

# Présentation des Personal APIs Orange : exemple d'Authentification et de Calendrier

par Fabien Venries Karim Sbata

Date de publication : 16/06/2008

Dernière mise à jour :

Les Personal APIs d'Orange permettent aux concepteurs de sites web d'offrir des fonctions avancées aux utilisateurs Orange France, en leur proposant d'interagir directement avec leurs outils et données personnelles. Il est ainsi possible pour un utilisateur d'ajouter directement des contacts à son carnet d'adresse Orange, de s'authentifier et partager des informations de profil, d'utiliser ses photos stockées sur Orange photo, etc.

I - Introduction.....	3
II - Préambule au développement.....	3
II-A - Authentification et Privacy.....	3
II-B - Descriptif rapide des APIs.....	4
III - API d'Authentification.....	4
III-A - Requête.....	4
III-B - Réponse.....	5
IV - Ajouter un évènement au Calendrier.....	7
IV-A - Requête.....	7
IV-B - Gestion de l'authentification et de la Privacy.....	8
IV-C - Réponse.....	9
V - Conclusion.....	10
VI-A - À vous de jouer.....	10
VI-B - Pour aller plus loin.....	10

## I - Introduction

Depuis mi avril, Orange France met à disposition des développeurs des APIs permettant à un site web d'offrir à ses clients d'interagir avec ses outils Orange. Ces APIs sont présentées sur le site d'**Orange Partner**

Cet article se propose de vous présenter le concept de **Personal APIs** (1) à travers quelques exemples.

### Les API proposées par Orange vous permettent notamment :

- D'utiliser le système d'authentification Orange pour gérer vos utilisateur web et les authentifier de manière sécurisée ;
- De récupérer leur profil avec leur consentement afin de pouvoir préremplir plus aisément des formulaires ;
- D'interagir avec le carnet d'adresse des utilisateurs pour ajouter un contact, leur permettre de rechercher dans leur carnet d'adresse un contact particulier, afin, par exemple, de lui envoyer un email ;
- De proposer l'export de vos photos directement vers les albums des clients Orange ;
- D'ajouter un événement au calendrier des utilisateurs ;
- Et bien d'autres fonctionnalités dans le futur.

## II - Préambule au développement

Ce chapitre a pour objectif de présenter l'ensemble des prérequis pour commencer le développement d'un service utilisant les Personal APIs Orange.

### II-A - Authentification et Privacy

Avant de commencer tout développement, il est nécessaire de comprendre quelques concepts.

Les Personal APIs permettent de créer un site interagissant avec les données utilisateurs, et pour cette raison, il est nécessaire de mettre en oeuvre deux concepts : l'authentification, afin de savoir quel est l'utilisateur à qui on s'adresse, et la Privacy, outils de gestion du partage des données personnelles, protégeant l'utilisateur contre un usage abusif.

### Le scénario global d'usage des APIs est le suivant (exemple du Calendrier) :

- Un utilisateur navigue sur votre site et veut que vous ajoutiez un évènement à son calendrier.
- Afin d'identifier le client, vous le redirigez vers une URL d'authentification.
- Une fois le client authentifié, il est redirigé vers l'URL du site avec un jeton qui sera utilisé pour identifier l'utilisateur lors de la requête aux Personal APIs.
- Ce jeton d'utilisateur est temporaire et devra, après expiration, être renouvelé de la même manière.
- Le site fait la requête à la fonction des Personal APIs en utilisant le jeton de l'utilisateur.
- Si l'utilisateur n'a pas encore autorisé l'accès à ses données personnelles, l'appel déclenchera un message d'erreur "Privacy" suite auquel le service pourra inviter l'utilisateur à donner son consentement en le redirigeant vers l'URL de Privacy et il lui sera demandé de donner un consentement temporaire ou permanent avant d'être redirigé vers votre service.
- Après l'interaction de Privacy avec l'utilisateur, votre service peut à nouveau appeler la Personal API.
- Si l'utilisateur a autorisé l'accès, le service reçoit la réponse de la Personal API.

Afin de pouvoir utiliser ces services, le développeur doit se rendre sur le site d'**Orange Partner**, s'identifier, puis demander sa souscription via un formulaire, après avoir validé les conditions d'utilisation. Il pourra souscrire indépendamment à chaque API de la suite d'APIs.

## Notez qu'il est nécessaire de prévoir :

- Une URL pour la redirection après authentification ;
- Une adresse publique du site web, et un logo, pour que votre site apparaisse dans le module de gestion de Privacy.

