GUIDE DE DÉMARRAGE DE L'HORODATAGE UNIVERSIGN

Universign

VERSION : **1**	DIFFUSION : ***PUBLIQUE***	26/08/2016

Table des Matières

1. Avant-Propos

Pour accéder de manière automatisée à ses services, Universign met à disposition des développeurs des APIs (Application Programming Interface) documentées. Ces interfaces, basées sur des protocoles Web standards, vous permettent d'intégrer très facilement l'horodatage à vos applications.

2. Horodatage standard

L'horodatage d'un fichier produit une preuve nommée sceau d'horodatage. Cette opération est applicable à tout type de document, sans limitation de format ou d'extension. De plus, seule une empreinte du document est transmise au service d'horodatage.

• Configuration

Universign offre deux options d'accès à son service d'horodatage :

- le protocole d'horodatage défini dans le standard RFC 3161. Pour l'utiliser, il vous suffit de configurer votre logiciel client.
- une API simple basée sur un envoi de formulaire HTTP avec une requête POST facilement intégrable dans n'importe quel environnement de production.

• Paramètres d'authentification

Sécurité	SSL
Mode d'authentification	HTTP basic authentification
Identifiants	email et mot de passe

• Accès suivant le procotole d'horodatage (RFC3161)

URL: https://ws.universign.eu/tsa/

• Accès avec l'envoi d'un formulaire HTTP par requête POST

URL: https://ws.universign.eu/tsa/post/

Paramètres du formulaire encodés selon le mime-type 'application/x-www-form-urlencoded' :

hashAlgo	Nom de l'algorithme de calcul d'empreinte utilisé : SHA256, SHA384 ou SHA512
hashValue	HTTP basic authentification
withCert	booléen (true ou false) indiquant si le sceau d'horodatage contient le certificat de l'unité d'horodatage

Le serveur renvoie le sceau d'horodatage encodé en ASN.1 dans le format *TimeStampToken* tel que décrit dans la RFC 3161.

- Quelques exemples de code

En Python

```
#!/usr/bin/python

import urllib;
import urllib2;
import hashlib;
import base64;

# first we construct the parameters for the request
data = {};
data['hashAlgo'] = "SHA256";
data['withCert'] = "true";
data['hashValue'] = hashlib.sha256(dataToTimestamp).hexdigest();
params = urllib.urlencode(data);

# basic HTTP authentication is needed to access this service
headers = {};
auth = base64.encodestring(username + ":" + password);
headers["Authorization"] = "Basic " + auth;

# then the request itself
request = urllib2.Request("https://ws.universign.eu/tsa/post/", params, headers);

# all is ready, the request is made
response = urllib2.urlopen(request);
tsp = response.read();
```

En PHP

```php
$hashedDataToTimestamp = hash('sha256', $dataToTimestamp);
$dataToSend = array ('hashAlgo' => 'SHA256', 'withCert' => 'true', 'hashValue' => $hashedDataToTimestamp);
$dataQuery = http_build_query($dataToSend);
$context_options = array (
     'http' => array (
       'method' => 'POST',
       'header'=> "Content-type: application/x-www-form-urlencoded\r\n"
        ."Content-Length: " . strlen($dataQuery) . "\r\n"
        ."Authorization: Basic ".base64_encode($login.':'.$password)."\r\n",
       'content' => $dataQuery
      )
   );

$context = stream_context_create($context_options);
$fp = fopen("https://ws.universign.eu/tsa/post/", 'r', false, $context);
$tsp = stream_get_contents($fp);
```

En Java

```java
static InputStream doTsp(String login, String pwd, String hash, String algo)
 throws Exception
{
 URLConnection conn = new URL("https://ws.universign.eu/tsa/post/").openConnection();
 conn.setDoOutput(true);
 conn.setDoInput(true);
 String authString = login + ":" + pwd;
 String authStringEnc = Base64.encode(authString);
 conn.setRequestProperty("Authorization", "Basic " + authStringEnc);

 OutputStream out = conn.getOutputStream();
 String params = "hashAlgo=" + URLEncoder.encode(algo, "UTF-8") + "&hashValue="
   + URLEncoder.encode(hash, "UTF-8") + "&withCert=" + URLEncoder.encode("false", "UTF-8");
 out.write(params.getBytes("UTF-8"));
 out.flush();

 return conn.getInputStream();
}
```

