

# Fast Parapheur

# Documentation sur les Web

# Services REST

Version : **En relecture**

Date de la version : **05 Mai 2021**

Auteur : **Equipe de Développement FAST**

Société : **DOCAPOSTE FAST**

Confidentialité : **Diffusion publique**

# SOMMAIRE

1.	Accrochage avec FAST-Parapheur	3
1.1.	Dépôt de fichier	3
1.2.	Traitement du fichier	3
2.	Détails des Méthodes des web services	3
2.1.	download	3
2.2.	upload	4
2.3.	ondemand	4
2.4.	uploadMeta	7
2.5.	uploadOTPInformation	8
2.6.	history	9
2.7.	delete	10
2.8.	Download Acknowledgement	11
2.9.	Download PES Retour	11
2.10.	Download Chorus	11
2.11.	list remaining acknowledgement	12
2.12.	list remaining chorus	12
2.13.	list remaining pes retour	13
2.14.	GetRefusalMessage	13
2.15.	GetSMSURL	13
2.16.	uploadOTP	14
2.17.	exportUsersData	15
2.18.	synchronizationUsers	16

---

## 1. Accrochage avec FAST-Parapheur

### 1.1. Dépôt de fichier

Lors du dépôt du fichier, il faut choisir le « circuit » dans lequel lancer le document (et de manière optionnelle un « libellé » qui vous permet de classer les documents dans la cinématique). Chaque circuit possède son propre paramétrage (signature manuelle, signature serveur, Boucle SMS, circuit de validation, etc. etc.) et ses propres utilisateurs.

Lors d'un dépôt en Web service, le parapheur renvoie en retour

L'identifiant unique du fichier (NodeRefId) si le dépôt s'est déroulé avec succès

Une erreur (refus de traiter ou erreur technique) dans le cas contraire

Note sur les erreurs : Dans le cas d'une erreur, il se peut que le fichier ait été refusé car il ne respecte pas les règles de traitement définies sur le parapheur. Ces règles sont les suivantes :

**Doublon** : le triplet « circuit » / « libellé » / « nom de fichier » doit être unique. Cette règle évite de déposer des doublons.

**Format refusé** : un « circuit » est paramétré pour accepter un ou plusieurs types de fichier (PDF, DOC, XML, etc.). Dans le cas où le type de fichier déposé ne correspond pas, celui-ci sera refusé.

### 1.2. Traitement du fichier

Les traitements associés au fichier (validation, signature, vérification de signature, archivage, classement) peuvent être manuels ou automatiques. Ceci est paramétré au sein du circuit choisi.

## 2. Détails des Méthodes des web services

### 2.1. download

La méthode download permet de récupérer le document et, optionnellement, certain de ces documents associés.

#### (GET)

<https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<id>/download>

#### Paramètres

**id** (obligatoire) - 10 caractères.

Identifiant du document. Retourné lors du dépôt du document.

**acquit** - Booléen

Ne concerne que les fichiers PES. Si le PES a reçu un acquittement de la DGFIP, alors cet acquittement sera récupéré, en plus du PES.

**FdC** - Booléen

FdC signifie "Fiche de Circulation". Permet d'ajouter au téléchargement la fiche de circulation associée au document.

#### Réponse

En cas d'opération réussie, Le fichier est retourné si aucun des paramètres optionnels n'est renseigné à TRUE. Sinon, c'est un contenu de type ZIP qui est retourné.

## 2.2. *upload*

La méthode upload permet de déposer un document sur le Parapheur électronique.

### (POST)

[https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<siren>/<circuit\\_id>/upload](https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<siren>/<circuit_id>/upload)

### Paramètres

**siren** (obligatoire) - Entier, 9 caractères.

Le siren du client. Il s'agit de l'identifiant unique permettant d'identifier le client chez FAST.

**circuit\_id** (obligatoire) - 255 caractères maximum

Le circuit dans lequel le document doit être déposé.

**doc** (obligatoire) - Fichier (multipart/form-data)

Le document à déposer sur le portail.

**label** - 255 caractères maximum

Le dossier dans lequel le document doit être déposé. Le dossier permet à l'utilisateur de trier et de filtrer ses documents plus efficacement sur le portail.

**comment** - 255 caractères maximum

Un commentaire à associer au document. Le commentaire sera visible de tous les utilisateurs du circuit.

**emailDestinataire** - 255 caractères maximum

Un ou des emails séparés par une virgule ou un point-virgule. Une fois le document signé, un email sera envoyé à ces adresses avec un lien permettant de télécharger le document signé.

