inWebo API Documentation

Table des matières

Introduction **3**

Authentification avec l'API SOAP **3**

Authenticate *3*

AuthenticateWithIP *3*

Authentification avec l'API REST **4**

Authentification en mode « 2-Step » avec requête Push sur le Smartphone **5**

PushAuthenticate *5*

CheckPushResult *6*

Scellement de transactions avec l'API REST **7**

Gestion des utilisateurs avec l'API SOAP **8**

loginsQuery *8*

loginQuery *9*

loginSearch *10*

loginCreate *11*

loginUpdate *12*

loginSendByMail *12*

loginActivateCode *12*

loginDelete *12*

loginResetPwd *13*

loginResetPwdExtended *13*

loginResetPINErrorCounter *14*

loginRestore *14*

loginAddDevice *14*

loginDeleteTool *15*

Introduction

Ce document décrit la liste des Web Services de l'API inWebo. La partie SOAP de l'API comporte 2 fichiers WSDL :

- Authenticate.wsdl // à utiliser pour les demandes d'authentification
- Provisioning.wsdl // à utiliser pour la gestion des utilisateurs

L'API SOAP est sécurisée par 2 facteurs, configurables dans la console d'administration inWebo :

- Restriction des adresses IP sources autorisées à faire des requêtes
- Implémentation d'une authentification par certificat SSL

Certaines méthodes de l'API sont également proposées au format REST.

Attention : en fonction de l'offre souscrite (Safe Transactions, Entreprise), certaines fonctions ne seront pas disponibles.

Authentification avec l'API SOAP

Il existe 2 méthodes d'authentification: authenticate et authenticateWithIP. La première va simplement valider un login et un OTP. La 2e permettra, dans le cas d'un OTP généré par l'application inWebo pour Mac & PC ou inWebo Helium, de vérifier l'adresse IP Source de l'utilisateur final.

Authenticate

```
authenticate(String userId, String serviceId, String token)
```

Retourne :

```
authenticateResponse
```

C'est un objet ayant pour propriétés :

```
public String authenticateReturn;
```

Cette fonction permet de valider un mot de passe non rejouable pour un login donné. La string retournée renvoie un code de résultat « OK » (succès) ou « NOK:<cause> » (erreur).

AuthenticateWithIP

```
authenticateWithIpResponse authenticateWithIP(String userId, String serviceId, String token, String ip)
```

Retourne :

```
authenticateWithIpResponse
```

C'est un objet ayant pour propriétés :

```
public String authenticateReturn;
```

AuthenticateWithIP offre une fonction supplémentaire de détection MITM pour les OTP inWebo. Le fonctionnement est le suivant :

- Si l'OTP a été généré par l'application inWebo pour Mac & PC ou inWebo Helium, en plus de vérifier l'OTP, inWebo compare l'IP à celle qu'il connait
- Si l'OTP a été généré par nCode, inWebo ignore l'adresse IP

La string retournée renvoie un code de résultat « OK » (succès) ou « NOK:<cause> » (erreur).

Authentification avec l'API REST

authenticateExtended

URL :

https://api.myinwebo.com/FS?action=authenticateExtended + paramètres

Les paramètres obligatoires sont :

```
&serviceId= <id du service> //integer
&userId=<login> //string
&token=<l'otp généré> //string
```

Les paramètres optionnels sont :

```
&format=json //permet de récupérer la réponse à l'appel API au format json
```

Réponse:

Par défaut, la réponse de l'API est au format XML

Cette réponse contient les éléments suivants :

- err : le résultat de l'authentification (« OK » or « NOK:<cause> »)
- name: le nom de l'outil qui a réalisé l'authentification
- alias: l'alias de l'id de cet outil
- version: la version de l'outil
- platform: la plateforme de l'outil - peut valoir « helium » / « windows » / « mac » / « android »…
- type: le type de l'outil – peut valoir « ma » (app mobile) / « ca » (application inWebo ou inWebo Helium / « mac « (une app basée sur maccess)

Si l'id du service passé dans l'appel ne représente pas un id de service valide, la valeur de « err » sera « NOK:srv unknown ».

