

# SSO Bayard: Webservices

## [SSO Bayard: Webservices](#)

### [Webservices](#)

#### [Authentification / connexion aux WS](#)

#### [Accès aux webservices](#)

### [Compte client \(/user\)](#)

#### [Identification](#)

##### [Existence du compte](#)

##### [Login](#)

##### [Perte de mot de passe](#)

#### [Création de compte](#)

#### [Mot de passe](#)

#### [Stockage de données](#)

##### [Données générales](#)

##### [Données particulières & Namespace](#)

##### [Lecture](#)

##### [Ecriture](#)

#### [Comptes enfants](#)

#### [PARENT ACCOUNTS](#)

#### [CHILD LOGIN](#)

### [Lien avec Advantage \(/advantage\)](#)

#### [Recherche du numéro d'abonné](#)

#### [Lecture des accès et abonnements](#)

### [Exemples de code](#)

### [A documenter](#)

# 1 Webservices

La base utilisateur Bayard comporte un ensemble de webservices REST permettant d'agir avec elle.

## a Authentification / connexion aux WS

Chaque application ou site web se connectant aux webservices dispose de deux clés de 32 caractères permettant de sécuriser les échanges:

- une clé à passer en GET à chaque échange qui permet d'identifier la source de la demande et ses droits, réservée à certaines adresses IP: c'est l'**url\_key**
- une clé de chiffrement (salt pour MD5) permettant de crypter les informations confidentielles échangées (ex: mot de passe lors de la création d'un compte)

Ces deux clés permettent une meilleure sécurité en permettant l'utilisation sur de nombreux supports et langages (uniquement des fonctions MD5)

## b Accès aux webservices

L'accès aux webservices se fait en REST, en passant à chaque fois le module appelé (compte utilisateur, paiement, etc.) et la clé en paramètre GET (pas encore de sticky sessions)

Les webservices servent une réponse formatée en XML.

Exemple: pour envoyer un e-mail de changement de mot de passe à l'utilisateur (module **user**), le format d'appel HTTP est le suivant:

`http://bayardsso.fr/api/user/forgot_password/:url_key/:email`

Les appels doivent être url encodés pour respecter le protocole HTTP.  
Exemple avec l'url ci-dessus:

[http://preprod.bayardsso.fr/api/user/forgot\\_password/58867ff8e5c617b7eec5117d9d860591/cedric%40nilly.fr](http://preprod.bayardsso.fr/api/user/forgot_password/58867ff8e5c617b7eec5117d9d860591/cedric%40nilly.fr)

Chaque réponse des webservices Userbase contient un status, qui permet de connaître l'état de la requête: un statut success indique que l'application SSO a fonctionné correctement sur cette requête REST. Exemple après un login:

```
<ApiUser generator="zend" version="1.0">
  <login>
    <response>RESPONSE_FAILURE</response>
    <status>success</status>
  </login>
</ApiUser>
```

Les webservices ont fonctionné correctement, mais la réponse à la question est négative (ici, tentative de login).

## c Compte client (/user)

### i Identification

#### 1 Existence du compte

Il est possible de savoir si un compte existe via l'URL d'appel suivante:

[http://bayardsso.fr/api/user/lookup/:url\\_key/:email](http://bayardsso.fr/api/user/lookup/:url_key/:email)

qui peut aussi être appelée avec un numéro de compte SSO en lieu et place de l'e-mail:

[http://bayardsso.fr/api/user/lookup/:url\\_key/:id](http://bayardsso.fr/api/user/lookup/:url_key/:id)

La réponse contiendra les informations sur le compte, ou un statut négatif; il y a trois cas différents:

- Exemple 1: ce compte existe, et est un compte **full** (la personne a choisi un mot de passe et a entré plusieurs informations):

```
<ApiUser generator="zend" version="1.0">
  <lookup>
    <response>RESPONSE_SUCCESS</response>
    <user>
      <email>cedric@nilly.fr</email>
      <id>33</id>
      <registered>FULL</registered>
    </user>
    <status>success</status>
  </lookup>
</ApiUser>
```

- Exemple 2: Le compte suivant est un prospect: nous avons collecté son adresse e-mail, mais l'internaute n'a pas véritablement créé de compte:

```
<ApiUser generator="zend" version="1.0">
  <lookup>
    <response>RESPONSE_SUCCESS</response>
    <user>
      <email>cedric2@nilly.fr</email>
      <id>11</id>
      <registered>NO_PASSWORD</registered>
    </user>
    <status>success</status>
  </lookup>
</ApiUser>
```