### Note

Le triplet circuit\_id / label / nom\_du\_fichier doit être unique. Si vous désirez, dans un même circuit, déposer deux documents avec le même nom, alors il faudra leur donner un nom de dossier différent.

### Réponse

En cas d'opération réussie, l'identifiant unique du document est retourné. Cet identifiant sera demandé pour toute autre opération concernant ce document.

## 2.3. *ondemand*

La méthode "ondemand" est le service qui permet de déposer un document dans un circuit à la volée. Il faut fournir au service le document à déposer, ses potentielles annexes, ainsi que la configuration du circuit, contenant les différentes étapes et leurs acteurs.

### (POST)

<https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/ondemand/<siren>/upload>

---

---

## Paramètres

**siren** (obligatoire) - Entier, 9 carac.

Le siren du client. Il s'agit de l'identifiant unique permettant d'identifier le client chez FAST.

**circuit\_id** (obligatoire) - 255 carac. max

Le circuit dans lequel le document doit être déposé.

**doc** (obligatoire) - Fichier (multipart/form-data)

Le document à déposer sur le portail.

**circuit** (obligatoire) - JSON

La liste des étapes du document, ainsi que ses acteurs. Plus de détails plus bas.

**annexes** - Fichiers (multipart/form-data)

Les annexes associées au document. Les annexes seront consultables par tous les acteurs renseignés dans la configuration du circuit.

**agents** – 255 carac max

Les emails des utilisateurs, séparé par une virgule, dont on souhaite donner le rôle AGENT sur le document. Les agents ne peuvent signer, ni valider les documents, mais ils peuvent consulter le document, le faire avancer dans les diverses étapes qui ne concerne ni la signature, ni la validation et ajouter des commentaires et annexes.

**comment** - 255 carac. max

Un commentaire à associer au document. Le commentaire sera visible de tous les utilisateurs du circuit.

**email\_destinataire** - 255 carac. max

Un ou des emails séparés par une virgule ou un point-virgule. Une fois le document signé, un email sera envoyé à ces adresse avec un lien permettant de télécharger le document signé.

**email\_CC** – 255 carac. Max

Idem que **email\_destinataire** mais en copie carbone.

## Exemple de circuit

```
{
  "type" : "BUREAUTIQUE_PDF",
  "steps": [
    {
      "step" : "visa",
      "members" : ["AAA@domain.fr", "BBB@domain.fr"]
    },
    {
      "step" : "visa",
      "members": ["CCC@domain.fr"]
    }
  ]
}
```

```
    },  
    {  
      "step" : "signature",  
      "members": ["DDD@domain.fr"]  
    },  
    {  
      "step" : "signature",  
      "members": ["AAA@domain.fr"]  
    }  
  ]}
```

Les types autorisés sont :

BUREAUTIQUE\_PDF : Signature PADES

BUREAUTIQUE\_XML : Signature XADES détaché

HELIOS\_AVEC\_TELETRANSMISSION : Signature PES avec télétransmission vers la DGFIP

HELIOS\_SANS\_TELETRANSMISSION : Signature PES sans télétransmission

Les step autorisés sont :

Signature

Visa

D'autres étapes se rajouteront dans le futur

Les members sont les utilisateurs associés à l'étape. Attention, l'utilisateur qui dépose le document ne peut pas être signataire ou valideur sur le circuit.

## Réponse

En cas d'opération réussie, l'identifiant unique du document est retourné. Cet identifiant sera demandé pour toute autre opération concernant ce document.

## Erreurs

Les erreurs sont de la forme :

```
{  
  "generation": 1561987005624,  
  "developerMessage": "Invalid circuit type",  
  "userFriendlyMessage": "Le type de circuit est incorrect",  
  "errorCode": 117  
}
```

## Codes d'erreurs

Les codes d'erreurs les plus fréquemment retournés par cette API sont listés dans le tableau ci-dessous. Dans certains cas, d'autres codes peuvent être retournés.

Code d'erreur	Message d'erreur	Signification
100	Unauthorized	Votre utilisateur n'est pas autorisé à déposer dans ce siren.
111	Subscriber not found	Le siren renseigné n'existe pas
117	Invalid circuit type	La valeur <i>type</i> du paramètre <i>circuit</i> est incorrect
118	Invalid step type	La valeur <i>step</i> du paramètre <i>circuit</i> est incorrect
119	Invalid step member	L'un des <i>member</i> du paramètre <i>circuit</i> est incorrect ou n'existe pas
120	Upload doc error	Une erreur est survenue lors du dépôt du document.
121	Circuit definition error	Le paramètre <i>circuit</i> n'est pas correctement renseigné.
122	The agent cannot be a step member	L'agent déposant ne peut être signataire ou valideur du document.

