



SERVICE FAST-PARAPHEUR

GUIDE D'INTERFAÇAGE

Version : **3.4**

Date de la version : **01 Juin 2022**

Auteur : **Support Editeurs**

Société : **DOCAPOSTE FAST**

Confidentialité : **Diffusion restreinte**

Nombre de pages : **34**

→ Ce document est la propriété de DOCAPOSTE FAST. Son usage est réservé à l'ensemble des personnes habilitées selon leur niveau de confidentialité. Sa reproduction est régie par le Code de la propriété intellectuelle qui ne l'autorise qu'à l'usage privé du copiste. ←



SOMMAIRE

1.	Introduction	4
1.1.	Le Service FAST-Parapheur	4
1.1.1.	Usages	4
1.1.2.	Documents bureautique	4
1.1.3.	PES déjà signés et PES-PJ.....	4
1.1.4.	Présentation de l'interconnexion au service FAST-Parapheur	4
1.1.5.	Références	6
1.2.	Fonctionnement du service	6
1.2.1.	Authentification	6
1.2.2.	Accès au service (utilisateurs)	7
1.2.3.	Workflows sur FAST Parapheur	7
1.2.4.	Gestion des flux « PES Retour ».....	8
1.2.5.	Gestion des flux « Chorus ».....	8
1.2.6.	Paramètres à connaître et présentation sur le portail.....	8
1.2.7.	Vérifications au dépôt	9
1.2.8.	Informations à communiquer, compte de test.....	10
1.3.	Choix de l'interface (WebDAV ou Web Service).....	10
2.	Connecteur Web Service (SOAP-REST) – interface recommandée	11
2.1.	Dépôt de fichier	11
2.2.	Cas de la signature par OTP (one-time password).....	12
2.3.	Retours de la plateforme.....	15
2.4.	Bonnes pratiques à respecter par l'éditeur (ou le client)	16
2.5.	Détail des Méthodes des Web Services SOAP	17
2.5.1.	download.....	17
2.5.2.	upload.....	17
2.5.3.	uploadMeta.....	19
2.5.4.	uploadOTPInformation	21
2.5.5.	History.....	22
2.5.6.	getFdc	24
2.5.7.	getmetas.....	24
2.5.8.	delete	24
2.5.9.	downloadAcknowledgement	25
2.5.10.	listRemainingAcknowledgements.....	25
2.5.11.	downloadChorus	25
2.5.12.	listRemainingChorus	26
2.5.13.	downloadPesRetour	26
2.5.14.	listRemainingPesRetour	26
2.5.15.	getRefusalMessage.....	26
2.5.16.	getSmsUrl	26
2.5.17.	getMultildsSmsUrl.....	27
2.5.18.	uploadBSMS	28
2.5.19.	otpUpload	28
2.5.20.	getSMSHistory.....	30
2.5.21.	exportUsersData.....	31
2.5.22.	search.....	31
2.5.23.	UploadOnDemand ou Circuits/Upload (Circuit à la volée)	33
2.5.24.	Détails de la méthode de synchronisation des utilisateurs	35
2.6.	Détail des Méthodes des Web Services REST	36
2.6.1.	download.....	36
2.6.2.	upload.....	36



2.6.3.	UploadOnDemand ou Circuits/Upload (Circuit à la volée)	37
2.6.4.	uploadMeta.....	40
2.6.5.	uploadOTPInformation	42
2.6.6.	History.....	43
2.6.7.	getFdc	45
2.6.8.	getmetas.....	45
2.6.9.	delete	46
2.6.10.	downloadAcknowledgement	46
2.6.11.	downloadPesRetour	46
2.6.12.	downloadChorus	47
2.6.13.	listRemainingAcknowledgements.....	47
2.6.14.	listRemainingChorus.....	47
2.6.15.	listRemainingPesRetour	48
2.6.16.	getRefusalMessage.....	48
2.6.17.	getSmsUrl	49
2.6.18.	uploadOTP.....	49
2.6.19.	exportUsersData	51
2.6.20.	Search.....	51
2.6.21.	Détails de la méthode de synchronisation des utilisateurs	53
2.7.	Ajout d'annexe en mode connecteur WebService et Webdav.....	54
3.	Outil SIGNATURE serveur - Web Service (REST)	55
3.1.	Caractéristiques du service.....	55
3.2.	Utilisateurs.....	55
3.3.	Accessibilité	55
3.4.	Les différents cas d'utilisation.....	56
3.4.1.	La signature serveur.....	56
3.4.2.	La signature serveur avec Authentification par Certificat	56
3.4.3.	La signature serveur par SMS / Mail.....	56
3.4.4.	La signature serveur avec certificat à la volée.....	57
3.5.	Détail des Méthodes des Web Services.....	57
3.5.1.	signServerDoc	57
3.5.2.	uploadSA.....	58
3.5.3.	uploadSMS.....	58
3.5.4.	uploadSGC.....	59
3.5.5.	downloadDoc	59
3.5.6.	docInformation.....	60
3.5.7.	generateSAUrl.....	60
3.5.8.	generateSMSUrl.....	61
4.	Connecteur « WebDAV »	62
4.1.	Dépôt de fichier.....	62
4.1.1.	Ajout Métadonnées en mode Connecteur Webdav.....	63
4.2.	Organisation des répertoires.....	63
4.3.	Retours de la plateforme.....	66
4.4.	Gestions des erreurs suite au dépôt	67
4.5.	Bonnes pratiques à respecter par l'éditeur (ou le client)	68
4.6.	Exemples de requêtes WebDAV avec cURL	70



1. Introduction

1.1. Le Service FAST-Parapheur

1.1.1. Usages

FAST-Parapheur est un service de signature et de télétransmission de DOCAPOSTE FAST, destiné à :

- signer des documents de natures diverses (PES, PDF, bureautique, ...)
- signer des documents en masse (signature serveur)
- offrir une interface aux utilisateurs depuis un portail web ou un équipement mobile
- offrir une interface applicative aux serveurs du client
- être un point central de communication vers les services FAST de signature électronique, de télétransmission de fichiers PES, d'archivage électronique à valeur probante, ...

