).....	7
Retour du webservice ConnectorWebService.....	8
WEBSERVICE TRACKING	9
Description	9
Méthodes du webservice TrackingWebService.....	9
Fonctionnement général	9
La méthode « <i>checkPendingNotifications</i> »	10
La méthode « <i>getPendingNotificationDetails</i> »	10
Paramètres du webservice TrackingWebService.....	11
L'élément « <i>PendingNotificationTypeWs</i> »	11
Retour du webservice TrackingWebService.....	12
L'élément « <i>PendingNotificationReferenceWs</i> »	12
L'élément « <i>PendingNotificationWs</i> »	13
Annexes	17
Code d'erreurs	17
Clients et librairies webservices	18
Client SOAP UI	18
Création du projet « <i>connecteur webservice</i> » :	18
Information d'identification dans SoapUI	18
Configuration du Proxy	18
Client JAVA.....	21
Classes génériques d'accès au webservice connecteur	21
<i>Classe utilitaire d'accès « ConnectorWsClient »</i>	21
<i>Configuration du Proxy</i>	22
<i>Classes utilitaires pour la construction de la campagne</i>	22
<i>Exemples de tests</i>	23
Lancement du client en ligne de commande	23

OBJET

Ce document présente la spécification d'accès au service Maileva via le connecteur webservice (protocole SOAP).

Ce document s'appuie et complète la dernière version du document **specifications_connecteur_xml_Maileva_v5.0.21_fr** qui décrit le format des données (XML).

Dans la spécification du format des données, les éléments composant un envoi sont décrits en détail.

Le présent document décrit les spécificités liées au dépôt et à la récupération du suivi via webservices.

DESCRIPTION DES WEBSERVICES

Limitations

Attention, le connecteur webservice ne permet pas de faire des envois volumineux: la taille de l'ensemble des documents déposés est limitée à 2 Mo.

Pour des envois plus volumineux, il est conseillé d'utiliser le dépôt par FTP / FTPS ou par CFT.

Accès aux webservices

Le webservice de dépôt est accessible en test à l'adresse suivante :

<https://webservices.recette.maileva.com/java/public/connector/ConnectorWebService?wsdl>

Le webservice de suivi est accessible en test à l'adresse suivante :

<https://webservices.recette.maileva.com/java/public/connector/TrackingWebService?wsdl>

L'accès à cette URL peut être restreint par adresse IP.

L'accès aux webservices n'est autorisé qu'aux utilisateurs Maileva via une authentification HTTP basique. Ces informations d'authentification sont fournies dans l'entête de la requête SOAP conformément aux standards HTTP (spécifications RFC 2617).

Lors de cette authentification vous devrez utiliser le login et le mot de passe fournis par Maileva.

Un cookie de session est renvoyé dans la réponse HTTP. Il permet de ne pas avoir à renvoyer le login / mot de passe Maileva lors des appels webservices suivants tant que le cookie est valable (le cookie a une durée de vie limitée à 2 heures). Une fois la durée de vie du cookie dépassée, une nouvelle authentification HTTP basique est nécessaire.

WEBSERVICE CONNECTOR

Description

Le webservice connector permet de réaliser des envois via Maileva

Méthodes du webservice ConnectorWebService

Ce webservice expose une méthode unique : « *submit* »

Cette méthode permet de déposer des campagnes d'envois multicanal.

La spécification prévoit les canaux Courrier, E-mail, Fax et SMS, mais seuls les canaux Courrier et SMS sont utilisables.

La méthode « *submit* »

Cette méthode n'accepte qu'un seul paramètre en entrée : l'élément « Campaign »

Elle retourne un seul élément : la string « référence d'envoi de la campagne »

Paramètres du webservice ConnectorWebService

L'élément « Campaign » (campagne)

L'entité « Campaign » ainsi que tous les éléments composant la campagne sont décrits en détail dans la spécification du format des données.

Cependant certaines spécificités existent pour le connecteur webservice; elles sont décrites par la suite dans ce document.

Le paramètre « Campaign » permet de regrouper des envois (« request ») et fournit certaines informations d'identification.

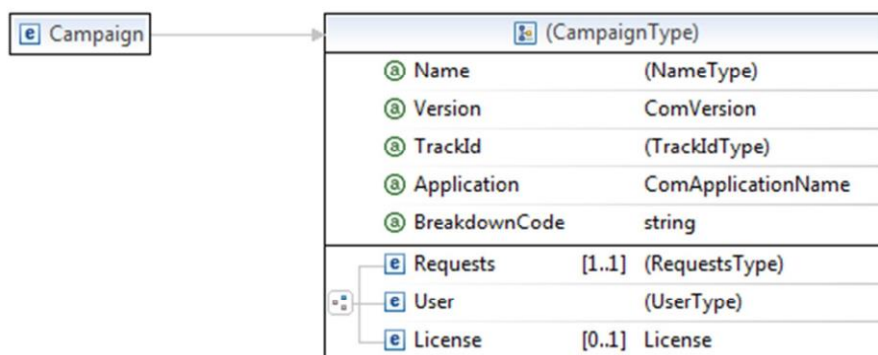


Figure 1. Description de l'élément « Campaign »

Attention, les propriétés suivantes ne sont pas prises en compte en connecteur webservice :

- Version : paramètre ignoré par le webservice
- User : paramètre ignoré par le webservice ; les informations utilisateurs sont récupérées lors de l'authentification HTTP

L'élément « Request » (envoi)

Cet élément décrit un envoi sur canal donné vers un ou plusieurs destinataires.