## 2.4. *uploadMeta*

La méthode "uploadMeta" est le service qui permet de renseigner les métadonnées d'un document. Le fichier peut avoir des métadonnées uniquement si une des étapes suivantes est définie dans son circuit :

- Définition des métadonnées
- OTP
- Digiposte

### (PUT)

[https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<Document\\_ID>/metadata/define](https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<Document_ID>/metadata/define)

### Paramètres

**Document\_ID** (obligatoire) - Entier, 9 carac.

Il s'agit de l'identifiant unique d'un document. C'est l'identifiant retourné par les méthodes de dépôt de document.

**metadata** (obligatoire) - Fichier (multipart/form-data)

Un fichier XML contenant les métadonnées.

### Exemple de fichier metadata

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<meta-data-list>
  <meta-data name="TEXT_001" value="John"/>
  <meta-data name="TEXT_002" value="Doe"/>
  <meta-data name="TEXT_003" value="0606060606"/>
  <meta-data name="NUM_001" value="45812"/>
</meta-data-list>
```

## 2.5. *uploadOTPInformation*

La méthode "uploadOTPInformation" permet de renseigner les métadonnées OTP d'un document.

**(PUT)**

[https://localhost/parapheur-ws/rest/v1/documents/v2/otp/<Document\\_ID>/metadata/define](https://localhost/parapheur-ws/rest/v1/documents/v2/otp/<Document_ID>/metadata/define)

### Paramètres

**Document\_ID** (obligatoire) - Entier, 9 caractères.

Il s'agit de l'identifiant unique d'un document. C'est l'identifiant retourné par les méthodes de dépôt de document.

**otpinformation** (obligatoire) - Fichier (multipart/form-data)

Un fichier XML contenant les métadonnées.

### Exemple de fichier otpinformation

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<meta-data-list>
```

```
  <meta-data name="OTP_firstname_0" value="John"/>
```

```
  <meta-data name="OTP_lastname_0" value="Doe"/>
```

```
  <meta-data name="OTP_phonenumber_0" value="0606060606"/>
```

```
  <meta-data name="OTP_email_0" value="john.doe@mail.fr"/>
```

```
  <meta-data name="OTP_firstname_1" value="John"/>
```

```
  <meta-data name="OTP_lastname_1" value="Doe"/>
```

```
  <meta-data name="OTP_phonenumber_1" value="0606060606"/>
```

```
  <meta-data name="OTP_email_1" value="john.doe@mail.fr"/>
```

```
  (...)
```

```
  <meta-data name="OTP_firstname_n-1" value="John"/>
```

```
  <meta-data name="OTP_lastname_n-1" value="Doe"/>
```

```
  <meta-data name="OTP_phonenumber_n-1" value="0606060606"/>
```

```
  <meta-data name="OTP_email_n-1" value="john.doe@mail.fr"/>
```

```
</meta-data-list>
```

Le fichier doit contenir les informations OTP de chaque étape. Les noms des informations OTP sont suffixés par l'index de l'étape OTP à laquelle ils sont liés (0 pour la 1<sup>ère</sup>, 1 pour la 2<sup>ème</sup>, 2 pour la 3<sup>ème</sup>, ..., n-1 pour la n<sup>ème</sup>)

---

## 2.6. *history*

La méthode "history" permet de récupérer les différentes étapes par lesquels le document est passé.

### (GET)

[https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<Document\\_ID>/history](https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<Document_ID>/history)

### Paramètres

**Document\_ID** (obligatoire) - Entier, 9 carac.

Il s'agit de l'identifiant unique d'un document. C'est l'identifiant retourné par les méthodes de dépôt de document.