Si le login passé dans l'appel ne représente pas un login valide du service, la valeur de « err » sera « NOK:account unknown ».

Format de la réponse en XML :

```
<authenticateExtended>
 <err> </err>
 <name> </name>
 <alias> </alias>
 <version> </version>
 <platform> </platform>
 <type> </type>
 <timestamp> </ timestamp >
</authenticateExtended>
```

Format de la réponse en json :

```
{ "timestamp":"","platform":"","alias":"","name":"","err":"","type":"","version":""}
```

Authentification en mode « 2-Step » avec requête Push sur le Smartphone

Ce mode d'authentification est utilisé pour ajouter une couche de sécurité à une authentification par mot de passe classique. D'un point de vue « utilisateur », il y a 2 étapes :

1. Authentification par mot de passe classique
2. Puis, authentification forte sur mobile, via une requête Push envoyée sur le smartphone

Pour cela, inWebo expose 2 Web Services REST qui permettent de réaliser la 2e étape. Le 1er appel envoie une notification Push pour réveiller l'application inWebo Authenticator sur le smartphone de l'utilisateur. Le 2e appel permet à votre serveur de vérifier si l'authentification forte a bien été effectuée.

Comme il n'est pas possible de prédire le temps que mettra l'utilisateur à valider l'authentification, une implémentation asynchrone a été retenue :

1. Votre serveur émet une demande l'authentification auprès d'inWebo
2. Votre serveur vérifie périodiquement auprès d'inWebo le résultat de l'authentification

PushAuthenticate

URL :

```
https://api.myinwebo.com/FS?action= pushAuthenticate + paramètres
```

Les paramètres obligatoires sont :

```
&serviceId= <id du service> //integer
&userId=<login> //string
```

Les paramètres optionnels sont :

```
&format=json //permet de récupérer la réponse à l'appel API au format json
```

Réponse:

Par défaut, la réponse de l'API est au format XML

Cette réponse contient les éléments suivants :

- err : le résultat de la demande de notification (« OK » or « NOK:<cause> »)
- name: le nom de l'outil qui a réalisé l'authentification
- alias: l'alias de l'id de cet outil
- version: la version de l'outil
- platform: la plateforme de l'outil - peut valoir « iphone » / « wp8 » / « android »
- type: le type de l'outil. Vaut « ma » (app mobile inWebo) ou « mac » (votre application développée avec mAccess)
- sessionId : l'Id de session qui permettra ultérieurement de vérifier le résultat de l'authentification (à utiliser dans checkPushResult)

Codes d'erreur possibles :

- NOK:NoPush : l'utilisateur a une application inWebo qui ne supporte pas le Push (inWebo nCode)
- NOK:NoMA : l'utilisateur n'a pas activé l'application inWebo
- NOK:NOLOGIN : l'utilisateur n'existe pas
- NOK:SN : Problème de syntaxe dans l'un des paramètres envoyés
- NOK:access forbidden : Le serviceId que vous avez passé est erroné

Format de la réponse en json :

```
{"timestamp":"","platform":"","sessionId":"","alias":"","name":"","err":"","type":"","version":""}
```

CheckPushResult

Cet appel permet d'obtenir le résultat d'une demande d'authentification.

URL :

```
https://api.myinwebo.com/FS?action= checkPushResult + paramètres
```

Les paramètres obligatoires sont :

```
&serviceId= <id du service> //integer
&sessionId=<id de session> //string
&userId=<login> //string
```

Les paramètres optionnels sont :

```
&format=json //permet de récupérer la réponse à l'appel API au format json
```

Réponse:

Par défaut, la réponse de l'API est au format XML

Cette réponse contient les elements suivants :

- err : le résultat de l'authentification (« OK » or « NOK:<cause> »)
- name: le nom de l'outil qui a réalisé l'authentification
- alias: l'alias de l'id de cet outil
- version: la version de l'outil
- platform: la plateforme de l'outil - peut valoir « iphone » / « wp8 » / « android »
- type: le type de l'outil – peut valoir « ma » (app mobile) / « mac « (une app basée sur mAccess)

Codes d'erreur possibles :