- Exemple 3: Ce compte n'existe tout simplement pas dans la UserBase:

```
<ApiUser generator="zend" version="1.0">
  <lookup>
```

```
<response>RESPONSE_FAILURE</response>
<status>success</status>
</lookup>
</ApiUser>
```

## 2 Login

Il est possible de tester la validité d'un couple e-mail / mot de passe pour identifier un internaute sur un site ou une application.  
Le mot de passe doit être crypté en md5 en ajoutant le salt (la clé de cryptage 32 caractères associée à chaque partenaire), selon la formule:  
`md5PasswordSalted = md5 ( salt32 . password_user )`

Il suffit alors d'utiliser l'appel REST du module /user:

`http://bayardsso.fr/api/user/login/:url_key/:email/:md5PasswordSalted`

- Exemple 1: Login réussi, l'application peut donc sauver la session de l'utilisateur et le considérer comme identifié:

```
<ApiUser generator="zend" version="1.0">
  <login>
    <response>RESPONSE_SUCCESS</response>
    <status>success</status>
  </login>
</ApiUser>
```

- Exemple 2: Identification ratée: le site ou l'application doit afficher un message d'erreur à l'utilisateur et lui demander de s'identifier à nouveau:

```
<ApiUser generator="zend" version="1.0">
  <login>
    <response>RESPONSE_FAILURE</response>
    <status>success</status>
  </login>
</ApiUser>
```

Note: il n'y a pas de distinction entre utilisateur non trouvé et mot de passe invalide. Si besoin, ce point doit être vu avec le précédent webservice.

La UserBase ne gère donc qu'une réponse, et en aucun cas une session de son côté, à gérer du côté applicatif.

## 3 Perte de mot de passe

En cas de perte de mot de passe, il est possible de demander à la UserBase d'envoyer un e-mail à l'utilisateur. Cet e-mail contiendra un lien sécurisé lui permettant de changer son mot de passe:

`http://bayardsso.fr/api/user/forgot_password/:url_key/:email`

Deux cas de figure: l'utilisateur est trouvé, l'e-mail est envoyé, ou l'utilisateur est impossible à retrouver dans la UserBase:

```
<ApiUser generator="zend" version="1.0">
  <forgot_password>
    <response>RESPONSE_SUCCESS</response>
    <status>success</status>
  </forgot_password>
</ApiUser>
```

OU:

```
<ApiUser generator="zend" version="1.0">
  <forgot_password>
    <response>RESPONSE_FAILURE</response>
    <status>success</status>
  </forgot_password>
</ApiUser>
```

## ii Création de compte

La création de compte, comme toutes les interactions en écriture, nécessite le compte (e-mail) en GET, et on peut y ajouter à la volée des données via un passage d'informations en POST (voir plus loin: [Ecriture de données](#)), suivant le schéma d'URL:

`http://bayardsso.fr/api/user/create/:url_key/:email`

- Exemple 1: création de compte réussie.  
`http://bayardsso.fr/api/user/create/:url_key/cedric@nilly.fr`

Réponse:

```
<ApiUser generator="zend" version="1.0">
  <create>
    <response>RESPONSE_SUCCESS</response>
    <user>
      <email>someone@smile.fr</email>
      <id>51</id>
    </user>
    <status>success</status>
  </create>
</ApiUser>
```

- Exemple 2: l'utilisateur existe déjà.  
`http://bayardsso.fr/api/user/create/:url_key/cedric@nilly.net`

Réponse:

```
<ApiUser generator="zend" version="1.0">
  <create>
    <response>RESPONSE_FAILURE</response>
    <user>ERROR_USER_EXISTS</user>
    <status>success</status>
  </create>
</ApiUser>
```

### iii Mot de passe

Le mot de passe est une données particulièrement sensible, c'est pour cela que l'on doit s'assurer de la crypter au login (pas lors de sa première définition) et qu'une méthode lui est dédié:

[http://bayardsso.fr/api/user/set\\_password/:url\\_key/:email](http://bayardsso.fr/api/user/set_password/:url_key/:email) en passant en POST le "password" choisi ~~et crypté via la seconde clé fournie au compte développeur.~~ en clair pour le moment.