Une campagne se compose d'un ou plusieurs envois « Request ».

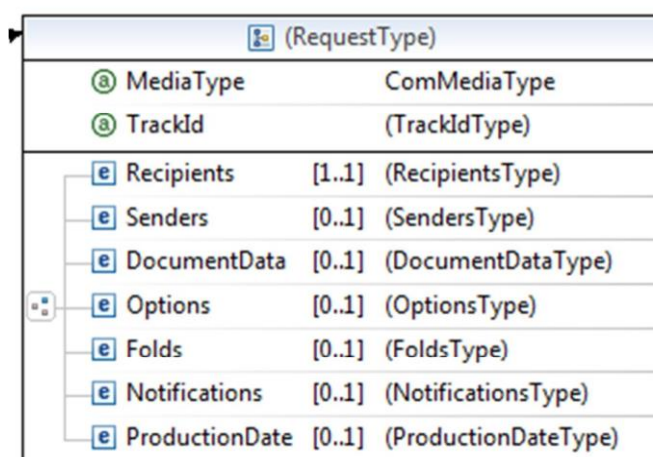


Figure 2. Description de l'élément « Request »

- « mediaType » : actuellement seul les médias « Courrier » et « SMS » sont supportés.
- « DocumentData » : ce paramètre est obligatoire en connecteur webservice

L'élément « DocumentData » (documents)

L'élément « DocumentData » est obligatoire pour les dépôts webservice. Il se compose des éléments « MergeFields » et « Documents » qui sont respectivement des listes de « MergeField » et de « Document ».

L'élément « Document » (document)

Détail d'un document à utiliser

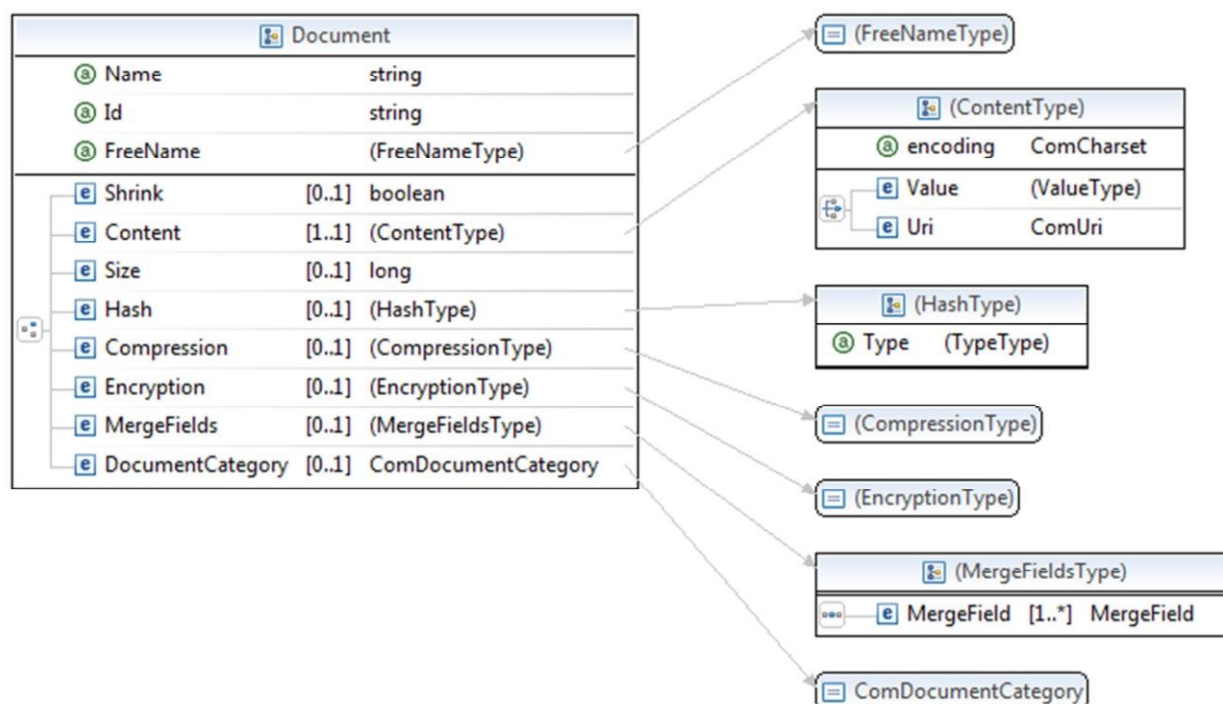


Figure 3. Description de l'élément « Document »

Attention, les propriétés suivantes ne sont pas prises en compte dans ce webservice :

- « Encryption » : non supporté dans cette version du webservice
- « Compression » : non supporté dans cette version du webservice
- « Hash » : non supporté dans cette version du webservice
- « Size » : non supporté dans cette version du webservice

L'élément « Content » (contenu)

Cet élément représente le contenu du document.

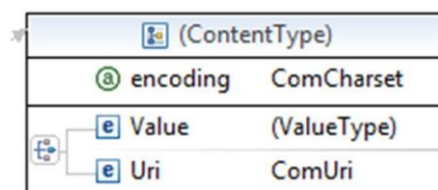


Figure 4. Description de l'élément « Content »

Propriété obligatoire en connecteur webservice :

- « value » : contenu du document encodé en base64

Les propriétés suivantes ne sont pas prises en compte en connecteur webservice :

- « Uri » : paramètre ignoré en connecteur webservice
- « encoding » : paramètre ignoré en connecteur webservice

Retour du webservice ConnectorWebService

Si un envoi de campagne est valide, un identifiant lui est attribué et est retourné par le webservice. Cet identifiant est dénommé « *Deposit Track ID* » et permet le suivi par le client. Cet identifiant est au format alphanumérique sur 32 caractères.