## II-B - Descriptif rapide des APIs

- L'**Authentification API** permet la mise en oeuvre des fonctionnalités de base d'authentification, et doit par conséquent être utilisée avant toute autre API. Deuxièmement, elle simplifie l'accès des utilisateurs Orange à votre site Web en leur permettant d'utiliser les identifiants de leurs comptes Orange existants.
- L'**API Personal Calendar** fournit à votre service web un accès en temps réel aux calendriers vous permettant ainsi d'ajouter de nouvelles entrées.
- L'**API Personal Contacts** permet aux utilisateurs d'accéder à leurs carnets d'adresses.
- L'**API Personal Messages** fournit une vue simplifiée à la messagerie Orange de l'utilisateur.
- L'**API Personal Photos** permet d'interagir avec le service photo Orange France.
- L'**API Personal Profile** vous permet de proposer à votre utilisateur Orange la récupération de toute ou partie de ses informations de profil.

Ceci en toute sécurité et confiance avec la permission de l'utilisateur.

## III - API d'Authentification

### III-A - Requête

Pour tester votre code, vous devez utiliser l'un des comptes de test qui vous a été communiqué à l'inscription. Ceci vous permettra de vous authentifier de la même manière que l'utilisateur final le fera lorsque votre service sera en production.

La requête d'authentification est envoyée au fournisseur d'identité d'Orange via le navigateur, sur la base d'une redirection HTTP 302. Il s'agit d'une requête SAML ([http://fr.wikipedia.org/wiki/Security\\_assertion\\_markup\\_language](http://fr.wikipedia.org/wiki/Security_assertion_markup_language)). La requête SAML doit être **compressée au format ZIP (fonction DEFLATE, RFC 1951), puis envoyée comme paramètre d'URL encodé en Base64**.

Voici un exemple de la requête d'authentification SAML :

#### Requête HTTP

```
HTTP redirection from user's browser to Orange:
[IDP_SingleSignOnURL]?SAMLRequest=PEFldGhuUmVxdWVzdC
B4bWxucz1cInVybjpvYXNpczpuYW1lczp0YzptQUlMOjIuMDpwcm9
0b2NvbFwiICBJRD1cIl9iMDUwZTBmNGM2NzNlYzI1NzJmYzY1ZDk
1MzU5YWZlNlwiIFZ1cnNpb249XCiYlJbcIiBJc3N1ZU1uc3RhbnQ9XCI
yMDA3LTAxLTE2VDE2OjU0OjAwWlwiID48SXNzdWVyIHhtbG5zPV
widXJuOm9hc2lzOm5hbWVzOnRjO1NBTUw6Mi4wOmFzc2VydGlzbn
wiPmh0dHA6Ly9teXNwPC9Jc3N1ZXI+PC9BdXRoblJlcXVlc3Q+DQ
o%3D
```

#### Requête SAML décodée

```
<AuthnRequest
  xmlns="urn:oasis:names:tc:SAML:2.0:protocol"
  ID="_b050e0f4c673ec2572fc65d95359afe6"
  Version="2.0"
  IssueInstant="2007-01-16T16:54:00Z" >
  <Issuer xmlns="urn:oasis:names:tc:SAML:2.0:assertion">[SERVICE_ID]</Issuer>
</AuthnRequest>
```

Ci-dessous, un exemple du code PHP utilisé pour générer et envoyer la requête :

```
<?php
function randomhex($length)
{
    $key = "";

    for ( $i=0; $i < $length; $i++ )
    {
        $key .= dechex( rand(0,15) );
    }
    return $key;
}
## Metadata
require_once("idpMetadata.php");
$issuer = "[SERVICE_ID]";
$idpTargetUrl = $idpMetadata['[IDP_ID]']['SingleSignOnUrl'];

## Dynamic data of the SAML request
$id = randomhex(32);
$issueInstant = gmdate("Y-m-d\TH:i:s\Z");
## <AuthnRequest>
$authnRequest =
    "<AuthnRequest xmlns=\"urn:oasis:names:tc:SAML:2.0:protocol\" " .
    "ID=\"_" . $id . "\" " .
    "Version=\"2.0\" " .
    "IssueInstant=\"_\" . $issueInstant . "\">\n" .
    "<Issuer xmlns=\"urn:oasis:names:tc:SAML:2.0:assertion\">" .
    $issuer . "</Issuer>\n" .
    "</AuthnRequest>";

## HTTP-Redirect Binding
$encodedAuthnRequest = urlencode( base64_encode( gzdeflate( $authnRequest ) ));
$redirectUrl = $idpTargetUrl . "?SAMLRequest=" . $encodedAuthnRequest;
## Redirect
Header("Location: ".$redirectUrl); ?>
"idpMetadata.php" configuration file (used in previous code example):

<?php
# The partner SP must store the metadata to communicate with Orange identity provider.
$idpMetadata = array(
    "[IDP_ID]" =>
    array( "SingleSignOnUrl" => "[IDP_SingleSignOnURL]",
          "certificate" => "[IDP_Certificate]" ) );
?>
```