1.1.2. Documents bureautique

Le service FAST-Parapheur permet de gérer tous les documents bureautiques tels que des bons de commandes, des notes internes, des arrêtés, des délibérations, des propositions commerciales, des pièces de marché, etc... Il supporte les formats suivants :

PAdES : pour les documents pdf

XAdES : pour les documents bureautiques (Word, Excel, images, ...)

L'usage du format PAdES et de documents pdf est vivement conseillé pour son apport ergonomique.

1.1.3. PES déjà signés et PES-PJ

Lors de l'envoi d'un fichier PES déjà signé ou de fichiers PES-PJ, le service FAST-Parapheur peut être configuré pour passer directement à l'étape de transmission vers Helios, sans étape de signature.

La signature des fichiers PES reçus peut être de l'un ou l'autre des 2 types prévus par la norme : signature globale, signature par bordereau.

1.1.4. Présentation de l'interconnexion au service FAST-Parapheur

Le service **FAST-Parapheur** dissocie la partie métier (toujours assurée par l'éditeur) et les parties signature électronique / télétransmission sécurisée (signature et télétransmission sur le portail FAST-Parapheur). Il met en œuvre une interface simple et ouverte, basée sur des normes et des outils standards.

Les principaux avantages sont :

L'éditeur n'a pas besoin de maintenir de code spécifique pour FAST au niveau de la signature



électronique, ni d'avoir à inclure dans son logiciel des librairies externes (comme c'était par exemple le cas avec les anciennes API FAST). Il n'a donc plus besoin de maintenir de compétences spécifiques et techniques sur la signature électronique, puisque pris en charge par les équipes de DOCAPOSTE FAST. Il peut se concentrer sur l'implémentation et le suivi des évolutions réglementaires des normes métier, telle que la norme PES du Ministère des finances ; L'accrochage avec FAST est simple et rapide à intégrer puisque les échanges de fichiers se font par le biais de l'une des solutions standard, normée et sécurisée suivantes, au choix de l'éditeur :

- utilisation de « Web Services » sur HTTPS (connecteur Web Service) : mise à disposition de commandes correspondant à autant d'ordres donnés au serveur FAST ;
- utilisation du protocole « WebDAV » sur HTTPS (connecteur WebDAV) : mise à disposition de répertoires dédiés au client (dossiers d'envoi, dossiers de retour pour les acquittements et les documents signés, dossier « PES_Retour ») reposant sur ce protocole d'échange standard.

Le choix du connecteur Web Service est **fortement recommandé** par DOCAPOSTE FAST.

Dans chaque situation, le transport au-dessus de la couche HTTPS et le fait de laisser l'initiative de l'établissement de la connexion au serveur du client facilitent le déploiement en clientèle en évitant le paramétrage de routages réseau spécifiques.

De manière générale, la répartition des rôles se présente ainsi :

Rôle	Composant
Création des fichiers (PES, pdf, ...)	Logiciel Métier / logiciel bureautique
Dépôt des fichiers	Dépôt automatique : logiciel Métier Dépôt manuel : portail FAST Parapheur
Workflows de visa, signature, transmission, archivage, ...	Portail FAST-Parapheur (ces workflows peuvent également être suivis en amont du dépôt au sein d'un logiciel métier du client. Par exemple dans le cas d'un parapheur interne)
Gestion des retours (acquittements, fichiers signés, PES_Retour, ...)	Récupération automatique : logiciel Métier Récupération manuelle : portail FAST Parapheur
Interface de suivi des workflows (consultation des envois, recherche)	Logiciel Métier et/ou FAST-Parapheur



1.1.5. Références

Références Externes

Nom	Adresse
Protocole WebDAV	http://www.webdav.org/
Architectures Web Services	http://www.w3.org/TR/ws-arch/
Programme (DGCP) Helios	http://www.collectivites-locales.gouv.fr/dematerialisation-chaine-comptable-et-financiere-0
Norme d'échange Helios (DGCP)	http://www.collectivites-locales.gouv.fr/protocole-dechange-standard-pes-0

Informations DOCAPOSTE FAST

Nom	Adresse
Assistance Editeurs	https://docapost-fast.mantishub.io
Portail FAST-Parapheur de test	https://demo-parapheur.dfast.fr/
URL d'accès au Web-Services SOAP	https://demo-parapheur.dfast.fr/parapheur-soap/soap/v1/Documents?wsdl
URL D'accès au Web-Services REST	https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/
URL d'accès à la plateforme REST	https://demo-parapheur.dfast.fr/oss/

1.2. Fonctionnement du service

1.2.1. Authentification

L'accès à la plateforme FAST que ce soit sur le portail ou via le connecteur ne peut se faire que via un certificat électronique provenant d'une autorité de certification reconnue.

L'accès via le connecteur s'effectue à l'aide d'un certificat logiciel : certificat personnel ou certificat serveur de type 'Serveur Client RGS *'.



1.2.2. Accès au service (utilisateurs)

Le service est mis à disposition des clients sur une plateforme internet sécurisée. L'application est développée en mode SaaS (*Software as a Service*) et l'accès au portail requiert simplement un accès à internet (et un certificat électronique pour l'authentification).

Les signataires doivent installer un plugin de signature, développé par DOCAPOSTE FAST.

FAST-Parapheur s'adresse aux :

- agents en charge de l'alimentation du parapheur et de la récupération des documents signés,
- valideurs qui apposent leur visa sur un document avant envoi pour signature,
- signataires, habilités à signer électroniquement les documents.

1.2.3. Workflows sur FAST Parapheur

Chaque compte client est identifié par un SIREN et comprend un ou plusieurs workflows. Le workflow est le parcours que va suivre un document suite à son dépôt. Le Workflow est également appelé Cinématique ou Circuit ou Parcours de signature.