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <env:Body>
    <ns2:submitResponse xmlns:ns2="http://connecteur.services.siclv2.maileva.fr/">
      <return>957dc3a778df45dcac305d3d7eaca3f0</return>
    </ns2:submitResponse>
  </env:Body>
</env:Envelope>
```

Figure 5. Exemple d'un retour du webservice ConnectorWebService sur un dépôt valide

Si l'envoi de campagne échoue, un code d'erreur ainsi qu'un message d'erreur sont retournés par le webservice sous forme d'exception SOAP « Soap Fault Exception ».

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <env:Body>
    <env:Fault>
      <faultcode xmlns:codeNS="http://www.maileva.fr/MailevaPJSSchema">codeNS:5000001</faultcode>
      <faultstring>La campagne ne contient pas de documents</faultstring>
    </env:Fault>
  </env:Body>
</env:Envelope>
```

Figure 6. Exemple d'un retour d'erreur du webservice ConnectorWebService

Voir la section code d'erreurs pour la liste complète des codes d'erreurs.

Remarque : Les contrôles du ou des envois constituant la campagne sont fait ultérieurement.

Il est donc possible qu'un envoi de campagne soit valide, mais que le ou les envois associés soient rejetés par la suite.

WEBSERVICE TRACKING

Description

Le webservice tracking permet de suivre l'état d'avancement d'envois réalisés via Maileva

Méthodes du webservice TrackingWebService

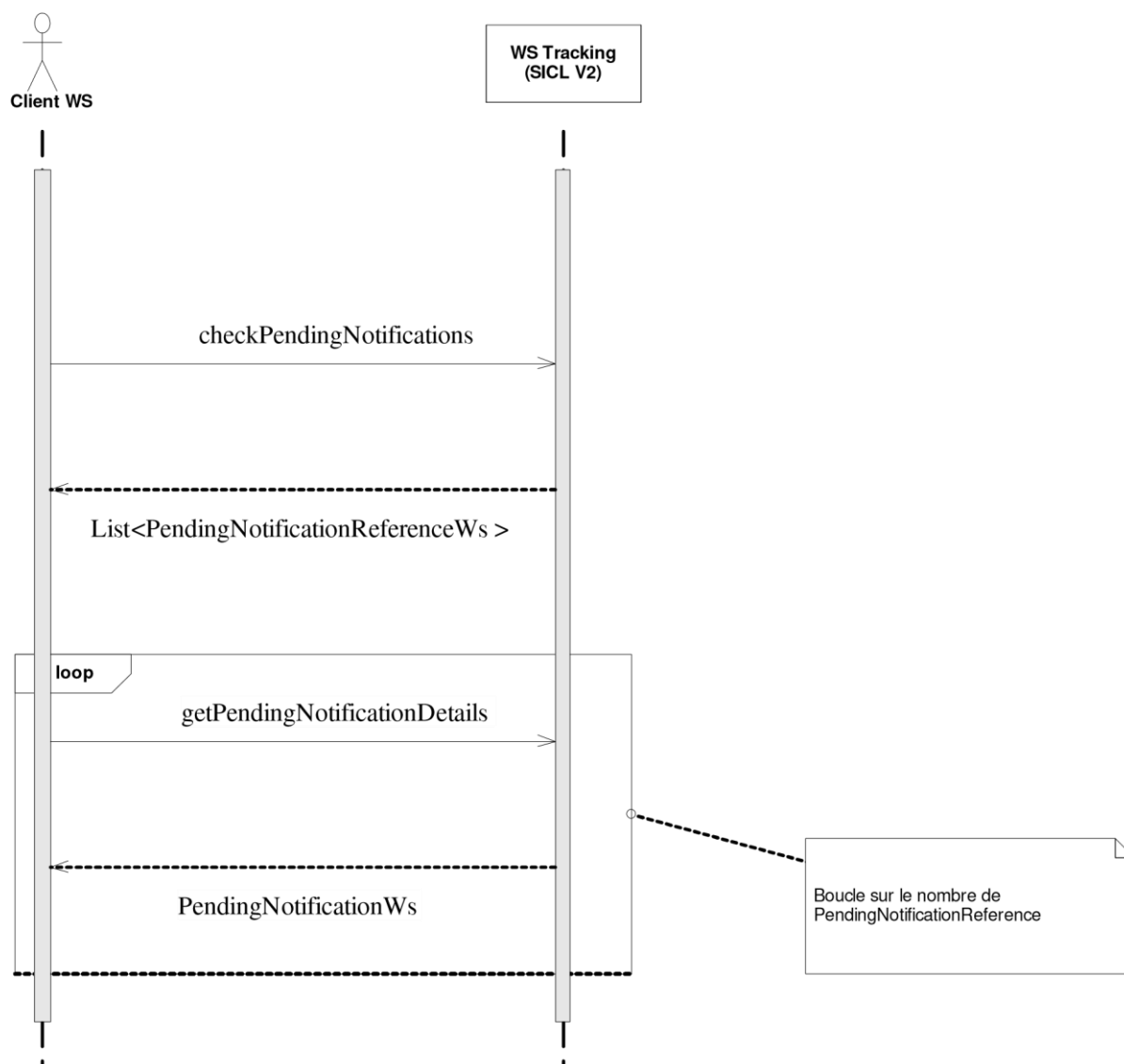
Ce webservice expose 2 méthodes :