Ce mot de passe doit être crypté suivant la méthode

Exemple 1: définition ou changement de mot de passe validé:

```
<ApiUser generator="zend" version="1.0">
  <set_password>
    <response>RESPONSE_SUCCESS</response>
    <status>success</status>
  </set_password>
</ApiUser>
```

Exemple 2: erreur commune:

```
<ApiUser generator="zend" version="1.0">
  <set_password>
    <response>RESPONSE_FAILURE</response>
    <status>success</status>
  </set_password>
</ApiUser>
```

Exemple 3: utilisation inconnu:

```
<ApiUser generator="zend" version="1.0">
  <set_password>
    <response>RESPONSE_FAILURE</response>
    <user>ERROR_USER_NOT_EXIST</user>
    <status>success</status>
  </set_password>
</ApiUser>
```

### iv Stockage de données

La base utilisateur stocke les informations de connexions (e-mail, mot de passe) des internautes possédant un compte Bayard, mais également une multitude d'informations associées au compte (comme le numéro d'abonné Advantage, ou des informations démographiques: âge, civilité, etc.)

Suivant l'`url_key` utilisée pour se connecter, l'accès se fera en lecture, en écriture ou pas du tout.

## 1 Données générales

Certaines données doivent être partagées entre tous les sites: ce sont les données générales (on parle aussi de namespace "`root`").

Exemple: âge, date de naissance, prénom sont des informations qui doivent être identiques à tous les sites ou applications utilisant les webservices UserBase.

## 2 Données particulières & Namespace

D'autres données sont plus spécifiques à un site ou une application en particulier, voire un marché. Par exemple, les sites de marque jeunesse peuvent partager un pseudo. Mais La-croix.com utilisera également un champ pseudo, mais différent.

C'est pour cela qu'a été mise en place la notion de namespace: pour ces données particulières à des environnements, on préfixe les clés d'un nom défini auparavant. Exemple: BJ pour l'ensemble des sites Bayard Jeunesse, ou CRX pour les données particulières à La Croix. Ainsi, nous pouvons gérer ces deux pseudos pour un même utilisateur, via les couples clés/valeur. On sépare le namespace de la clé par un double underscore.

CRX\_\_PSEUDO comportera ainsi le pseudo destiné à La Croix,  
BJ\_\_PSEUDO celui destiné aux sites Jeunesse.

Suivant les autorisations définies dans la Userbase selon l'`url_key`, une application ou un site pourra ou pas lire et écrire ces deux informations.

## 3 Lecture

Il est possible de lire plusieurs champs d'un même compte via la requête-type:  
`http://bayardsso.fr/api/user/get/:url_key/:email/:fields,comma,separated`

Pour chacun des champs, la réponse XML contiendra une indication: la valeur du champ ou un message d'erreur si celui n'existe pas ou si les droits de consultation en lecture de l'`url_key` ne sont pas suffisants.

En cas d'erreur non bloquante (exemple, un des champs demandés n'existe pas), un message **RESPONSE\_WARNING** sera adressé, à l'application de choisir de continuer ou de traiter ces erreurs.

- Exemple 1: utilisateur non trouvé:

[http://preprod.bayardsso.fr/api/user/get/58867ff8e5c617b7eec5117d9d860591/cedric%40nilly.com/POSTAL\\_CITY,POSTAL\\_CODE,LACROIX\\_\\_PSEUDO,%20NAMESPACE\\_\\_PERMS\\_TEST](http://preprod.bayardsso.fr/api/user/get/58867ff8e5c617b7eec5117d9d860591/cedric%40nilly.com/POSTAL_CITY,POSTAL_CODE,LACROIX__PSEUDO,%20NAMESPACE__PERMS_TEST)

Réponse:

```
<ApiUser generator="zend" version="1.0">
  <get>
    <response>RESPONSE_FAILURE</response>
    <status>success</status>
  </get>
</ApiUser>
```

- Exemple 2: multiples champs, dont certains avec erreurs:

[http://preprod.bayardsso.fr/api/user/get/58867ff8e5c617b7eec5117d9d860591/cedric%40nilly.fr/POSTAL\\_CITY,,POSTAL\\_CODE,LACROIX\\_PSEUDO,NAMESPACE\\_\\_PERMS\\_TEST,BAD\\_FIELD](http://preprod.bayardsso.fr/api/user/get/58867ff8e5c617b7eec5117d9d860591/cedric%40nilly.fr/POSTAL_CITY,,POSTAL_CODE,LACROIX_PSEUDO,NAMESPACE__PERMS_TEST,BAD_FIELD)

#### Réponse:

```
<ApiUser generator="zend" version="1.0">
  <get>
    <response>RESPONSE_WARNING</response>
    <user>
      <email>cedric@nilly.fr</email>
      <id>33</id>
    </user>
    <fields>
      <POSTAL_CITY>montreuil</POSTAL_CITY>
      <POSTAL_CODE>93100</POSTAL_CODE>
      <LACROIX_PSEUDO/>
      <NAMESPACE__PERMS_TEST>ERROR_EAV_FIELD_PERMISSION</NAMESPAC
E__PERMS_TEST>
      <BAD_FIELD>ERROR_EAV_FIELD_NONEXISTENT</BAD_FIELD>
    </fields>
    <status>success</status>
  </get>
</ApiUser>
```

#### o Exemple 3: tout est correct.

[http://preprod.bayardsso.fr/api/user/get/58867ff8e5c617b7eec5117d9d860591/cedric%40nilly.fr/POSTAL\\_CITY,,POSTAL\\_CODE,LACROIX\\_PSEUDO](http://preprod.bayardsso.fr/api/user/get/58867ff8e5c617b7eec5117d9d860591/cedric%40nilly.fr/POSTAL_CITY,,POSTAL_CODE,LACROIX_PSEUDO)

#### Réponse:

```
<ApiUser generator="zend" version="1.0">
  <get>
    <response>RESPONSE_SUCCESS</response>
    <user>
      <email>cedric@nilly.fr</email>
      <id>33</id>
    </user>
    <fields>
      <POSTAL_CITY>montreuil</POSTAL_CITY>
      <POSTAL_CODE>93100</POSTAL_CODE>
      <LACROIX_PSEUDO/>
    </fields>
    <status>success</status>
  </get>
</ApiUser>
```

## 4 Ecriture

Suivant les droits d'écriture accordés à l'utilisateur de l'url\_key, des champs généraux ("root") ou d'un namespace en particulier peuvent être écrits.

Il suffit alors d'appeler la méthode set avec les paramètres en POST, en suivant le schéma d'URL:

[http://bayardsso.fr/api/user/set/:url\\_key/:email](http://bayardsso.fr/api/user/set/:url_key/:email)



### Exemple d'une requête complète:

```
POST /api/user/set/58867ff8e5c617b7eec5117d9d860591/cedric%40nilly.fr HTTP/1.0
User-Agent: HTTPTool/1.0
Content-Type: application/x-www-form-urlencoded
NAME=Nilly&FIRSTNAME=Cédric&CRX_PSEUDO=cnilly
```

Plusieurs cas de figure peuvent alors se présenter:

- **Ecriture ok:**

```
<ApiUser generator="zend" version="1.0">
  <set>
    <response>RESPONSE_SUCCESS</response>
    <status>success</status>
  </set>
</ApiUser>
```

- **L'utilisateur n'existe pas:**

```
<ApiUser generator="zend" version="1.0">
  <set>
    <response>RESPONSE_FAILURE</response>
    <user>ERROR_USER_NOT_EXIST</user>
    <status>success</status>
  </set>
</ApiUser>
```

- **Certains champs ont posé problème:**

```
<ApiUser generator="zend" version="1.0">
  <set>
    <response>RESPONSE_WARNING</response>
    <fields>
      <NONEXISTENT_FIELD>ERROR_EAV_FIELD_NONEXISTENT</NONEXISTENT_FIELD>
      <NAMESPACE__PERMS_TEST>ERROR_EAV_FIELD_PERMISSION
    </NAMESPACE__PERMS_TEST>
      <FAILS_VALIDATION>
        <validation_error>
          <value>MyValue</value>
          <description>The value must be 'good'; The value must
be 'bad'</description>
        </validation_error>
      </FAILS_VALIDATION>
    </fields>
    <status>success</status>
  </set>
</ApiUser>
```