- NOK:WAITING : la demande est en cours. Votre serveur devra ré-essayer plus tard (0.5 sec plus tard par exemple)
- NOK:REFUSED : l'authentification a été refusée par l'utilisateur
- NOK:TIMEOUT : L'utilisateur n'a pas accepté l'authentification dans le délai imparti (1 minute)
- NOK:SN : Problème de syntaxe dans l'un des paramètres envoyés
- NOK:access forbidden : Le serviceId que vous avez passé est erroné
- NOK: Le sessionId n'existe pas ou plus

Format de la réponse en json :

```
{"timestamp":"","platform":"","alias":"","name":"","err":"","type":"","version":""}
```

Scellement de transactions avec l'API REST

```
sealVerify
```

URL :

```
https://api.myinwebo.com/FS?action=sealVerify + paramètres
```

Les paramètres obligatoires sont :

```
&serviceId= <id du service> //integer
&userId=<login> //string
&token=<l'otp généré> //string
&data=<la donnée à sceller> //string
```

Les paramètres optionnels sont :

```
&format=json //permet de récupérer la réponse à l'appel API au format json
```

Réponse:

Par défaut, la réponse de l'API est au format XML

Cette réponse contient les elements suivants :

- err : le résultat de l'opération (« OK » or « NOK:<cause> »)
- name: le nom de l'outil qui a réalisé le scellement
- alias: l'alias de l'id de cet outil
- version: la version de l'outil
- platform: la plateforme de l'outil – celle que vous avez passé à mAccess dans votre application
- type: vaudra systématiquement « mac » (une app basée sur maccess)

Si l'id du service passé dans l'appel ne représente pas un id de service valide, la valeur de « err » sera « NOK:srv unknown ».

Si le login passé dans l'appel ne représente pas un login valide du service, la valeur de « err » sera « NOK:account unknown ».

Autres codes d'erreur possibles :

- NOK:NoKey : l'utilisateur n'a pas de clé de scellement. Cela veut dire qu'il n'a probablement pas activé votre application
- NOK:BadData : la donnée scellée dans l'OTP est différente de celle envoyée par le Web Service
- NOK:FORBIDDEN : Opération interdite. Le scellement de transactions n'est pas autorisé pour votre compte

Format de la réponse en XML :

```
<authenticateExtended>
 <err> </err>
```

```
<name> </name>
<alias> </alias>
<version> </version>
<platform> </platform>
<type> </type>
<timestamp> </ timestamp >
</authenticateExtended>
```

Format de la réponse en json :

```
{ "timestamp":"","platform":"","alias":"","name":"","err":"","type":"","version":""}
```

Gestion des utilisateurs avec l'API SOAP

loginsQuery

```
LoginsQueryResult loginsQuery(long userid, long serviceid, long offset, long nmax, long sort)
```

Retourne :

```
LoginsQueryResult
```

C'est un objet ayant pour propriétés :

```
public String err;
public int n;
public long[] id;
public long[] count;
public String[] login;
public String[] code;
public long[] status;
public long[] role;
public String[] firstname;
public String[] name;
public String[] mail;
public String[] phone;
public String[] extrafields;
public long[] createdby;
```

Cette fonction renvoie la liste des utilisateurs associés à un service.

La variable « n » renvoie le nombre d'entrées dans la liste.

Valeurs possibles de « code » :

- « ok » : le code de ce login a été consommé par un utilisateur. Ce login peut donc être considéré comme actif
- « expired » : le code n'a pas été consommé, il n'y a donc plus de code d'activation en cours de validité pour ce login.
- « in:code » : Le code de ce login n'a pas été consommé et est actuellement inactif. Utiliser loginActivateCode pour le rendre actif (voir loginCreate avec codetype=1)
- « link » : Le code n'a pas été consommé, un lien d'activation valable 3 semaines a été envoyé à l'adresse email précisée lors de la création du login (voir loginCreate avec codetype=2)

- « code à 9 chiffres » : le code d'activation en cours de validité (le login n'est pas encore relié à un utilisateur)

Les paramètres offset et nmax sont utilisés pour faire des requêtes paginées: offset est le premier champ récupéré (en commençant par 0), et nmax le nombre de lignes demandées.