- `checkPendingNotifications` : permet de récupérer la liste d'envois pour lesquelles des informations de suivi sont récupérables.
- `getPendingNotificationDetails` : permet de récupérer le détail des informations de suivi sur une demande donnée

Fonctionnement général

La récupération du suivi se déroule globalement en 2 étapes qui correspondent aux 2 méthodes exposées :

- 1ère étape : le client appelle la méthode « `checkPendingNotifications` » pour récupérer les identifiants des notifications en attente.
- 2^{de} étape : le client boucle sur la liste de notifications en attente et récupère le détail pour chaque notification via la méthode « `getPendingNotificationDetails` »



La méthode « checkPendingNotifications »

Cette méthode permet de récupérer la liste d'envois pour lesquelles des informations de suivi sont disponibles. Elle doit être appelée régulièrement par le client webservice pour récupérer de nouvelles informations de suivi en attente.

A chaque appel, la liste retournée des informations de suivi en attente est limitée à 1000.

Cette méthode n'accepte qu'un seul paramètre en entrée : l'élément « PendingNotificationTypeWs ». Elle retourne une liste de « PendingNotificationReferenceWs ».

La méthode « getPendingNotificationDetails »

Cette méthode permet de récupérer le détail des informations de suivi sur une demande donnée.

Une fois les informations de suivi récupérées pour une demande donnée, celles-ci ne seront plus récupérables.

Cette méthode n'accepte qu'un seul paramètre en entrée : un élément de type « long » `pendingNotificationId`, l'identifiant de la notification à récupérer.

Paramètres du webservice `TrackingWebService`

L'élément « `PendingNotificationTypeWs` »

L'entité « *PendingNotificationTypeWs* » est une énumération qui permet de spécifier le type de suivi à récupérer.

Pour l'instant un seul type de suivi est récupérable via ce webservice, le type « GENERAL ». Merci de vous référer aux spécifications connecteur pour plus de détails sur ce type de notifications.

Retour du webservice TrackingWebService

L'élément « PendingNotificationReferenceWs »

L'entité « PendingNotificationReferenceWs » permet de récupérer les informations de base (identifiants) associées à une notification en attente.

id (long) : Identifiant de la notification (à utiliser pour récupérer le détail)
req_id (long) : Identifiant interne de la demande
req_track_id (string) : Identifiant client de la demande
de_deposit_id (string): Référence de l'envoi
de_deposit_track_id (string) : Référence du dépôt (renvoyée lors d'un dépôt par webservice)

Figure 7. Exemple d'un retour du webservice TrackingWebService (checkPendingNotifications)

<pre><env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"> <env:Header/> <env:Body> <ns2:checkPendingNotificationsResponse xmlns:ns2="http://tracking.services.siclv2.maileva.fr/"> <pendingNotificationsReferences> <id>5</id> <reqId>50409</reqId> <reqTrackId>567</reqTrackId> <depositId>897</depositId> <depositTrackId>846</depositTrackId> </pendingNotificationsReferences> <pendingNotificationsReferences> <id>4</id> <reqId>50409</reqId> <reqTrackId>456</reqTrackId> <depositId>789</depositId> <depositTrackId>123</depositTrackId> </pendingNotificationsReferences> <pendingNotificationsReferences> <id>3</id> <reqId>50409</reqId> <reqTrackId>BULLETIN DE PAIE JUILLET 2011</reqTrackId> <depositId>25324800301</depositId> <depositTrackId/> </pendingNotificationsReferences> <pendingNotificationsReferences> <id>1</id> <reqId>1</reqId> <reqTrackId>123</reqTrackId> <depositId>456</depositId> <depositTrackId>789</depositTrackId> </pendingNotificationsReferences> </ns2:checkPendingNotificationsResponse> </env:Body> </env:Envelope></pre>	
--	--

L'élément « PendingNotificationWs »

L'entité « PendingNotificationWs » permet de récupérer le détail des informations d'une notification en attente.

Attribut	Type	Description
Id	Long	Identifiant de la notification en attente
status	string	Statut de la notification en attente (PENDING SENT)
notificationStatus	string	Statut de la notification en attente (ACCEPT NACCEPT NOK OK)
notificationType	string	Type de notification en attente (GENERAL LRE PND AR POSTE)
reqId	long	Identifiant de la request associée
reqTrackId	string	Identifiant client de la demande
reqMediaType	string	Type de media de la request (PAPER DIGITAL SMS FAX EMAIL VOTE)
reqMediaSubType	string	Sous-type de media de la request (INVOICE)
depositId	int	Référence Maileva du dépôt
depositTrackId	string	Référence du dépôt attribuée par le client
receptionDate	date	Date de réception de l'envoi
productionDate	date	Date de production de l'envoi
foldsCount	int	Nombre de plis associé à l'envoi
documentsCount	int	Nombre de documents associé à l'envoi si media COURRIER DIGITAL VOTE
pagesCount	int	Nombre de pages associé à l'envoi si média COURRIER DIGITAL VOTE FAX
nbBilledPage	int	Nombre de pages facturées associé à l'envoi si média COURRIER DIGITAL VOTE FAX
digitalFoldsCount	int	Nombre de plis numériques associé à un envoi DIGITAL
envelopeType	string	Options : type d'enveloppe si média COURRIER DIGITAL VOTE
postageClass	string	Options : catégorie d'affranchissement si média COURRIER DIGITAL VOTE
hasColorPage	boolean	Options : indique si l'envoi est imprimé en couleur si média COURRIER DIGITAL VOTE
printDuplex	boolean	Options : indique si l'envoi est imprimé en recto/verso si média COURRIER DIGITAL VOTE
useFlyLeaf	boolean	Options : indique l'utilisation d'une page de garde si média FAX
flyLeafCustomRef	string	Options : Référence de la page de garde personnalisée si média FAX
flyLeafWarning	string	Options : Mention « urgent » ajoutée sur la page de garde si média FAX
useAddressBook	boolean	Le consentement du destinataire est dans le carnet d'adresses si média PAPER et sous-type INVOICE