Chaque workflow possède son propre paramétrage (signature manuelle, signature serveur, signature OTP, circuit de validation, etc. etc.) et ses propres utilisateurs.

Pour un utilisateur travaillant sur son logiciel métier, les étapes du workflow sont les suivantes :

- Saisie du document par l'agent dans le logiciel métier ;
- Choix de la cinématique vers lequel le ou les documents seront déposés (*) ;
- Dépôt du ou des documents par le logiciel métier, via le connecteur ;
- Déclenchement automatique du workflow ;
- Après exécution du workflow, récupération du retour (document signé, acquittement Helios) par le logiciel métier ;
- Connecteur WebDAV : après récupération du retour par le logiciel métier, suppression de l'AR dans le répertoire « Out » ;
- Poursuite du traitement dans le logiciel métier

(*) Choix de la cinématique – important

Les workflows sont paramétrés à l'avance avec le client, selon ses besoins et son mode de fonctionnement :

- Par entité ou direction (mairie, CCAS, DSI, service marchés, ...) ;
- Par type de document (dépense, recette, note de frais, bon de commande, ...)
- Par budget (général, stationnement, ligne de produit, ...)
- Par signataire (Président, Vice-Président, Directeur, ...)



Le logiciel métier doit donc être en mesure de différencier le dépôt vers les différents workflows (et la récupération des retours), en fonction de tout ou partie de ces critères.

1.2.4. Gestion des flux « PES Retour »

Le service FAST-Parapheur permet de gérer le nouveau flux « PES_Retour » défini par le pôle Helios de la DGFIP, émis de la trésorerie vers la collectivité.

Après paramétrage, ces flux sont accessibles :

- Web Service : dans la réponse à la commande d'interrogation des flux à récupérer
- WebDAV : dans un dossier « PES_Retour » du répertoire out, comme les acquittements

1.2.5. Gestion des flux « Chorus »

Le service FAST-Parapheur permet également de gérer les flux « Chorus » défini par le Ministère de l'Economie et des Finances, il permet de récupérer les factures issues de la plate-forme CPP du ministère des Finances.

Après paramétrage, ces flux sont accessibles :

- Web Service : dans la réponse à la commande d'interrogation des flux à récupérer
- WebDAV : dans un dossier « chorus-pivot » du répertoire out, comme les acquittements

1.2.6. Paramètres à connaître et présentation sur le portail

Les paramètres à prendre en compte au niveau de l'interface sont les suivants :

- SIREN du compte
- Identifiant du workflow
- Libellé (ou label), correspondant à un code de regroupement des fichiers (date du jour, nom du budget, nom du service interne, ...)

Note Information : le libellé est optionnel. Dans le cas où il n'est pas utilisé, la création automatique d'un libellé égal à la date du jour AAAA-MM-JJ sera paramétrée dans l'administration du Parapheur.

Ces paramètres apparaissent de la manière suivante sur le portail :



Parapheur Electronique | Tableau de Bord | Dépôt | Recherche

bureautique | pes | pes_pj

2014-12-16

Fichiers en attente d'un retour de la trésorerie: 1

<input type="checkbox"/>	Nom du fichier	Date de dépôt	Signataire	Etat
<input type="checkbox"/>	pes1Mo1.xml (838,5 ko)	16/12/2014	C-Davy SIGNATAIRE	Envoyé

Télécharger

TesoT

Retours de la trésorerie: 1

<input type="checkbox"/>	Nom du fichier	Date de dépôt	Signataire	Etat
<input type="checkbox"/>	pes1Mo2.xml (838,5 ko)	16/12/2014	C-Davy SIGNATAIRE	Acquittement

Classer | Télécharger

pas de libellé

Fichiers à transmettre pour signature: 5

<input type="checkbox"/>	Nom du fichier	Date de dépôt	Signataire	Etat
<input type="checkbox"/>	pesTarek_Testo.xml (986,2 ko)	29/12/2014		Préparé
<input type="checkbox"/>	pes1Mo9.xml (819,6 ko)	29/12/2014		Préparé
<input type="checkbox"/>	pes1Mo10.xml (819,6 ko)	29/12/2014		Préparé
<input type="checkbox"/>	PR_depenses1-2015-BO-1.xml (6,2 Mo)	14/01/2015		Préparé
<input type="checkbox"/>	pesTarek_Test1.xml (986,2 ko)	29/12/2014		Préparé

Envoyer pour signature | Supprimer | Télécharger

Workflows (indicated by red arrows pointing to 'pes' and 'pes_pj')

Libellés (ou label) (indicated by red arrows pointing to 'TesoT' and 'pas de libellé')

1.2.7. Vérifications au dépôt

Suite au dépôt, le serveur FAST Parapheur procède immédiatement aux vérifications suivantes :

- Format du fichier selon ce qui a été paramétré (pdf pour la signature PAdES, xml pour les fichiers PES, ...)
- Unicité du fichier : basée sur le triplet cinématique/libellé/nom du fichier
- Unicité du contenu du fichier PES : basée sur l'unicité de la valeur de la balise NomFic parmi tous les clients
- Respect de la norme PES (format xml) : respect du schéma PES défini par le pôle Helios (les vérifications métier, sur le contenu, étant effectuées sur le guichet XML Helios)
- Taille du fichier : limité fixée à 100 Mo à la date du présent document. Il est vivement conseillé de ne pas dépasser 50 Mo en raison de l'impact sur la facilité d'utilisation du service

En cas de problème, le fichier n'est pas pris en compte et un email est envoyé à l'acteur à l'origine du dépôt.



Note : même si l'organisation du contenu des fichiers PES ne fait l'objet d'aucune vérification, il est vivement conseillé de ne pas dépasser les limites de 20 bordereaux par fichier et 1.000 pièces par bordereau.