Le paramètre « sort » vaut :

- 0 : aucun tri
- 1 : tri par login (ascendant)
- 2 : tri par login (descendant)
- 3 : tri par name (ascendant)
- 4 : tri par name (descendant)
- 5 : tri par mail (ascendant)
- 6 : tri par mail (descendant)

Le champ « count » retourne le nombre **total** d'utilisateurs du service, alors que le champ 'n' retourne le nombre d'utilisateurs renvoyés par la requête (donc n ≤ count).

Le champ « createdby » sert à distinguer les utilisateurs créés depuis la WebConsole In-Webo de ceux créés par l'API. Valeurs possibles :

- 0 : utilisateur créé par la WebConsole
- 1 : utilisateur créé par l'API

Le champ « extrafields » est réservé à un usage futur.

loginQuery

```
LoginQueryResult loginQuery(long userid, long loginid)
```

`Retourne :`

```
LoginQueryResult
```

C'est un objet ayant pour propriétés :

```
public String err;
public String login;
public String code;
public long status;
public long role;
public String firstname;
public String name;
public String mail;
public long createdby;
public long lastauthdate;     // timestamp de la dernière authentification réussie (0 si aucune)
public long nca;              //nb d'instances de l'application inWebo ou d'inWebo Helium
public long caid[];           //id de l'outil
public long castate[];        //etat de l'outil (0:actif, 1:ca bloqué; 2:pin bloqué)
public long caname[];         //nom de l'outil
public long caalias[];        //alias de l'outil (à corréler avec le retour de authenticateExtended)
public long cault[];          //type de l'ouil (0 : application; 1:Helium)
public long nma;              //nb d'applications mobiles
```

```
public long maid[];          //id de l'application mobile
public long mastate[];       //etat de l'application mobile (1=bloqué)
public long maname[];        //nom de l'application mobile
public long maalias[];       //alias de l'application mobile
public long nmac;            //nb de maccess
public long macid[];         //id du maccess
public long macstate[];      //état du maccess
public long macname[];       //nom du maccess
public long macalias[];      //alias du maccess
```

Cette fonction renvoie les paramètres d'un login précis.

Code vaut :

- « ok » : le compte est actif
- « in:code » : en cours de validation, le code est actuellement inactif
- « link » : en cours d'activation, via un lien
- le code d'activation, dans les autres cas

Role vaut :

- 0 : utilisateur
- 1 : manager du service (peut créer, modifier, supprimer des utilisateurs)
- 2 : administrateur du service (peut en plus modifier les paramètres du service)

Status vaut :

- 0 : utilisateur actif
- 1 : utilisateur inactif (les requêtes d'authentification seront rejetées)

loginSearch

```
loginSearch(long userid, long serviceid, String loginname, long exactmatch, long offset, long nmax, long sort)
```

Retourne :

```
LoginSerachResult
```

C'est un objet ayant pour propriétés :

```
public String err;
public int n;
public long[] id;
public String[] login;
public String[] code;
public long[] status;
public long[] role;
public String[] firstname;
public String[] name;
public String[] mail;
public String[] phone;
```

```
public long[] activation_status;
```

Cette fonction permet de rechercher l'utilisateur d'un service et d'en obtenir les informations.

exactmatch vaut :

- 0 : la recherche retournera tous les utilisateurs dont le login contient la chaîne loginname
- 1 : la recherche ne retournera que l'utilisateur dont le login est la chaîne exacte loginname

Le paramètre activation_status est un masque :

- 1 : L'utilisateur dispose d'au moins un nCode actif et non bloqué
- 2 : L'utilisateur dispose d'au moins une Toobar active et non bloqué

L'utilisation des paramètres offset, nmax et sort est identique à loginsQuery.

loginCreate

```
loginCreate(long userid, long serviceid, String login, String firstname, String name, String mail, String phone, long status, long role, long access, long codetype, String lang, String extrafields)
```

Retourne :

```
LoginCreateResult
```

C'est un objet ayant pour propriétés :

```
public String err;
public String code;
public long id;
```

La fonction provoque la création d'un code d'activation, que l'on peut consulter sur la console, ou bien envoyer par mail avec loginSendByMail. Ce code d'activation sera saisi par l'utilisateur final dans l'outil qu'il voudra activer. Celui-ci sera alors capable de générer des OTPs pour le service concerné.