emailFrom	string	Nom libre associé à l'adresse e-mail de l'émetteur des factures dématérialisées (média PAPER et sous-type INVOICE)
emailReply	string	Adresse e-mail de retour si média PAPER et sous-type INVOICE
emailFooter	string	Pied de page des factures dématérialisées (média PAPER et sous-type INVOICE)
msgObject	string	Options : objet du message électronique si média EMAIL ou si média PAPER et sous-type INVOICE
senderTaxNumber	string	Numéro de TVA intra-communautaire de l'émetteur
senderSiret	string	Siret de l'émetteur
senderCompanyName	string	Raison sociale de l'émetteur
senderCountryCode	string	Code pays de l'émetteur
senderPaymentType	string	Type de paiement de l'émetteur : VAT_DEBIT VAT_CASHING
recipients	Container (recipient)	Liste des destinataires

L'élément « recipients »

L'entité « *recipients* » est une liste d'éléments « recipient ».

Elément	Description	Type	Cardinalité	Valeur
recipient	Détail d'un destinataire	Container	1..n	

- recipient : Détail d'un destinataire

Voir [Élément recipient](#)

L'élément « recipient »

Cet élément fournit les informations concernant un destinataire.

Attribut	Description	Type	Présence	Valeur
foldId	Identifiant technique du pli	String	O	
foldTrackId	Identifiant du pli attribué par le client	String	F	
sendingMode	Canal d'envoi pour un destinataire d'une facture	String	F	EMAIL PAPER CHORUS
postalAddress1	Ligne 1 de l'adresse postale	String (38)	F	
postalAddress2	Ligne 2 de l'adresse postale	String (38)	F	

postalAddress3	Ligne 3 de l'adresse postale	String (38)	F	
postalAddress4	Ligne 4 de l'adresse postale	String (38)	F	
postalAddress5	Ligne 5 de l'adresse postale	String (38)	F	
postalAddress6	Ligne 6 de l'adresse postale	String (38)	F	
postalCountryCode	Code pays de l'adresse postale	String (2)	F	
postalCountryName	Nom du pays de l'adresse postale	String (38)	F	
erlNumber	Numéro de recommandé	String	F	
erlDeliveryDate	Date de preuve de dépôt	Datetime	F	
foldErrorCode	Code d'erreur si le pli est rejeté	Int	F	
foldErrorLabel	Libellé d'erreur si le pli est rejeté	String	F	
invoiceIdentifier	Identifiant du destinataire de la facture	String	F	
invoiceEmail	E-mail du destinataire de la facture	String	F	

- **foldId** : Identifiant technique du pli
Identifiant technique attribué par Maileva à chaque pli
- **foldTrackId** : Identifiant du pli attribué par le client
Il s'agit du paramètre TrackId qui a été fourni au niveau du pli au moment de l'envoi
- **sendingMode** : Canal d'envoi pour un destinataire d'une facture
Dans le cas d'un envoi de facture, ce champ indique le canal utilisé pour un destinataire (courrier postal ou envoi dématérialisé)
PAPER : courrier postal
EMAIL : envoi dématérialisé
CHORUS : envoi vers le portail Chorus Pro
- **postalAddress1** à **postalAddress6** : Lignes de l'adresse postale Ces 6 lignes désignent l'adresse postale du destinataire.
La signification des lignes d'adresses est donnée en annexe.
- **postalCountryCode** : Code pays de l'adresse postale
Indique le code pays associé à l'adresse postale selon la codification ISO 3166
- **postalCountryName** : Nom du pays de l'adresse postale
- **erlNumber** : Numéro de recommandé
Identifiant postal attribué à une lettre recommandée.
Cet identifiant alphanumérique permet de suivre la distribution de la lettre recommandée dans le réseau postal.
- **erlDeliveryDate** : Date de preuve de dépôt
Il s'agit de la date de preuve électronique de dépôt dans le réseau postal.
- **foldErrorCode** : Code d'erreur

Code d'erreur concernant un destinataire dans le cas où il est rejeté.

La liste des codes d'erreur est fournie dans les spécifications connecteur XML

- `foldErrorLabel` : Libellé d'erreur

Description de l'erreur concernant un destinataire dans le cas où il est rejeté.

- `invoiceIdentifier` : Identifiant du destinataire de la facture

Ce champ fournit par le client permet d'identifier de manière unique chez le client, un destinataire d'une facture.

- `invoiceEmail` : E-mail du destinataire de la facture

Adresse e-mail du destinataire utilisée pour un envoi de facture dématérialisé.