Pour toutes questions complémentaires, merci de nous contacter au 01 44 08 73 00 ou sales@universign.com

3. Horodatage de PDF

Horodater un fichier PDF diffère de l'horodatage de fichiers courants car le sceau d'horodatage est embarqué dans le fichier PDF. Lors de l'opération d'horodatage, le document PDF source doit être transmis intégralement afin d'y apposer le jeton d'horodatage.

• Configuration

Universign offre deux options d'accès à son service d'horodatage de PDF :

- le protocole de communication XML-RPC qui permet la transmission de paramètres depuis n'importe quelle plateforme.
- une API simple basée sur un envoi de formulaire HTTP avec une requête POST facilement intégrable dans n'importe quel environnement de production.

• Paramètres d'authentification

Sécurité	SSL
Mode d'authentification	HTTP basic authentification
Identifiants	email et mot de passe

Accès suivant le procotole XML-RPC

URL	**https://ws.universign.eu/sign/rpc**
Service	timestamper
Méthode	*timestamp ()*
Paramètres	Le document PDF à horodater
Retour	Le document est horodaté

Accès avec l'envoi d'un formulaire HTTP par requête POST

URL: https://ws.universign.eu/tsa/pdf/post/

Paramètres du formulaire encodés selon le mime-type 'multipart/form-data' :

file	le document à horodater

- Quelques exemples de code

En Python

```
import xmlrpclib;

if __name__ == "__main__":
pdfDocument = xmlrpclib.Binary(open("myDoc.pdf").read())

proxy = xmlrpclib.ServerProxy("https://me@myCompany.com:myPwd@ws.universign.eu/tsa/pdf/rpc")
myTimestampedPdfDocument = proxy.timestamper.timestamp(pdfDocument)
f = open("myTimestampedDoc.pdf", "w+")
f.write(myTimestampedPdfDocument.data)
f.close()
```

En PHP

```
$destination = "https://".$login.':'.$password."@ws.universign.eu/tsa/pdf/post/";

$eol = "\r\n";
$data = '';
$mime_boundary = md5(time());
$data .= '--'.$mime_boundary.$eol;
$data .= 'Content-Disposition: form-data; name="file"; filename="'.$fileName.'"'.$eol;
$data .= 'Content-Type: text/plain'.$eol;
$data .= 'Content-Transfert-Encoding: base64'.$eol.$eol;
$data .= $dataToTimestamp.$eol;
$data .= "--".$mime_boundary."--".$eol.$eol;
$params = array('http' => array(
    'method' => 'POST',
    'header' => 'Content-Type: multipart/form-data; boundary='.$mime_boundary.$eol,
    'content' => $data
));

$context = stream_context_create($params);

$tsp = @file_get_contents($destination, FILE_TEXT, $context);
```

4. Support & Aide Universign

Le service de signature Universign, disponible 24h/24 et 7j/7, est limité à une signature par seconde.

Si une erreur survient lors du processus d'horodatage, votre compte Universign ne sera pas débité.

Lorsqu'une requête échoue, Universign indique un code d'erreur qui vous permet de comprendre et de résoudre le problème rencontré.

- 400 - requête malformée ou option non supportée.
- 401 - problème d'authentification : utilisateur inconnu ou mot de passe incorrect.
- 403 - problème sur le compte : plus de sceau ou pack expiré.
- 404 - ressource non trouvée, ceci est probablement dû à une erreur d'URL.
- 500 - erreur interne au service Universign. Veuillez réessayer plus tard ou contacter le service support.

5. Contactez-nous !

Vous souhaitez intégrer pleinement la solution Universign à votre environnement IT, merci de prendre contact en cliquant-ici