## 1 Comptes enfants

### a CREATE CHILD

[http://preprod.bayardssso.fr/api/user/create\\_child/:url\\_key/:pseudo/:code\\_secret/:identifier](http://preprod.bayardssso.fr/api/user/create_child/:url_key/:pseudo/:code_secret/:identifier)

[http://preprod.bayardssso.fr/api/user/create\\_child/58867ff8e5c617b7eec5117d9d860591/car001/caro01/lotal@smile.fr](http://preprod.bayardssso.fr/api/user/create_child/58867ff8e5c617b7eec5117d9d860591/car001/caro01/lotal@smile.fr)

ALL OK :

```
<ApiUser generator="zend" version="1.0">
  <create_child>
    <response>RESPONSE_SUCCESS</response>
    <status>success</status>
  </create_child>
</ApiUser>
```

ERROR :

```
<ApiUser generator="zend" version="1.0">
  <create_child>
    <response>RESPONSE_FAILURE</response>
    <status>success</status>
  </create_child>
</ApiUser>
```

## CHILD PSEUDO EXIST

bayardsso.fr/api/user/child\_pseudo\_exist/:url\_key/:pseudo

[http://preprod.bayardsso.fr/api/user/child\\_pseudo\\_exist/58867ff8e5c617b7eec5117d9d860591/caro000](http://preprod.bayardsso.fr/api/user/child_pseudo_exist/58867ff8e5c617b7eec5117d9d860591/caro000)

ALL OK :

```
<ApiUser generator="zend" version="1.0">
  <child_pseudo_exist>
    <response>RESPONSE_SUCCESS</response>
    <status>success</status>
  </child_pseudo_exist>
</ApiUser>
```

ERROR :

```
<ApiUser generator="zend" version="1.0">
  <child_pseudo_exist>
    <response>RESPONSE_FAILURE</response>
    <status>success</status>
  </child_pseudo_exist>
</ApiUser>
```

## CHILD ACCOUNTS

[http://bayardsso.fr/api/user/child\\_accounts/:url\\_key/:identifier/:fields](http://bayardsso.fr/api/user/child_accounts/:url_key/:identifier/:fields)

[http://preprod.bayardsso.fr/api/user/child\\_accounts/5f9747407e58b5497a79ff10cf56bf99/f00@bar.com/NAME,FIRSTNAME](http://preprod.bayardsso.fr/api/user/child_accounts/5f9747407e58b5497a79ff10cf56bf99/f00@bar.com/NAME,FIRSTNAME)

### ALL OK :

```
<ApiUser generator="zend" version="1.0">
  <child_accounts>
    <key_0>
      <id>1621600</id>
      <pseudo>carol0</pseudo>
      <secretcode/>
      <fields>
        <NAME/>
        <FIRSTNAME/>
      </fields>
    </key_0>
    <key_1>
      <id>1621610</id>
      <pseudo>caro3</pseudo>
      <secretcode/>
      <fields>
        <NAME/>
        <FIRSTNAME/>
      </fields>
    </key_1>
    <response>RESPONSE_SUCCESS</response>
    <status>success</status>
  </child_accounts>
</ApiUser>
```

### NO RESULT :

```
<ApiUser generator="zend" version="1.0">
  <child_accounts>
    <response>RESPONSE_SUCCESS</response>
    <status>success</status>
  </child_accounts>
</ApiUser>
```

### ERROR :

```
<ApiUser generator="zend" version="1.0">
```

```
<child_accounts>
  <response>RESPONSE_FAILURE</response>
  <status>success</status>
</child_accounts>
</ApiUser>
```

## PARENT ACCOUNTS

[http://bayardsso.fr/api/user/parent\\_accounts/:url\\_key/:identifier](http://bayardsso.fr/api/user/parent_accounts/:url_key/:identifier)

[http://preprod.bayardsso.fr/api/user/parent\\_accounts/58867ff8e5c617b7eec5117d9d860591/caro2](http://preprod.bayardsso.fr/api/user/parent_accounts/58867ff8e5c617b7eec5117d9d860591/caro2)

ALL OK :

```
<ApiUser generator="zend" version="1.0">
  <parent_accounts>
    <key_0>
      <id>60</id>
      <email>alan.wagner@smile.fr</email>
    </key_0>
    <response>RESPONSE_SUCCESS</response>
    <status>success</status>
  </parent_accounts>
</ApiUser>
```