### III-B - Réponse

Pour récupérer le jeton d'utilisateur contenu dans la réponse SAML, il vous suffit simplement :


- De décoder la réponse SAML en Base64 SAML reçue dans la requête HTTP POST ;
- D'analyser la réponse SAML (document XML) pour trouver le jeton d'utilisateur. Vous le récupérerez en utilisant l'expression XPATH :

```
/samlp:Response/saml:Assertion/saml:AttributeStatement/saml:Attribute[@Name='OrangeAPIToken']/saml:AttributeValue
```

Le jeton d'utilisateur se présente comme suit :

#### Jeton utilisateur codé base 64

```
B64W7E1XG1IgvEdGOq1H9zuQQoSulCS4QOSv9/NoPtNva4psRc
+c5BFR3z0xc3DkZrelwNonn+fVG41RDBWZfdYovGxJvXZ9NTSLk
MZeQwmN08=|sau=UNAVAILABLE|ted=1200503587|oJkued8sD2
qnZiMj2HIDyLrpkhM=
```

 *Ce jeton doit être URL-encoded lorsqu'il est utilisé pour l'appel aux Personal APIs.*

La réponse d'authentification SAML décodée se présente sous la forme suivante :

#### Requête SAML décodée

```
<Response xmlns="urn:oasis:names:tc:SAML:2.0:protocol"
  ID="dBKSwwXQJpOBjBRjDjuMnWBGs6ygNXHZR"
  Version="2.0"
  IssueInstant="2008-01-16T16:54:10Z"
  Destination="[SERVICE_RETURN_URL]"
  InResponseTo="_b050e0f4c673ec2572fc65d95359afe6">
  <Status>
    <StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
  </Status>

  <Assertion xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
  ID="z2gQhxAMMUVVUOsji3HO02o95QyqupLv"
  Version="2.0"
  IssueInstant="2008-01-16T16:54:01Z">

    <Issuer>[IDP_ID]</Issuer>
    <Subject>
      <NameID Format="urn:oasis:names:tc:SAML:2.0:nameidformat:transient">
        YFOzpbIBX0zAJ61BI5jIRrze3ygWxW4V
      </NameID>
      <SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
        <SubjectConfirmationData Recipient="[SERVICE_RETURN_URL]"
          NotOnOrAfter="2008-01-16T17:09:01Z"
          InResponseTo="_b050e0f4c673ec2572fc65d95359afe6"/>
      </SubjectConfirmation>
    </Subject>
    <Conditions>
      <AudienceRestriction>
        <Audience>[SERVICE_ID]</Audience>
      </AudienceRestriction>
    </Conditions>

    <AuthnStatement AuthnInstant="2008-01-16T16:54:15Z">
      <AuthnContext>
        <AuthnContextClassRef>
          urn:oasis:names:tc:SAML:2.0:ac:classes:Password
        </AuthnContextClassRef>
      </AuthnContext>
    </AuthnStatement>
    <AttributeStatement xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <Attribute Name="OrangeAPIToken"
        NameFormat="urn:oasis:names:tc:SAML:2.0:profiles:attribute:basic">
        <AttributeValue xsi:type="xs:string">
          B64W7ElXG1IgvEdGOq1H9zuQQoSulCS4QOSv9/NoPtNva4psRc+
          c5BFR3z0xc3DkZrelwNonn+fVG41RDBWZfdYovGxJv
          XZ9NTSLkMZeQwmN08=|sau=UNAVAILABLE|ted=
          1200503587|oJ
          kued8sD2qnZiMj2HIDyLrpkhM=
        </AttributeValue>
      </Attribute>
    </AttributeStatement>
  </Assertion>
</Response>
```