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <env:Body>
    <ns2:getPendingNotificationDetailsResponse xmlns:ns2="http://tracking.services.siclv2.maileva.fr/">
      <pendingNotification>
        <id>5</id>
        <status>SENT</status>
        <notificationStatus>ACCEPT</notificationStatus>
        <notificationType>GENERAL</notificationType>
        <reqId>50409</reqId>
        <reqTrackId>567</reqTrackId>
        <reqMediaType>DIGIT</reqMediaType>
        <depositId>897</depositId>
        <depositTrackId>846</depositTrackId>
        <receptionDate>2013-09-07T03:04:05.600+02:00</receptionDate>
        <productionDate>1900-01-01T00:00:00+01:00</productionDate>
        <foldsCount>4</foldsCount>
        <documentsCount>4</documentsCount>
        <pagesCount>4</pagesCount>
        <nbBilledPage>4</nbBilledPage>
        <digitalFoldsCount>4</digitalFoldsCount>
        <envelopeType>C6</envelopeType>
        <postageClass>STANDARD</postageClass>
        <hasColorPage>false</hasColorPage>
        <printDuplex>false</printDuplex>
        <useFlyLeaf>false</useFlyLeaf>
        <flyLeafCustomRef/>
        <flyLeafWarning/>
        <msgObject/>
      </pendingNotification>
    </ns2:getPendingNotificationDetailsResponse>
  </env:Body>
</env:Envelope>
```

Figure 8. Exemple d'un retour du webservice TrackingWebService (`getPendingNotification`)

Annexes

Code d'erreurs

Code	Description
5000001	un des envois de la campagne ne contient pas de document
5000002	un des envois contient des documents non valides
5000003	un des paramètres de la campagne est invalide
5000009	type de canal non supporté
5200028	taille maximale autorisée des documents chargés dépassée
2000000	Erreur indéterminée lors du traitement de la demande

Clients et librairies webservices

Client SOAP UI

[SoapUI](#) est un utilitaire gratuit, très pratique pour le test de webservices.

Il offre une interface assez simple avec une multitude d'options et de paramétrages possibles (authentification, log, passage de paramétrage dans les requêtes, ...)

Création du projet « connecteur webservice » :