1.2.8. Informations à communiquer, compte de test

L'éditeur (ou le client) précisera dès que possible :

- en cas de test de la signature PDF dans le navigateur, la configuration du poste client (version Windows et navigateur utilisées), afin de transmettre les outils de signature correspondant à l'environnement.
- l'interface choisie : Web Service ou WebDAV
- les fonctionnalités optionnelles qu'il souhaite ou non implémenter.

Un compte de test est fourni à l'éditeur, avec un jeu de certificats de test. Ce compte permet à l'éditeur de travailler dans un environnement identique à la production, selon le mode choisi (WebDAV, Web service).

1.3. Choix de l'interface (WebDAV ou Web Service)

Une interface ouverte est mise à disposition des éditeurs et des clients qui souhaitent automatiser les échanges avec le serveur. Elle repose sur l'utilisation des 2 protocoles suivants, au-dessus de la couche HTTPS : WebDAV ou Web Service

Le choix est laissé à l'initiative de l'éditeur ou du client, chaque solution ayant ses propres avantages :

- Web Service (SOAP ou REST) : retour synchrone de la part du serveur (prise en charge ou erreur). Les outils de développement, s'appuyant sur les standards, permettent la création automatique de programmes utilisant les services web de la plateforme (cf. chapitre suivant)
- WebDAV : simplicité pour déposer, récupérer, afficher et supprimer les fichiers. Possibilité d'utiliser des scripts (basés par exemple sur l'outil cURL)

Le choix du connecteur Web Service est **fortement recommandé** par DOCAPOSTE FAST.



2. CONNECTEUR WEB SERVICE (SOAP-REST) – INTERFACE RECOMMANDÉE

2.1. Dépôt de fichier

Lors du dépôt du fichier en mode « Web Service », il faut choisir le « circuit » dans lequel déposer le document (et de manière optionnelle un « libellé » qui permet d'ordonner les documents dans la cinématique).

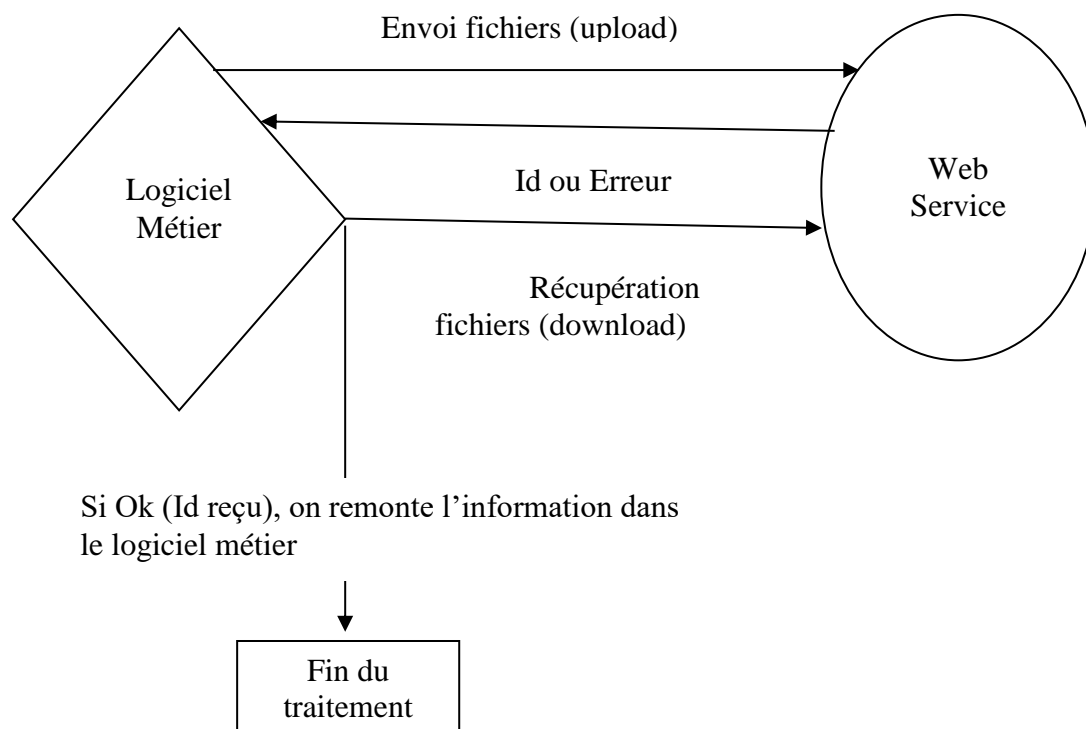
Dans la suite du document, on prendra à titre d'exemple les paramètres `circuitid = pes` et `label = TesoT`

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
- La cinématique d'envoi d'un fichier dans le workflow 'pes' se déroule comme ceci :
- Dépôt du fichier par l'intermédiaire de la Méthode « Upload » (ou « `uploadBSMS` »), complétée éventuellement d'un dépôt de méta-données (Méthode « `uploadMeta` »)(ou encore « `otpUpload` ») ;
- Vérification du dépôt correct du fichier : réponse avec l'identifiant unique ;
- Déclenchement du workflow et envoi d'une alerte mail au prochain acteur du workflow (Valideur, Signataire, ...) ;
- Poursuite des traitements et, en fin de parcours, envoi optionnel d'une alerte mail à l'acteur qui a initialement déposé le fichier ;
- En cas d'erreur suite au dépôt, FAST-Parapheur retourne un code erreur et envoie une alerte mail à l'acteur qui a initialement déposé le fichier ;
- Récupération du document à l'aide de l'identifiant unique ;



Le schéma ci-dessous reprend la cinématique d'échange exposée précédemment :



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.