Ci-dessous, un exemple de code PHP pour traiter la réponse et récupérer le token :

```
<?php
# decode the response
$authnResponse = base64_decode($_POST['SAMLResponse']);

# get data from XML
$xml = simplexml_load_string($authnResponse);
```

```

# user token
$xml->registerXPathNamespace("samlp", "urn:oasis:names:tc:SAML:2.0:protocol");
$xml->registerXPathNamespace("saml", "urn:oasis:names:tc:SAML:2.0:assertion");
$token = $xml->xpath("/samlp:Response/saml:Assertion/saml:
AttributeStatement/saml:Attribute
[@Name='OrangeAPIToken']/saml:AttributeValue");
if (count($token)>0) $token = $token[0];
# the user token can then be used to call other Personal APIs#
?>
    
```

Ceci est le développement minimal nécessaire pour récupérer le jeton.

### Pour effectuer un traitement complet et correct :

- Vérifier la signature ;
- Vérifier l'élément <status> - cela devrait être une réussite - **urn:oasis:names:tc:SAML:2.0:status:Success** ;
- Vérifier l'élément <SubjectConfirmationData> - « Recipient » doit correspondre à l'URL recevant la réponse et la date « NotOnOrAfter » ne doit pas être passée ;
- Vérifier l'élément <Audience> - il **doit** correspondre à l'identifiant technique du fournisseur de service tiers ;
- Analyser la réponse SAML (document XML) pour trouver l'élément <NameID> - c'est le nom d'identifiant ;

## IV - Ajouter un évènement au Calendrier

L'API Calendrier permet d'ajouter des évènements sur au calendrier de l'utilisateur.

### IV-A - Requête

Les requêtes aux Personal APIs se font en HTTP GET, avec la forme générique suivante :

#### Format :

```
[PersonalAPIURL]?action=[action name]&token=[user token]&param=[value]...
```

#### Dans le cas du calendrier :

```
[PersonalCalendarV1URL]?action=addevent&token=Hjlkzjlfkzef23423kjlkj&param=value...
```

Plus précisément, pour le calendrier, les paramètres sont les suivants :

#### Format de requête

```
[PersonalCalendarV1EndPoint]?action=addevent&title=[title]&
location=[location]&description=[description]&startdate=[start date]&
starttime=[start time]&enddate=[end date]&endtime=[end time]&
datepattern=dd/MM/yyyy&timepattern=HH:mm&token=[user token]
```

Qui donne par exemple :

#### Requête

```
[PersonalCalendarV1EndPoint]?action=addevent&title=test&location=some%20location&
description=some%20description&startdate=10/01/2000&starttime=1000&enddate=10/01/2000&
endtime=1100&datepattern=dd/MM/yyyy&timepattern=HHmm&token=Hjlkzjlfkzef23423kjlkj
```

Les paramètres d'entrée sont les suivants :

Nom	Description	Obligatoire	Type
title	the title of the event	Oui	String
startdate	start date of event in dd/MM/yyyy format	Oui	Date
enddate	end date of event	Oui	Date
starttime	start time of event in HH:mm format	Oui	Time
endtime	end time of event	Oui	Time
description	description of the event	Oui	String
location	location of the event	Oui	String
token	user token that is retrieved using the Authentication API	Oui	String

 Les paramètres **datepattern** et **timepatern** ne doivent pas être modifiés

 N'oubliez pas d'URL-encoder vos String.

## IV-B - Gestion de l'authentification et de la Privacy

Lors de l'appel aux APIs, les messages d'erreur suivants peuvent être retournés :

- Message d'erreur d'authentification. La réponse pour le code d'erreur « -1 » a le format suivant :

### XML / Erreur -1

```
<?xml version="1.0" encoding="UTF-8"?>
<error>
  <code>-1</code>
  <detail>InvalidTokenException</detail>
</error>
```

Ce message indique que votre jeton n'est plus valide, et il faut donc réauthentifier l'utilisateur. Pour cela, mettez en oeuvre le mécanisme d'authentification tel que vu ci-dessus.

- Message d'erreur La réponse pour le code d'erreur « -3 » est au format suivant :

### XML / Erreur -3

```
<?xml version="1.0" encoding="UTF-8"?>
<error>
  <code>-3</code>
  <detail>PrivacyAccessDeniedException</detail>
  <url>http://[PrivacyHost]/privacy/interaction.do?
family=agenda&serviceId=MYSERVICE&attributes=,add_event</url>
</error>
```

Ce message indique que l'utilisateur n'a pas encore donné son consentement pour l'accès au service.