1. Créez le nouveau projet à travers le menu « File », « New soapUI Project » (voir Figure 7)
2. Dans la boîte de dialogue qui apparaît, voir Figure 8, insérez dans le champ « Initial WSDL » l'URL d'accès au Connecteur Webservice
3. Si l'accès au webservice est correct, SoapUI créera le nouveau projet (interface d'accès ainsi que les méthodes d'appel) sous le nom indiqué (*TEST – Connecteur Webservice* par exemple), voir Figure 9.

Information d'identification dans SoapUI

L'accès au Webservice connecteur nécessite une authentification basique HTTP. Dans SoapUI, ces informations sont renseignées au niveau de chaque requête « Request » dans le volet « Aut », voir Figure 10.

Attention, pour l'authentification basique HTTP, faire attention à bien choisir « Preemptive » comme mode authentification dans le menu déroulant « Autorisation Type ».

Configuration du Proxy

Si votre accès internet se fait via un proxy :

1. Renseignez ces informations dans SoapUI dans le menu « File », « Preferences », volet « ProxySettings »
2. Une fois le proxy configuré, activez le proxy pour vos appels via le bouton « Apply proxy ... », voir
3. Figure 11

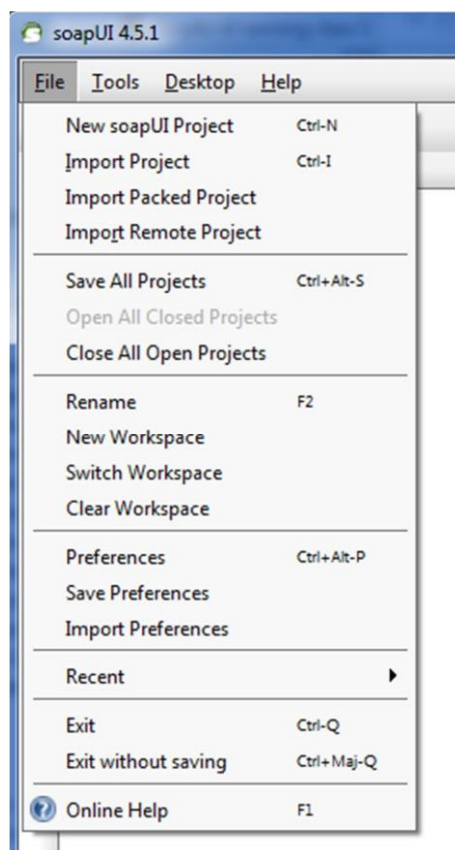


Figure 7. Création d'un nouveau projet dans SoapUI

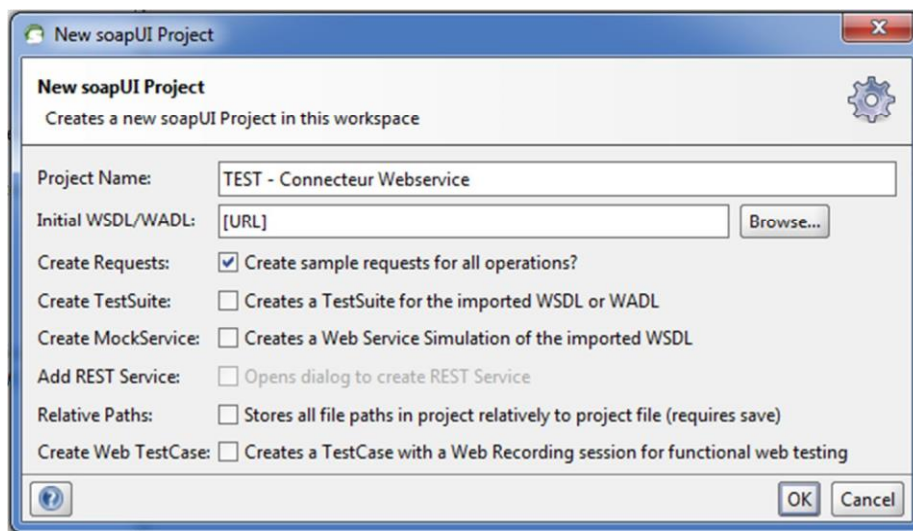


Figure 8. Dialogue de création du nouveau projet dans SoapUI

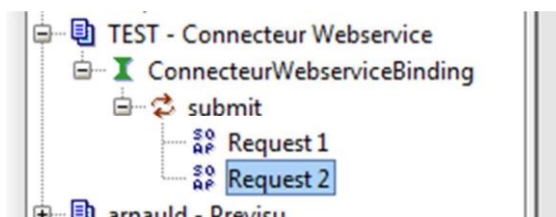


Figure 9. Présentation du projet dans SoapUI

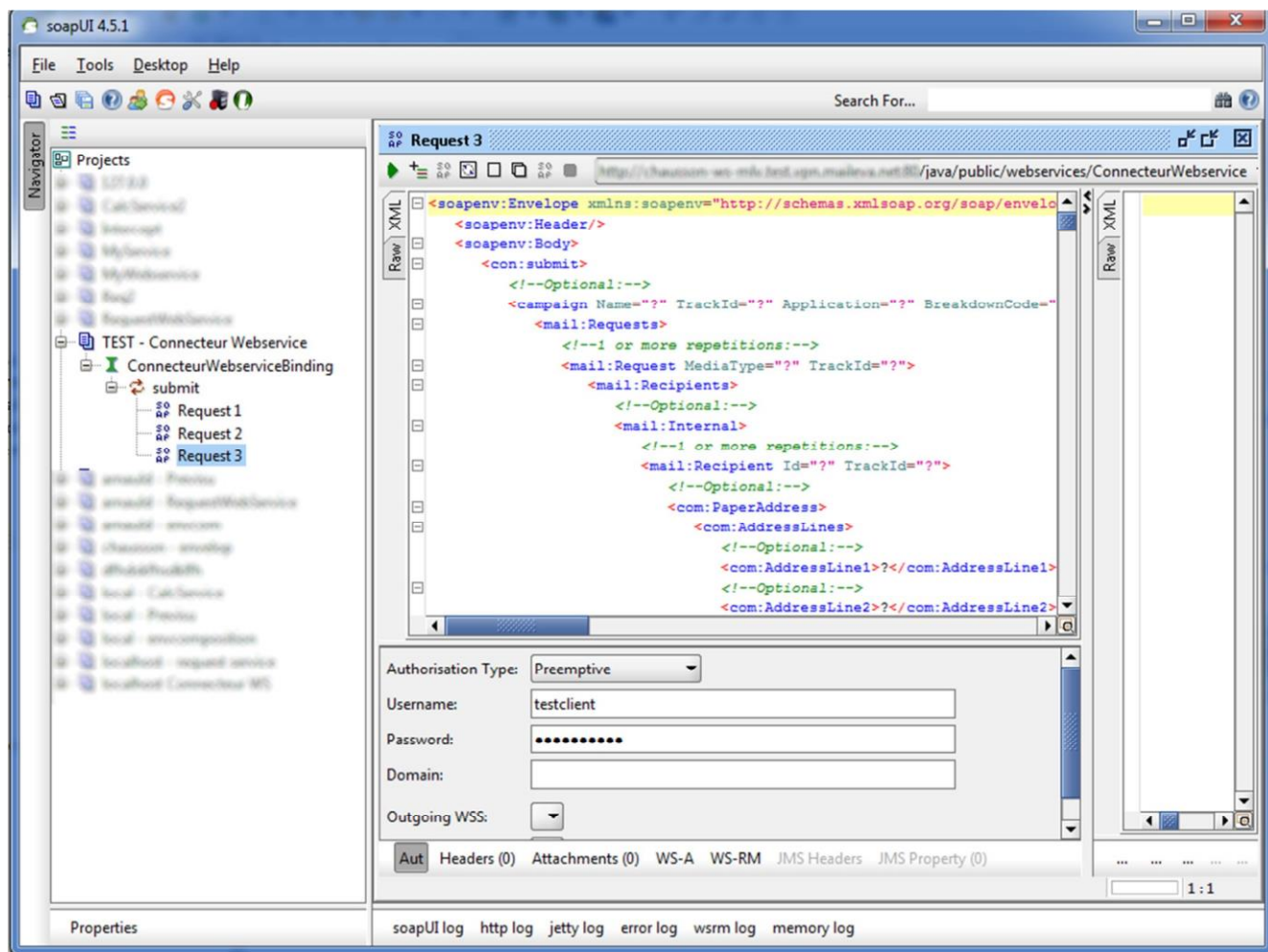


Figure 10. Authentification dans SoapUI

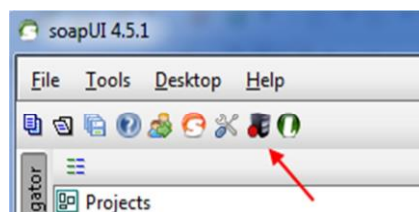


Figure 11. Activation du proxy dans SoapUI

Client JAVA

Pour l'accès au webservice connecteur, une librairie java « *connecteur-ws-client.jar* » est disponible. Elle contient un certain nombre de classes utilitaires pour l'accès au webservice et la construction des différents paramètres.

Remarque : l'utilisation de cette librairie nécessite la version 1.6 ou plus de Java.

Classes génériques d'accès au webservice connecteur

Les classes qui constituent le point d'entrée d'accès au webservice connecteur sont dans le package :

« *fr.maileva.siclv2.services.connecteur* »

Exemple d'utilisation du client webservice