2.2. Cas de la signature par OTP (one-time password)

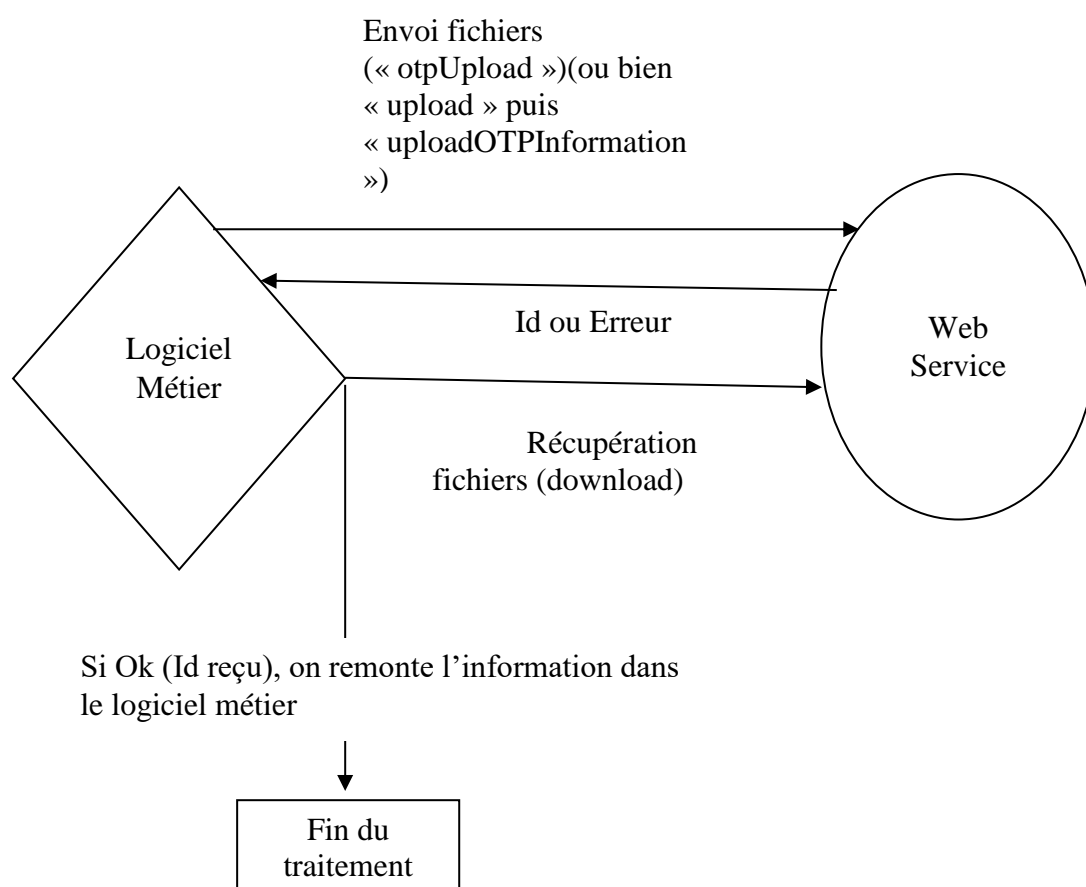
A titre d'exemple, la cinématique d'échange peut être la suivante :

- Création du document sur le logiciel métier du client ;
- L'utilisateur, connecté sur le logiciel métier du client, clique sur Signer ;
- Le logiciel métier transmet à FAST-Parapheur le document à signer ;
- Le logiciel métier transmet à FAST-Parapheur les informations du client (email, numéro de téléphone) et les métadonnées éventuelles (otpUpload) ;
- Le logiciel métier transmet à FAST-Parapheur l'adresse de retour pour rediriger le client à l'issue de la signature. FAST-Parapheur donne dans la réponse web service le lien vers l'interface web dédiée à l'affichage du document à signer ;
- Le logiciel métier redirige l'utilisateur vers le portail web d'affichage du document à signer en utilisant le lien ci-dessus (ou communique le lien par mail à l'utilisateur) ;
- FAST-Parapheur génère le code unique d'authentification à durée de validité limité, puis l'envoie par SMS sur le portable de l'utilisateur ou par e-mail à l'adresse de celui-ci ;