Le fonctionnement dépend du paramètre « codetype » :

- 0 : un code d'activation valable 15 min est généré immédiatement
- 1 : un code d'activation inactif valable 3 semaines est généré (il pourra être activé avec loginActivateCode)
- 2 : un lien de génération de code d'activation, valable 3 semaines, est généré

Le paramètre lang vaut « fr » ou « en ».

- Avec l'offre Identity Guard, vous ne pouvez utiliser que la valeur codetype=0
- Avec l'offre Safe Transactions, vous pouvez utiliser codetype=0 ou 1
- Avec l'offre Entreprise, vous pouvez utiliser n'importe quelle valeur

Access vaut :

- 0 : les favoris du service ne sont pas attribués à cet utilisateur
- 1 : tous les favoris du service sont attribués à cet utilisateur

loginCreate retournera « NOK:full » si le quota d'utilisateurs a été atteint pour le service, et « NOK:loginexists » si le login existe déjà.

loginUpdate

loginUpdate(**long** userid, **long** serviceid, **long** loginid, **String** firstname, **String** name, **String** mail, **String** phone, **long** status, **long** role, **String** extrafields)

Retourne :

loginUpdateResponse

C'est un objet ayant pour propriétés :

public String loginUpdateReturn

Cette fonction permet de mettre à jour un utilisateur (un login).

La string retournée renvoie un code de résultat « OK » (succès) ou « NOK:<cause> » (erreur). La fonction retournera « NOK:login already used » si vous avez essayé de changer le login et que ce login existe déjà.

loginSendByMail

loginSendByMail(**long** userid, **long** serviceid, **long** loginid)

Retourne :

loginSendByMailResponse

C'est un objet ayant pour propriétés :

public String loginSendByMailReturn

Envoie le code d'activation au login précisé par email. Attention, celui-ci devra avoir une adresse email valide préalablement renseignée.

La string retournée renvoie un code de résultat « OK » (succès) ou « NOK » (erreur).

Disponible uniquement avec l'offre Entreprise.

loginActivateCode

loginActivateCode(**long** userid, **long** serviceid, **long** loginid)

Retourne :

loginActivateCodeResponse

C'est un objet ayant pour propriétés :

public String loginActivateCodeReturn

La string retournée renvoie le code d'activation du login ou « NOK » (erreur).

A utiliser pour les login créés par loginCreate avec codetype=1.

Disponible uniquement avec les offres Safe Transactions et Entreprise.

loginDelete

loginDelete(**long** userid, **long** serviceid, **long** loginid)

Retourne :

loginDeleteResponse

C'est un objet ayant pour propriétés :

public String loginDeleteReturn

Cette fonction permet de supprimer un utilisateur (un login).

La string retournée renvoie un code de résultat « OK » (succès) ou « NOK » (erreur).

loginResetPwd

loginResetPwd(**long** userid, **long** serviceid, **long** loginid)

Retourne :

loginResetPwdResponse

C'est un objet ayant pour propriétés :

public String loginResetPwdReturn

Cette fonction permet d'envoyer un code de déblocage à un utilisateur pour qu'il réinitialise son mot de passe lorsqu'il a bloqué un outil avec 3 codes PIN ou mots de passe faux.

La string retournée renvoie un code de résultat « OK » (succès) ou « NOK:<cause> » (erreur).

NB: Si l'utilisateur a plusieurs services, seul le service courant sera débloqué. Les autres, y compris My inWebo, ne seront pas accessibles depuis cet outil.