```
// initialisation du client webservice
URL wsdlUrl = new URL
("https://webservices.recette.maileva.com/java/public/connecteur/ConnectorWebService?wsdl");
ConnectorWebservice service = new ConnectorWebserviceService(wsdlUrl)
                                                                    .getConnectorWebservicePort();

// code d'initialisation de la campagne ...
// code de paramétrage de l'authentification ...
// appel du webservice
String referenceCampaign = service.submit(campaign);
```

Attention à bien configurer, lors de la création d'un client, de :

- l'authentification (via la classe « *java.net.Authenticator* » par exemple)
- la négociation SSL (les webservices maileva sont accessibles via le protocole SSL)

Classe utilitaire d'accès « *ConnectorWsClient* »

La classe « *ConnectorWsClient* » du package « *fr.maileva.webservices.client* » est une classe utilitaire qui offre une facilité pour l'instanciation du client webservice avec configuration de l'authentification et gestion de la négociation SSL (validation du certificat) :

```
// initialisation du client utilitaire webservice
URL wsdlUrl = new URL
("https://webservices.recette.maileva.com/java/public/connecteur/ConnectorWebService?wsdl");
ConnectorWsClient client
= ConnectorWsClient
    .build(wsdlUrl)
    .withAuthentication( new PasswordAuthentication("login", "mot_de_passe".getBytes()) );
// code d'initialisation de la campagne ...
```

```
// appel du webservice
```

```
String referenceCampaign = client.submit(campaign);
```

Configuration du Proxy

Si l'accès au webservice nécessite un proxy, il suffit de rajouter une configuration proxy lors de l'instanciation du client :

```
// initialisation du client utilitaire webservice
```

```
URL wsdlUrl = new URL
```

```
("https://webservices.recette.maileva.com/java/public/connector/ConnectorWebService?wsdl"); ConnectorWsClient client  
= ConnectorWsClient
```

```
    .build(wsdlUrl)
```

```
    .withAuthentication( new PasswordAuthentication("login", "mot_de_passe".getBytes())
```

```
    .withProxy("http://monproxy.net", 8080, null) );
```

```
// code d'initialisation de la campagne ...
```

```
// appel du webservice
```

```
String referenceCampaign = client.submit(campaign);
```

Le troisième paramètre permet de préciser une liste d'URI pour lesquels l'accès est direct (sans PROXY).

Classes utilitaires pour la construction de la campagne

La construction d'une nouvelle campagne nécessite, du côté client, l'initialisation d'un certain nombre d'objets (Requests, Documents, ... etc.). Certains de ces objets doivent être encapsulés dans d'autres objets.

Le package « *fr.maileva.webservices.client* » offre un certain nombre de classes utilitaires qui permettent de faciliter la création des paramètres :

- La classe « AddressFactory » permet d'obtenir des adresses typées par canal (i.e. Courrier, Email, Fax, SMS) via les méthodes « paper() », « email() », « fax() » et « sms() »
- La classe « DocumentFactory » permet de retourner des instances de « *fr.maileva.mailevapjsschema.Request.DocumentData.Documents.Document* » à partir d'un fichier ou d'une URI. Le contenu du document ainsi généré est encodé en Base64. La classe permet également d'encapsuler les documents ainsi générés dans « *fr.maileva.mailevapjsschema.Request.DocumentData* » qui est utilisé dans la campagne.
- La classe « RecipientFactory » permet d'obtenir des objets destinataires de type « *fr.maileva.commonschema.Recipient* » via les méthodes « paper() », « email() », ... Les « Recipient » ainsi générés peuvent ensuite être encapsulés en « *fr.maileva.mailevapjsschema.Request.Recipients* » via la méthode « wrap() ».

Les classes citées ci-dessus peuvent contenir d'autres méthodes utilitaires, référez à la documentation java (javadoc) disponible pour une description complète.

Exemples de tests

La librairie (code source) est accompagnée d'un lot de tests (un par type de canal) dans le package et sous packages « *fr.maileva.webservices.client.tests* »

La classe « *fr.maileva.webservices.client.tests.PaperCampaignTest* » est un exemple de test sur le canal courrier.

Ces tests utilisent la librairie « org.junit » pour l'exécution des tests.

Lancement du client en ligne de commande

Il est possible de lancer le client java depuis une ligne de commande pour *tester* l'accessibilité au webservice :

```
java -jar connector-ws-client-1.0.0.jar -email mon-email@company.com -file mondoc.pdf  
-login monlogin -password monpwd  
-url https://webservices.recette.maileva.com/java/public/connector/ConnectorWebService?wsdl
```

Un descriptif complet des options peut être obtenu en lançant le programme sans option :

```
java -jar connector-ws-client-1.0.0.jar
```

ce qui provoque l'affichage du message d'aide suivant :

```
2013-01-30 10:08:17,347 [main] ERROR WsClient - Missing required options: url, login, password, email, file ...  
usage: java -jar connector-ws-client -email <email> -file <file> -login    <login> -password <password> [-  
proxy <host:port>] -url <webservice url>  
-email <email>      notification email  
-file <file>        document file (size <= 1 MO)  
-login <login>      user login  
-password <password> user password  
-proxy <host:port>  proxy hostname and port  
-url <webservice url> Webservice URL
```