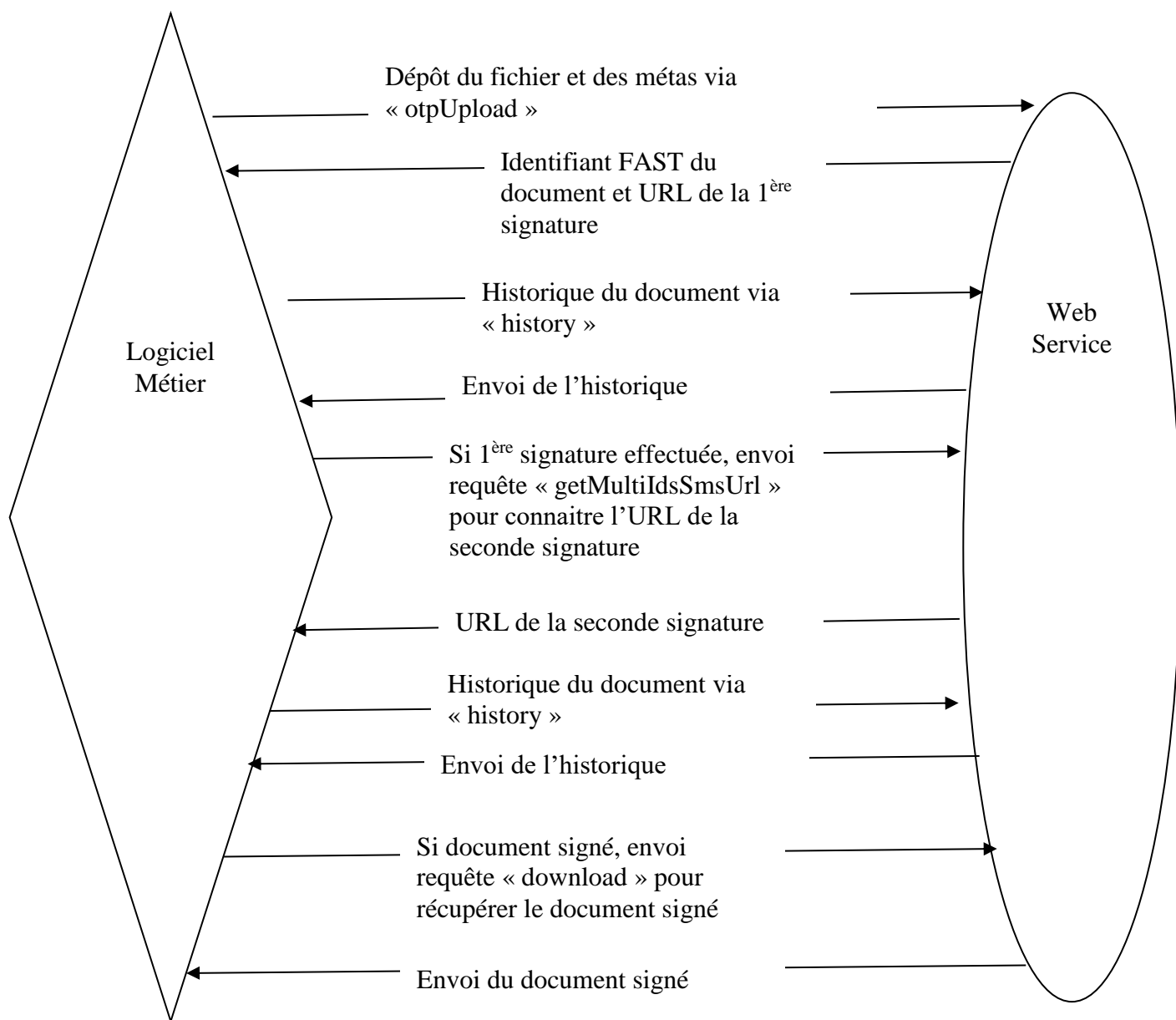
- L'utilisateur saisit le code après avoir visualisé le document à signer. Il clique sur Signer ;
- Si le code est correct, FAST-Parapheur effectue une signature serveur PAdES sur le document PDF et y intègre un pictogramme de signature ;
- FAST-Parapheur redirige l'utilisateur vers le site Internet du client en utilisant le lien renvoyé par le logiciel métier ;
- Le logiciel métier récupère le document signé ;

Le schéma ci-dessous reprend la cinématique d'échange exposée précédemment :





Vous trouverez ci-dessous un schéma d'exemple avec deux étapes de signature « OTP » dont les URL de signature sont transmis via les « Webservices » au logiciel métier qui remettra par un autre biais celles-ci à l'utilisateur :



Note : les méthodes disponibles via le Web Service sont décrites dans le chapitre suivant.



2.3. Retours de la plateforme

Les retours de la plateforme sont asynchrones. Il convient donc d'interroger la plateforme à intervalle régulier pour vérifier ce qui doit être récupéré.

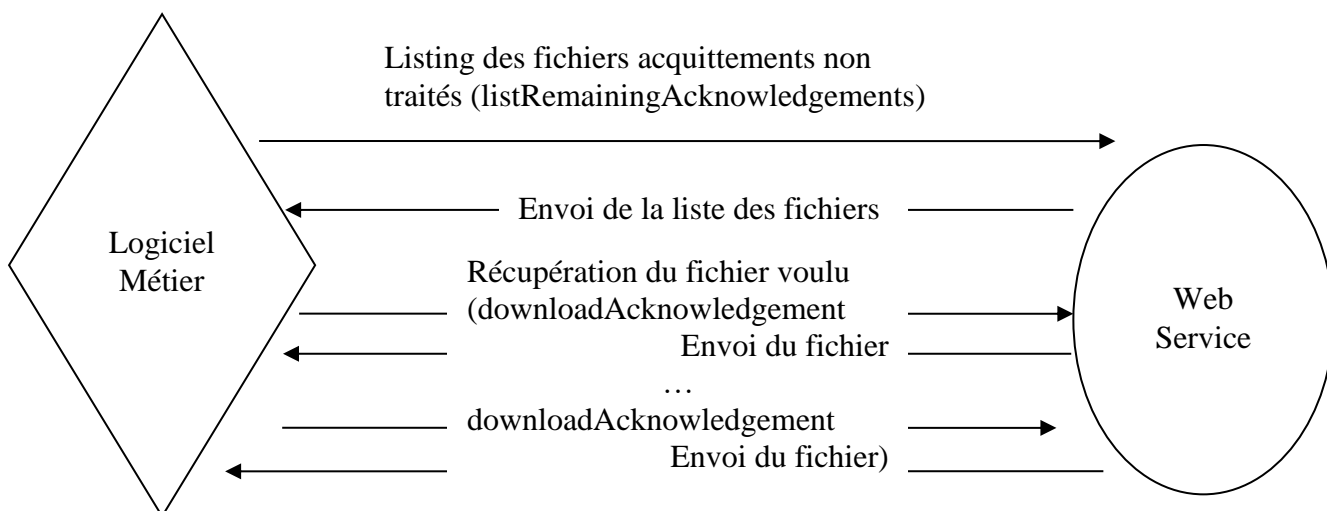
Les étapes pour la récupération se déroulent comme suit :

- Afin de vérifier la présence des retours, un listing est nécessaire (via la Méthode « listRemainingAcknowledgements ») ;
- Réponse de la plateforme avec la liste des fichiers présents ;
- Récupération du ou des documents à récupérer pour traitement par le logiciel métier (via la Méthode « downloadAcknowledgement ») ;

Ce type d'échange est également valable pour les flux « Pes-retour » et « Chorus-Pivot » transmis par le guichet Hélios ou Chorus qui ont leur propre méthode « Webservice ».