Disponible uniquement avec les offres Safe Transactions et Entreprise.

loginResetPwdExtended

loginResetPwdExtended(**long** userid, **long** serviceid, **long** loginid, **long** codetype)

Retourne :

loginResetPwdExtendedResponse

C'est un objet ayant pour propriétés :

public String loginResetPwdExtendedReturn

Cette fonction permet d'envoyer un code de déblocage à un utilisateur pour qu'il réinitialise son mot de passe lorsqu'il a bloqué un outil avec 3 codes PIN ou mots de passe faux.

Comparé à LoginResetPwd, cette fonction dispose en plus du paramètre « codetype » :

- 0 : un code de déblocage valable 15 min est généré immédiatement
- 1 : un code de déblocage inactif valable 3 semaines est généré (il pourra être activé avec loginActivateCode)
- 2 : un lien de déblocage, valable 3 semaines, est généré

La string retournée renvoie un code de résultat « OK » (succès) ou « NOK:<cause> » (erreur).

NB: Si l'utilisateur a plusieurs services, seul le service courant sera débloqué. Les autres, y compris My inWebo, ne seront pas accessibles depuis cet outil.

Disponible uniquement avec les offres Safe Transactions et Entreprise.

loginResetPINErrorCounter

```
loginResetPINErrorCounter(long userid, long serviceid, long loginid)
```

Retourne :

```
loginResetPINErrorCounterResponse
```

C'est un objet ayant pour propriétés :

```
public String loginResetPINErrorCounterReturn
```

Cette fonction permet de remettre à zéro le compteur du nombre de saisies de mot de passe ou de code PIN erronées. Elle n'est utilisable qu'une fois pour un utilisateur donné, jusqu'à la prochaine authentification réussie.

La string retournée renvoie un code de résultat « OK » (succès) ou « NOK:<cause> » (erreur).

Si le code est une erreur et vaut « NOK:MULTI », cela signifie que l'utilisateur est lié à d'autres services qui interdisent l'utilisation de cette fonction.

Disponible uniquement avec les offres Safe Transactions et Entreprise.

loginRestore

```
loginRestore(long userid, long serviceid, long loginid)
```

Retourne :

```
loginRestoreResponse
```

C'est un objet ayant pour propriétés :

```
public String LoginRestoreReturn
```

Cette fonction permet d'obtenir un code pour un utilisateur, qui lui permettra de rétablir son accès à votre service, par exemple lorsque tous ses outils sont bloqués ou perdus.

Quand le code de restauration est utilisé, tous les autres outils existants de l'utilisateur sont supprimés.

Seul son accès à VOTRE service sera rétabli. Tous les accès à des services tiers devront être rétablis un par un auprès de ces services.

La string retournée renvoie un code de résultat « OK » (succès) ou « NOK » (erreur).

Disponible uniquement avec les offres Safe Transactions et Entreprise.

loginAddDevice

```
loginAddDevice(long userid, long serviceid, long loginid, long codetype)
```

Retourne :

```
loginAddDeviceResponse
```

C'est un objet ayant pour propriétés :

```
public String loginAddDeviceReturn
```

Cette fonction permet de générer un code pour qu'un utilisateur active un outil de son choix.

La string retournée renvoie un code de résultat « OK » (succès) ou « NOK:<cause> » (erreur).

Si le code est une erreur et vaut :

- « NOK:NoPassword » : cela signifie que le mot de passe de l'utilisateur est vide.

NB: Si l'utilisateur a plusieurs services, seul le service courant sera ajouté. Les autres, y compris My inWebo, ne seront pas accessibles depuis cet outil.

Disponible uniquement avec les offres Safe Transactions et Entreprise.

loginDeleteTool

```
loginDeleteTool(long userid, long serviceid, long toolid, String tooltype)
```

Retourne :

```
loginAddDeviceResponse
```

C'est un objet ayant pour propriétés :

```
public String loginAddDeviceReturn
```

Cette fonction permet d'effacer l'outil d'un utilisateur. L'id et le type de l'outil à effacer sont obtenus par loginQuery. Type peut valoir 'ma' (nCode / Authenticator), 'ca' (inWebo pour PC / Helium) ou 'mac' (mAccess).

La string retournée renvoie un code de résultat « OK » (succès) ou « NOK:<cause> » (erreur).

Disponible uniquement avec les offres Safe Transactions et Entreprise.