### Exemple de réponse

```
[
  {
    "userFullname": "Marc Antiome",
    "date": "2019-12-11T09:41:10.000Z",
    "stateName": "Préparé"
  },
  {
    "userFullname": "boucle-sms",
    "date": "2019-12-11T09:41:31.000Z",
    "stateName": "Informations OTP définies"
  },
  {
    "userFullname": "boucle sms",
    "date": "2019-12-11T09:59:22.000Z",
    "stateName": "OTP validé"
  },
  {
    "userFullname": "boucle sms",
    "date": "2019-12-11T09:59:22.000Z",
    "stateName": null
  },
  {
    "userFullname": "parapheur parapheur",
    "date": "2019-12-11T09:59:22.000Z",
    "stateName": "Classé"
  }
]
```

Chaque étape sera détaillée, avec l'auteur de l'action, la date et l'état dans lequel le document est passé. Si le userFullname est "boucle sms" ou "parapheur parapheur", c'est que l'opération a été faite automatiquement, sans opération manuel d'un utilisateur.

---

Voici les explications de certains états renvoyés par cette méthode :

- Préparé : Le document est déposé sur FAST Parapheur ;
- Métadonnées définies : Les métadonnées ont été ajoutés ;
- Envoyé pour signature : Le document a été envoyé pour signature ;
- Signé : Le signataire a signé le document ;
- Refusé : Le signataire a refusé le document ;
- Signature validée : (Ne concerne que la signature PDF et détachée) La signature du document a été validée par notre valideur de signature ;
- signature rejetée : le vérificateur de signature a rejeté la signature ;
- Envoyé pour visa : Le document a été envoyé pour validation d'un valideur ;
- Validé : Le valideur a validé le document ;
- Rejeté : Le valideur a rejeté le document (le rejet est l'inverse de la validation) ;
- En cours d'envoi : (Ne concerne que les circuits PES) envoi en cours vers la plateforme FAST ;
- Envoyé à FAST : (Ne concerne que les circuits PES) Le document est envoyé sur la Plateforme FAST, afin d'être acheminé vers HELIOS
- Transmis à Hélios : (Ne concerne que les circuits PES) Le document est envoyé chez HELIOS
- Acquittance Hélios : (Ne concerne que les circuits PES) Le document a reçu l'acquittance d'HELIOS
- Échec de l'envoi à FAST : une erreur est survenue lors de l'envoi à la plateforme FAST ;
- Échec du traitement FAST : la plateforme FAST a renvoyé une erreur ;
- Échec de l'envoi à Hélios : une erreur est survenue pendant l'envoi à Hélios du document ;
- Informations OTP définies : (Ne concerne que les circuits OTP) : Le document a été déposé et les informations concernant les étapes d'OTP ont été définies.
- OTP validé (Ne concerne que les circuits OTP) : Le client a reçu son code SMS, l'a renseigné sur la page OTP et a cliqué sur le bouton « signer ».
- Classé : Le document est en fin de vie, il n'attend aucune action et n'apparaît plus sur le tableau de bord
- Archivé : Le document a été archivé dans le coffre-fort électronique.
- Document remplacé : le document a été remplacé par un autre ;

## 2.7. delete

La méthode "delete" permet de supprimer un document. Le document doit être dans le bon état pour pouvoir être supprimé. Attention, cette suppression est définitive. La liste des états permettant de supprimer un document sont :

- Préparé
- Refusé
- Rejeté

### (DELETE)

[https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<Document\\_ID>](https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<Document_ID>)

---

## Paramètres

**Document\_ID** (obligatoire) - Entier, 9 carac.

Il s'agit de l'identifiant unique d'un document. C'est l'identifiant retourné par les méthodes de dépôt de document.

## 2.8. *Download Acknowledgement*

La méthode "download acknowledgement" permet de récupérer l'acquittement HELIOS d'un document PES.

**(GET)**

[https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<Document\\_ID>/ack/download](https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<Document_ID>/ack/download)

## Paramètres

**Document\_ID** (obligatoire) - Entier, 9 carac.

Il s'agit de l'identifiant unique d'un document. C'est l'identifiant qui a été récupéré depuis le WS "list remaining acknowledgement".

## Réponse

En cas d'opération réussie, la réponse sera l'acquittement Hélios sous format XML.

## 2.9. *Download PES Retour*

La méthode "download PES Retour" permet de récupérer un PES Retour.

**(GET)**

[https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<Document\\_ID>/pes-retour/download](https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<Document_ID>/pes-retour/download)

## Paramètres

**Document\_ID** (obligatoire) - Entier, 9 carac.

Il s'agit de l'identifiant unique d'un document. C'est l'identifiant qui a été récupéré depuis le WS "list remaining PES Retour"

## Réponse

En cas d'opération réussie, la réponse sera le PES Retour sous format XML.

## 2.10. *Download Chorus*

La méthode "download Chorus" permet de récupérer un fichier Chorus.