Le schéma ci-dessous reprend la cinématique d'échange exposée précédemment

Les retours de la plateforme sont synchrones.



Note : les méthodes disponibles via le Web Service sont décrites dans le chapitre suivant.



2.4. Bonnes pratiques à respecter par l'éditeur (ou le client)

Prévoir les mécanismes de reprise nécessaires en cas d'erreur, qu'elles proviennent de la procédure de connexion ou du niveau applicatif, et de remonter les informations nécessaires à l'utilisateur pour le guider dans son diagnostic.

Éviter absolument de mettre un libellé différent par fichier. Ce choix n'a aucun intérêt et complexifie la gestion des données sur FAST-Parapheur. Le fait de mettre un libellé différent par fichier, signifie que le libellé n'est probablement d'aucune utilité et nous recommandons alors de ne rien mettre ou de renseigner la date du jour dans le libellé.

Penser à mettre des « NomFic » unique, selon une règle sûre, qui ne risquent pas d'être utilisés par d'autres éditeurs / collectivités. Nous conseillons d'utiliser le numéro de SIREN, la date et l'heure (avec les millisecondes).

Les noms de fichiers ne doivent pas excéder les 150 caractères.

La commande « listRemainingAcknowledgements » ne doit être utilisée que toutes les 30 minutes au minimum afin d'éviter une surcharge des serveurs (il est conseillé par ailleurs d'éviter une heure pile afin de répartir les requêtes des différents clients).

A noter : la plateforme FAST est arrêtée toutes les nuits entre 4h05 et 4h15 pour procéder aux sauvegardes

Maintenir une liste à jour des fichiers récupérés et éviter d'utiliser la commande « downloadAcknowledgement » pour des fichiers qui n'existent pas ou qui ont déjà été récupérés.

Il est recommandé de garder sur le serveur du client une sauvegarde temporaire des fichiers récupérés, le temps nécessaire pour rejouer éventuellement leur intégration dans le logiciel métier. Cela est important en cas de problème dans la chaîne d'intégration des acquittements qui conduirait à une reprise.

Pensez à bien gérer dans le cas de la signature d'un document bureautique toutes les étapes (dépôt, suivi signature, récupération document signé, détection refus, récupération motif refus).

Pensez à bien suivre toute les étapes de dépôt avec métas pour coffres (dépôt avec métas, suivi dépôt Digiposte, suivi dépôt archivage).

Attention à l'utilisation de la méthode « Webservice » « getMultildsSmsUrl » pour le dépôt groupé OTP (1 code SMS pour plusieurs docs).



2.5. Détail des Méthodes des Web Services SOAP

2.5.1. download

Paramètre de la méthode « download » :

- documentId : L'identifiant unique du document
- acquit – Booléen555 : 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.

Erreurs retournées :

- "Erreur lors de la récupération du contenu du document" : pendant la récupération du contenu du document
- "Le nœud n'existe pas" : le document n'existe pas
- "Erreur lors de l'authentification, Utilisateur non reconnu": utilisation d'un certificat non valide
- "Erreur lors de l'authentification, accès refusé": utilisateur non autorisé à effectuer l'opération

2.5.2. upload

Paramètres de la méthode « upload » :

- subscriberId : L'identifiant unique du client déposant le document ;
- circuitId : Le circuit (workflow / cinématique) dans lequel est déposé le document ;
- label : Le libellé (sous-dossier) associé au document (optionnel mais balise même vide doit être présente dans la requête) ;
- comment : Un commentaire associé au document (optionnel) ;
- dataFileVO : Le document à déposer ;
- receiver : e-mail_destinataire (facultatif) : L'email de la personne à qui l'on souhaite envoyer le document après sa signature;

Cette méthode permet de déposer un document (*dataFileVO*) pour le client ayant l'identifiant *subscriberId* dans le circuit *circuitId* avec le libellé *label*.



Note :

- Le subscriberId correspond au siren du client
- Le triplé circuitId / label / nomDuDocument doit être unique (ce qui signifie que pour déposer deux documents avec le même nom, il convient de mettre un libellé différent).
 - **A éviter absolument** : mettre un libellé différent par fichier. ;
 - Un garde-fou a été implémenté sur le serveur FAST-Parapheur qui surcharge le label pour mettre la date du jour, si au moment du dépôt d'un nouveau fichier le nombre de labels existants dépasse une valeur fixé à ce jour à 10.000,
- Le paramètre comment est optionnel

Erreurs retournées :

- "Erreur lors de l'authentification, Utilisateur non reconnu": utilisation d'un certificat non valide
- "Erreur: l'abonne est inconnu ou l'utilisateur n'a pas les permissions pour y réaliser l'action demandée": abonnée inconnu ou abonné ne pouvant pas être accédé par cet utilisateur.
- "Le circuit demandé n'a pas été trouvé": circuit non existant pour l'abonné sélectionné
- "Vous n'avez pas la permission de réaliser cette opération." : l'utilisateur n'a pas les droits pour uploader un document
- "Fichier refusé : un fichier PES avec le même NomFic a déjà été envoyé.": un fichier ayant le même NomFic est déjà présent sur le parapheur.
- "Fichier refusé : le fichier PES ne contient aucun bordereau."
- "Extension du fichier non autorisée."
- "Le fichier envoyé est vide."
- "Ce fichier n'est pas un document PES.": message retourné si la cinématique traite des documents de type PES et que le fichier n'en est pas un.
- "Fichier refusé : PDF protégé en écriture."
- "Le fichier PES-PJ créé n'est pas valide.": fichier de type PES-PJ non valide.
- "Fichier déjà déposé par un agent."
- "Fichier contenant une mauvaise séquence de caractères."
- "La lecture du fichier a échoué."
- "Fichier trop volumineux."
- "Erreur lors du traitement du document PES"



2.5.3. 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
- Digiposte
- Maileva

Paramètres de la méthode « uploadMeta » :

- documentId : l'identifiant unique du document
- arg1 : un fichier xml contenant les métadonnées correspondant au document.