ERROR :

```
<ApiUser generator="zend" version="1.0">
  <parent_accounts>
    <response>RESPONSE_FAILURE</response>
    <status>success</status>
  </parent_accounts>
</ApiUser>
```

## CHILD LOGIN

[http://bayardsso.fr/api/user/child\\_login/:url\\_key/:pseudo/:code\\_secret](http://bayardsso.fr/api/user/child_login/:url_key/:pseudo/:code_secret)

[http://preprod.bayardsso.fr/api/user/child\\_login/5f9747407e58b5497a79ff10cf56bf99/caro/caro](http://preprod.bayardsso.fr/api/user/child_login/5f9747407e58b5497a79ff10cf56bf99/caro/caro)

## ALL OK :

```
<ApiUser generator="zend" version="1.0">
  <child_login>
    <response>RESPONSE_SUCCESS</response>
    <status>success</status>
  </child_login>
</ApiUser>
```

## ERROR :

```
<ApiUser generator="zend" version="1.0">
  <child_login>
    <response>RESPONSE_FAILURE</response>
    <status>success</status>
  </child_login>
</ApiUser>
```

## **b Lien avec Advantage (/advantage)**

Les webservices SSO encapsulent des liens Advantage, simplifiant l'accès au système de gestion des abonnements papier et numérique.

Le lien avec Advantage se fait au travers d'un numéro d'abonné, qui est stocké dans le champ ADVANTAGE\_NUMBER.

Pour les comptes possédant déjà un numéro d'abonné, l'accès aux informations se fait directement via les webservices adéquats. Pour les comptes ne bénéficiant pas encore de numéro d'abonné et désirant obtenir des avantages numériques suite à leur abonnement papier, il va falloir retrouver ce numéro d'abonné, vérifier qu'il est correct (tenter d'éviter autant que possible une usurpation), puis le mettre à jour.

### **i Recherche du numéro d'abonné**

La recherche s'effectue en appelant un webservice avec les éléments suivants:

- nom du titulaire de l'abonnement
- prénom du titulaire
- code postal
- pays (FR pour France)

L'URL est du type:

[http://bayardsso.fr/api/advantage/search/:url\\_key/:name/:firstname/:postal\\_code/:country](http://bayardsso.fr/api/advantage/search/:url_key/:name/:firstname/:postal_code/:country)

Trois cas possibles se présentent alors.

#### **1 Aucun compte n'est retrouvé**

Il faut alors préciser au client que la recherche n'a rien donné. Le titulaire de l'abonnement est peut-être un autre membre du foyer ? (Exemple: cas d'un abonnement offert à un enfant) Si l'impossibilité de retrouver le compte persiste, il faut envoyer le client vers le service client qui pourra, manuellement, associer le compte au bon numéro d'abonné ou

Exemple:

<http://preprod.bayardsso.fr/api/advantage/search/58867ff8e5c617b7eec5117d9d860591/Avril/Frederic/42290/UK>

```
<ApiAdvantage generator="zend" version="1.0">
  <search>
    <SearchResult>
```

```
<SearchStatus>NoMatch</SearchStatus>
<CustomerNumber/>
<AddressCode/>
</SearchResult>
<status>success</status>
</search>
</ApiAdvantage>
```

### 3 Un compte et un seul est retrouvé

Le compte du client est bien retrouvé: il faut donc associer ce CustomerNumber au champ SSO ADVANTAGE\_\_NUMBER.

Par la suite, l'ensemble des webservices d'accès sera bien disponible pour compte client SSO.

[http://preprod.bayardsso.fr/api/advantage/search\\_many/58867ff8e5c617b7eec5117d9d860591/Nilly/Cedric/93130/FR](http://preprod.bayardsso.fr/api/advantage/search_many/58867ff8e5c617b7eec5117d9d860591/Nilly/Cedric/93130/FR)

```
<ApiAdvantage generator="zend" version="1.0">
  <search>
    <SearchResult>
      <SearchStatus>SingleMatch</SearchStatus>
      <CustomerNumber>000000032893</CustomerNumber>
      <AddressCode>ADR1</AddressCode>
    </SearchResult>
    <status>success</status>
  </search>
</ApiAdvantage>
```

### 4 Plusieurs comptes correspondent

Il est alors impossible d'identifier - pour le moment - le bon compte, il faut rediriger l'internaute vers le service client.