Dans ce dernier cas, le paramètre <url> contient l'URL de la page « respect de la vie privée » qu'il est nécessaire d'utiliser pour inviter l'utilisateur à donner son consentement pour l'accession aux données requises par la Personal API. L'URL de l'exemple est donnée à titre illustratif seulement.

Pour inviter l'utilisateur à donner son consentement, votre service doit le rediriger vers le paramètre concaténé avec votre URL de retour.

Par exemple, si votre URL de retour est « `http://myservice.com/displaypage.php` », et votre `serviceId` "MYSERVICE" il vous faudra rediriger l'utilisateur vers l'URL suivante :

#### URL de redirection

```
http://[PrivacyHost]/privacy/interaction.do?family=agenda&serviceId=MYSERVICE
&attributes=,add_event&urlRetour= http%3A%2F%2Fmyservice.com%2Fdisplaypage.php
```

Après cette interaction, l'utilisateur est redirigé vers votre URL de retour où vous pourrez appeler à nouveau la Personal API. **Ceci signifie que dans le cas d'une page accédée en POST, vous devez stocker les informations dans votre cookie par exemple, ou sous forme de session sur votre serveur.**

Le diagramme séquence ci-joint vous résume les interactions à mettre en oeuvre pour gérer la Privacy : **Diagramme séquence de la gestion d'une erreur de Privacy lors de l'appel aux APIs.**

## IV-C - Réponse

Si l'événement est correctement ajouté, la réponse contiendra l'ID de l'événement ainsi qu'un résultat = 0 et se présentera comme suit :

#### XML / exemple de réponse

```
<?xml version="1.0" encoding="UTF-8" ?>
  <xpage version="1.0">
    <command-list>
      <command action="cauupdate" request="s01">
        <event-data>
          <eventid>4790</eventid>
          <id>10012000</id>
        </event-data>
        <result>0</result>
      </command>
    </command-list>
    <parameter-list />
  </xpage>
```

Lorsqu'une erreur se produit, la réponse contient un code d'erreur (paramètre de « type »), un sous-type interne et le message :

#### XML / format générique d'erreur

```
<command request="" action="action">
  <result>yyy</result>
  <error subtype="xxx" type="yy">message</error>
</command>
```

### Les erreurs les plus significatives :

- 1 'mandatory parameter' is missing ;
- 2 'parameter' is invalid ;
- 4 The command 'action' is unknown ;
- 5 'end-userid' is incorrect (database is corrupted) ;
- 100 Service Calendar not provisioned for this end-user.

## V - Conclusion

### VI-A - À vous de jouer

L'objectif de cet article est de vous montrer comment utiliser les Personal APIs Orange sous forme d'un exemple d'usage d'une API, l'API calendrier.

À vous maintenant d'imaginer tous les usages basés sur l'authentification, le profil, les contacts, le calendrier, les photos, et bien d'autres services dans le futur.

C'est pour vous l'opportunité de rajouter des scénarios uniques à vos services en lignes, de vous différencier, d'augmenter l'usage et la valeurs de vos applications.

### VI-B - Pour aller plus loin

#### **Pour les utilisateurs en langue française**

Home page des Personal Apis Orange :

[http://www.orangepartner.com/site/frfr/access\\_orange\\_apis/personal\\_apis/p\\_personal\\_apis.jsp](http://www.orangepartner.com/site/frfr/access_orange_apis/personal_apis/p_personal_apis.jsp)

Documentation de l'API d'authentification :

[http://www.orangepartner.com/site/frfr/access\\_orange\\_apis/authentication\\_api/p\\_personal\\_authentication\\_api.jsp](http://www.orangepartner.com/site/frfr/access_orange_apis/authentication_api/p_personal_authentication_api.jsp)

Documentation de l'API Calendar :

[http://www.orangepartner.com/site/frfr/access\\_orange\\_apis/personal\\_calendar\\_api/p\\_personal\\_calendar\\_api.jsp](http://www.orangepartner.com/site/frfr/access_orange_apis/personal_calendar_api/p_personal_calendar_api.jsp)

S'inscrire aux Personal APIs :

[administrator\\_web\\_interface](#)

Pour les utilisateurs en langue anglaise, utilisez l'onglet "API" du site Orange partner.

1 : L'auteur de ce tutorial travaille pour Orange et a contribué aux développements des API.