**(GET)**

[https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<Document\\_ID>/chorus/download](https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<Document_ID>/chorus/download)

---

## Paramètres

**Document\_ID** (obligatoire) - Entier, 9 caractères.

Il s'agit de l'identifiant unique d'un document. C'est l'identifiant qui a été récupéré depuis le WS "list remaining Chorus"

## Réponse

En cas d'opération réussie, la réponse sera le fichier Chorus sous format XML.

### 2.11. *list remaining acknowledgement*

La méthode "list remaining acknowledgement" permet de récupérer la liste des documents dont l'acquittement HELIOS n'a pas encore été récupéré via WS.

#### (GET)

<https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<siren>/ack/remaining>

## Paramètres

**siren** (obligatoire) - Entier, 9 caractères.

Le siren du client. Il s'agit de l'identifiant unique permettant d'identifier le client chez FAST.

## Réponse

En cas d'opération réussie, la réponse renverra une liste de document\_ID à utiliser avec le service "download acknowledgement".

Exemple de réponse :

[320073963, 320073961]

### 2.12. *list remaining chorus*

La méthode "list remaining chorus" permet de récupérer la liste des documents CHORUS qui n'ont pas encore été récupérés via WS.

#### (GET)

<https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<siren>/chorus/remaining>

## Paramètres

**siren** (obligatoire) - Entier, 9 caractères.

Le siren du client. Il s'agit de l'identifiant unique permettant d'identifier le client chez FAST.

## Réponse

En cas d'opération réussie, la réponse renverra une liste de document\_ID à utiliser avec le service "download chorus".

Exemple de réponse :

[320073963, 320073961]

---

## 2.13. *list remaining pes retour*

La méthode "list remaining pes retour" permet de récupérer la liste des PES RETOUR qui n'ont pas encore été récupérés via WS.

### (GET)

<https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<siren>/ack/remaining>

### Paramètres

**siren** (obligatoire) - Entier, 9 caractères.

Le siren du client. Il s'agit de l'identifiant unique permettant d'identifier le client chez FAST.

### Réponse

En cas d'opération réussie, la réponse renverra une liste de document\_ID à utiliser avec le service "download pes retour".

Exemple de réponse :  
[320073963, 320073961]

## 2.14. *GetRefusalMessage*

La méthode "get refusal message" permet de récupérer le message de refus renseigné par un signataire lors d'un refus de signature. Cette information est aussi disponible dans la fiche de circulation disponible via la méthode "download" ou "<FICHEDECIRCU>".

### (GET)

[https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<Document\\_ID>/comments/refusal](https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/<Document_ID>/comments/refusal)

### Paramètres

**Document\_ID** (obligatoire) - Entier, 9 caractères.

Il s'agit de l'identifiant unique d'un document. C'est l'identifiant qui a été récupéré depuis le WS "download"

## 2.15. *GetSMSURL*

La méthode "getSMSURL" est utilisée lors de l'utilisation de signature OTP. Normalement, les fichiers OTP sont signés unitairement, mais il est possible de faire signer plusieurs fichiers OTP via une seule URL et un seul code SMS grâce à cette méthode. À condition que tous les fichiers concernés possèdent exactement les mêmes métadonnées (email et/ou numéro de téléphone).

### (GET)

[https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/otp/url?id=<Document\\_ID\\_1>&id=<Document\\_ID\\_2>](https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/otp/url?id=<Document_ID_1>&id=<Document_ID_2>)

---

## Paramètres

**Document\_IDs** (obligatoire) - Entier, 9 caractères.

Il s'agit des identifiants uniques des documents. C'est l'identifiant qui a été récupéré depuis le WS "download"

## Réponse

En cas d'opération réussie, la réponse renverra un URL permettant à votre signataire d'accéder à une signature OTP avec tous les documents renseignés

Exemple de réponse :

```
{
  "url": "https://demo-confiance.dfast.fr/boucle-
  sms/documentView?id=VlwFd9b6SJIRtFJnFDh0bpYr8dBSFGEwoEJNCAUttcw="
}
```

## 2.16. uploadOTP

La méthode "uploadOTP" permet de déposer un document dans un circuit de signature OTP.

### (POST)

[https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/otp/<siren>/<circuit\\_ID>/upload](https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/otp/<siren>/<circuit_ID>/upload)

## Paramètres

**siren** (obligatoire) - Entier, 9 caractères.

Le siren du client. Il s'agit de l'identifiant unique permettant d'identifier le client chez FAST.