<http://preprod.bayardsso.fr/api/advantage/search/.../Ly/Celine/77185/FR>

```
<ApiAdvantage generator="zend" version="1.0">
  <search>
    <SearchResult>
      <SearchStatus>MultipleMatches</SearchStatus>
      <CustomerNumber/>
      <AddressCode/>
    </SearchResult>
    <status>success</status>
  </search>
</ApiAdvantage>
```

## ii Lecture des accès et abonnements

Les abonnements sont de deux nature: les abonnements papier et les abonnements numériques.

Le webservice /access permet de lire de manière brute l'ensemble des abonnements d'un compte SSO; charge ensuite au site ou à l'application d'agir en fonction de cette liste: accès à un univers premium par exemple.

<http://www.bayardsso.fr/api/access/get/:key/:email/BAYARD>

Le noeud XML `SubscriptionAccessList` contient la liste des abonnements papier.

Le noeud XML `ContentAccessList` contient la liste des abonnements numériques, suivant le type de "ServiceId":

Site web	Service Id renvoyé par les webservices
Croire.com	"CROIRE COM"
La-Croix.com	"LA CROIX COM"

Ici, un abonnement à La-Croix.com valable jusqu'au 08 mai 2012:

<http://preprod.bayardsso.fr/api/access/get/58867ff8e5c617b7eec5117d9d860591/cedric@nilly.fr/BAYARD>

```
<ApiAccess generator="zend" version="1.0">
  <SCRIPT/>
  <get>
    <CustomerAccess>
      <SubscriptionAccessList>
        <SubscriptionAccess>...</SubscriptionAccess>
      </SubscriptionAccessList>
      <ProductAccessList/>
      <ContentAccessList>
        <ContentAccess>
          <HostCode>BAYARD</HostCode>
          <StartDate>2012-03-09T00:00:00</StartDate>
          <ExpireDate>2012-05-08T00:00:00</ExpireDate>
          <ServiceId>LA CROIX COM</ServiceId>
          <AccessItem>ACCES</AccessItem>
          <RateCode>OF01-001</RateCode>
        </ContentAccess>
      </ContentAccessList>
    </CustomerAccess>
    <status>success</status>
  </get>
</ApiAccess>
```



## Exemples de code

- Est-ce qu'un compte existe ?

```
function SSOExists($email) {  
  
    // http://bayardsso.fr/api/user/lookup/:url_key/:email  
    $result = simplexml_load_file( SSO_HOST . '/user/lookup/' . SSO_KEY . '/' .  
urlencode($email) );  
  
    if ( isset($result) && isset($result->lookup) && isset($result->lookup->user) &&  
isset($result->lookup->user->id) ) {  
        return $result->lookup->user->id;  
    }  
    return false;  
}
```

- Création de compte

```
function SSOCreate($email) {  
    // http://bayardsso.fr/api/user/create/:url_key/cedric@nilly.fr  
    $result = simplexml_load_file( SSO_HOST . '/user/create/' . SSO_KEY . '/' .  
urlencode($email) );  
  
    if ( isset($result) && isset($result->create) && isset($result->create->  
user) && isset($result->create->user->id) ) {  
        return $result->create->user->id;  
    }  
    return false;  
}
```

- Définir un mot de passe

```
// set password  
$c = curl_init();  
curl_setopt($c, CURLOPT_URL, self::SSO_URL . '/user/set_password/' . self::SSO_KEY . '/' .  
urlencode($username) );  
curl_setopt($c, CURLOPT_RETURNTRANSFER, true);  
curl_setopt($c, CURLOPT_HEADER, false);  
curl_setopt($c, CURLOPT_POST, true);  
curl_setopt($c, CURLOPT_POSTFIELDS, array(  
    'password' => $password  
));  
$output = curl_exec($c);  
$result = simplexml_load_string($output);  
if ( 'RESPONSE_SUCCESS'==$result->set_password->response ) {  
    echo "setPassword OK ({$password})\n";  
    return $password;  
} else {  
    echo "setPassword error\n";  
    return false;  
}
```