**circuit\_id** (obligatoire) - 255 caractères. max

Le circuit dans lequel le document doit être déposé.

**doc** (obligatoire) - Fichier (multipart/form-data)

Le document à déposer sur le portail.

**meta** (obligatoire) - Fichier (multipart/form-data)

Le fichier contenant les métadonnées liées à la signature OTP

**label** - 255 caractères. max

Le dossier dans lequel le document doit être déposé. Le dossier permet à l'utilisateur de trier et de filtrer ses documents plus efficacement sur le portail.

**comment** - 255 caractères. max

Un commentaire à associer au document. Le commentaire sera visible de tous les utilisateurs du circuit.

**referer** - 255 caractères. max

L'URL vers laquelle votre signataire sera redirigé après signature OTP.

---

## Exemple de fichier meta

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<meta-data-list>
  <meta-data name="OTP_firstname_0" value="John"/>
  <meta-data name="OTP_lastname_0" value="Doe"/>
  <meta-data name="OTP_phonenumber_0" value="0606060606"/>
  <meta-data name="OTP_email_0" value="john.doe@mail.fr"/>

  <meta-data name="OTP_firstname_1" value="John"/>
  <meta-data name="OTP_lastname_1" value="Doe"/>
  <meta-data name="OTP_phonenumber_1" value="0606060606"/>
  <meta-data name="OTP_email_1" value="john.doe@mail.fr"/>
  (...)
  <meta-data name="OTP_firstname_n-1" value="John"/>
  <meta-data name="OTP_lastname_n-1" value="Doe"/>
  <meta-data name="OTP_phonenumber_n-1" value="0606060606"/>
  <meta-data name="OTP_email_n-1" value="john.doe@mail.fr"/>
</meta-data-list>
```

Le fichier doit contenir les informations OTP de chaque étape. Les noms des informations OTP sont suffixés par l'index de l'étape OTP à laquelle ils sont liés (0 pour la 1<sup>ère</sup>, 1 pour la 2<sup>ème</sup>, 2 pour la 3<sup>ème</sup>, ..., n-1 pour la n<sup>ème</sup>)

## Réponse

En cas d'opération réussie, la réponse retournera l'identifiant du document, ainsi que l'URL permettant au signataire d'accéder à la signature OTP.

Exemple de réponse :

```
{
  "id": 320074168,
  "url": "https://demo-confiance.dfast.fr/boucle-sms/documentView?id=4URdeqt1oupdQlaKafjfMw=="
}
```

## 2.17. exportUsersData

La méthode "exportUsersData" permet de lister l'ensemble des utilisateurs, leurs affectations dans des circuits, leurs rôles dans ce circuit ainsi que le nom du groupe si l'utilisateur en fait partie.

### (GET)

<https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/exportUsersData?siren=<siren>>

---

## Paramètres

**siren** (obligatoire) - Entier, 9 caractères.

Le siren du client. Il s'agit de l'identifiant unique permettant d'identifier le client chez FAST.

## Réponse

En cas d'opération réussie, la réponse retournera un JSON.

## 2.18. *synchronizationUsers*

ATTENTION, l'utilisation de cette méthode doit se faire dans un cadre très particulier, il doit avoir fait le sujet de discussion avec FAST pour sa mise en place.

Cette Méthode permet de synchroniser les utilisateurs d'une collectivité sur le Parapheur.

Si un user est dans le json ET dans le parapheur : il est mis à jour en fonction des informations fournies

Si un user est dans le json MAIS pas dans le parapheur : il est créé sur le parapheur.

Si un user n'est pas dans le csv MAIS est dans le parapheur : il est supprimé du parapheur.

## (PUT)

<https://int-parapheur.dfast.fr/parapheur-ws/rest/v1/synchronization/<siren>/users>

## Paramètres

**siren** (obligatoire) - Entier, 9 caractères.

Le siren du client. Il s'agit de l'identifiant unique permettant d'identifier le client chez FAST.

**body** (obligatoire) - JSON

Le body de la méthode doit contenir le json permettant de mettre à jour la base utilisateur

## Exemple de body

```
[
  {
    "authenticationType": "CERTIFICAT", // CERTIFICAT ou BASIC
    "email": "email@domaine.com",      // Un email valide
    "firstName": "prénom",
    "lastName": "nom",
    "groups": ["groupe1", "groupe2", ...]
  },
  ...
]
```

## Réponse

La méthode retournera rien, si tout s'est bien passé et la liste de tous les utilisateurs problématiques et leurs erreurs sinon.