- Ecrire dans un champ

```
function setContent($email, $field, $value) {  
  
    $ch = curl_init();  
    curl_setopt($ch, CURLOPT_URL, SSOURL.'/api/user/set/'.SSOKEY.'/'.urlencode($email) );  
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);  
    curl_setopt($ch, CURLOPT_POST, true);  
    $data = array(  
        urlencode( strtoupper($field) ) => $value,  
    );  
    curl_setopt($ch, CURLOPT_POSTFIELDS, $data);  
    $output = curl_exec($ch);  
    curl_close($ch);  
  
    $t = simplexml_load_string($output);  
    if ($t->set->response=='RESPONSE_SUCCESS') {  
        return true;  
    } else {  
        return false;  
    }  
}
```

## A documenter

Récupérer les abonnements Advantage d'un compte SSO:

<http://preprod.bayardsso.fr/api/access/get/58867ff8e5c617b7eec5117d9d860591/cedric@nilly.fr/BAYARD>

1) api/advantage/all\_subscriptions/:url\_key/:identifiant ( email ou user\_id )

[http://preprod.bayardsso.fr/api/advantage/all\\_subscriptions/0954a890ff38e12bc92b7045d9d66c7b/famille@lamyriquois.com](http://preprod.bayardsso.fr/api/advantage/all_subscriptions/0954a890ff38e12bc92b7045d9d66c7b/famille@lamyriquois.com)

Cas d'erreur :

[http://preprod.bayardsso.fr/api/advantage/all\\_subscriptions/0954a890ff38e12bc92b7045d9d66c7b/XXX@lamyriquois.com](http://preprod.bayardsso.fr/api/advantage/all_subscriptions/0954a890ff38e12bc92b7045d9d66c7b/XXX@lamyriquois.com)

2) api/advantage/all\_subscriptions\_an/:url\_key/:advantage\_id

[http://preprod.bayardsso.fr/api/advantage/all\\_subscriptions\\_an/0954a890ff38e12bc92b7045d9d66c7b/20363710](http://preprod.bayardsso.fr/api/advantage/all_subscriptions_an/0954a890ff38e12bc92b7045d9d66c7b/20363710)

Rien trouvé :

[http://preprod.bayardsso.fr/api/advantage/all\\_subscriptions\\_an/0954a890ff38e12bc92b7045d9d66c7b/20363710999](http://preprod.bayardsso.fr/api/advantage/all_subscriptions_an/0954a890ff38e12bc92b7045d9d66c7b/20363710999)

Cas d'erreur (plantage) :

[http://preprod.bayardsso.fr/api/advantage/all\\_subscriptions\\_an/0954a890ff38e12bc92b7045d9d66c7b/00004500027](http://preprod.bayardsso.fr/api/advantage/all_subscriptions_an/0954a890ff38e12bc92b7045d9d66c7b/00004500027)

3) Egalement:

api/advantage/subscription\_list\_active  
api/advantage/subscription\_gifts\_list\_active

[http://preprod.bayardsso.fr/api/advantage/subscription\\_list\\_active/0954a890ff38e12bc92b7045d9d66c7b/20363710](http://preprod.bayardsso.fr/api/advantage/subscription_list_active/0954a890ff38e12bc92b7045d9d66c7b/20363710)

[http://preprod.bayardsso.fr/api/advantage/subscription\\_gifts\\_list\\_active/0954a890ff38e12bc92b7045d9d66c7b/20363710](http://preprod.bayardsso.fr/api/advantage/subscription_gifts_list_active/0954a890ff38e12bc92b7045d9d66c7b/20363710)

4) L'existant :

api/advantage/subscription\_list  
api/advantage/subscription\_gifts\_list

[http://preprod.bayardsso.fr/api/advantage/subscription\\_list/0954a890ff38e12bc92b7045d9d66c7b/20363710](http://preprod.bayardsso.fr/api/advantage/subscription_list/0954a890ff38e12bc92b7045d9d66c7b/20363710)

[http://preprod.bayardsso.fr/api/advantage/subscription\\_gifts\\_list/0954a890ff38e12bc92b7045d9d66c7b/20363710](http://preprod.bayardsso.fr/api/advantage/subscription_gifts_list/0954a890ff38e12bc92b7045d9d66c7b/20363710)

5) obsolète:

<http://preprod.bayardsso.fr/api/advantage/search/58867ff8e5c617b7eec5117d9d860591/Nilly/Cedric/93130/FR> (search qui ne renvoie pas les MultipleMatches)