Cette méthode permet le Dépôt de métadonnées *Metas* qui seront associées au document ayant pour identifiant *documentId*.

Les métadonnées ne sont paramétrables que par administrateur de compte sur le circuit où est déposé le document.

Erreurs retournées :

- "Erreur lors de l'authentification, Utilisateur non reconnu": utilisation d'un certificat non valide
- "Le document n'existe pas": document non existant
- "Vous n'avez pas la permission de réaliser cette opération."
- "Le document n'est pas à l'état 'Prêt à définir les métadonnées'"
- "Aucun modèle de métadonnées défini pour cette cinématique"
- "La métadonnée Nom_de_la_métadonnée n'est pas définie pour cette cinématique"
- "La valeur de la métadonnée Nom_de_la_métadonnée n'est pas au format...." message d'erreur si la valeur ne correspond pas au format (date, valeur numérique, texte et expression régulière)



Exemple de fichier metadata

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

<meta-data-list>
  <meta-data name="TEXT001" value="John"/>
  <meta-data name="TEXT002" value="Doe"/>
  <meta-data name="TEXT003" value="0606060606"/>
  <meta-data name="NUM001" value="45812"/>
</meta-data-list>
```

Exemple de fichier metadata pour Digiposte

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

<meta-data-list>
  <!-- l'identifiant du compte -->
  <meta-data name="DIGIPOSTE_CLIENT_ID" value="001"/>
  <!-- Le type de document. FDIPL : Diplôme -->
  <meta-data name="DIGIPOSTE_CONTENT_TYPE" value="FDIPL"/>
  <!-- La durée de validité du document, chez Digiposte, en année -->
  <meta-data name="DIGIPOSTE_VALIDITY" value="10"/>
</meta-data-list>
```

Exemple de fichier metadata pour Maileva

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<meta-data-list>
  <!-- Expéditeur - Mail -->
  <meta-data name="MAILEVA_SENDER_MAIL" value="FAST"/>
  <!-- Expéditeur - Nom de société -->
  <meta-data name="MAILEVA_SENDER_ADDR1" value="FAST"/>
  <!-- Expéditeur - Statut Civil -->
  <meta-data name="MAILEVA_SENDER_ADDR2" value="statut OK"/>
  <!-- Expéditeur - Détail résidence-->
  <meta-data name="MAILEVA_SENDER_ADDR3" value="residence les lilas"/>
  <!-- Expéditeur - Numéro et rue -->
```



```

<meta-data name="MAILEVA_SENDER_ADDR4" value="120-122 rue
Réaumur"/>
<!-- Expéditeur - Détail postage -->
<meta-data name="MAILEVA_SENDER_ADDR5" value="Boite postal 40"/>
<!-- Expéditeur - Code postal et ville -->
<meta-data name="MAILEVA_SENDER_ADDR6" value="75002 Paris"/>
<!-- Destinataire - Nom de société -->
<meta-data name="MAILEVA_RECEIVER_ADDR1" value="DOCAPOSTE"/>
<!-- Destinataire - Statut Civil -->
<meta-data name="MAILEVA_RECEIVER_ADDR2" value="statut KO"/>
<!-- Destinataire - Détail résidence-->
<meta-data name="MAILEVA_RECEIVER_ADDR3" value="residence les
Sauls pleureurs"/>
<!-- Destinataire - Numéro et rue -->
<meta-data name="MAILEVA_RECEIVER_ADDR4" value="Place de la
république"/>
<!-- Destinataire - Détail postage -->
<meta-data name="MAILEVA_RECEIVER_ADDR5" value="Boite postale 2"/>
<!-- Destinataire - Code postal et ville -->
<meta-data name="MAILEVA_RECEIVER_ADDR6" value="75000 Ivry"/>
</meta-data-list>

```

2.5.4. uploadOTPInformation

Paramètre de la méthode « uploadOTPInformation » :

- documentId : L'identifiant unique du document
- Arg1 : Un fichier XML contenant les informations OTP pour le document.

Cette méthode permet le dépôt des informations OTP *Arg1* qui seront associées au document ayant pour identifiant *documentId*.

Le fichier XML doit respecter le modèle suivant :

```
<?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^{ère}, 1 pour la 2^{ème}, 2 pour la 3^{ème}, ..., n-1 pour la n^{ème}).

Note :

- La balise « filename » correspondant au fichier de métadonnée doit tout comme pour le mode « API » « Webdav » avoir une dénomination spécifique tel que « METAS_XXX.xml » sans le nommage du fichier de métadonnées sous cette forme, le dépôt sera rejeté.

2.5.5. History

Paramètre de la méthode « history » :

- documentId : L'identifiant unique du document

Cette méthode permet de récupérer l'historique du document et son état actuel.

Note : explication de certains états renvoyés par cette méthode :

- Préparé : le document est déposé sur le Parapheur ;
- Métadonnées définies : Les métadonnées ont été ajoutés ;
- Envoyé pour signature : Le document est prêt à être signé ;
- 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 est prêt à être validé ;
- 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 : 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 par SMS ou e-mail, 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 ;

Erreurs retournées :

- "Erreur lors de l'authentification, Utilisateur non reconnu": utilisation d'un certificat non valide
- "Erreur lors de l'authentification, accès refusé": utilisateur non autorisé à effectuer l'opération



2.5.6. **getFdc**

La méthode "getFdc" permet de récupérer la fiche de circulation d'un document.

Paramètre de la méthode « getFdc » :

- documentId : L'identifiant unique du document.

Réponse :

En cas d'opération réussie, la réponse sera la fiche de circulation sous format PDF.

2.5.7. **getmetas**

La méthode "getMetas" permet de récupérer les métadonnées d'un document.

Paramètre de la méthode « getmetas » :

- documentId : L'identifiant unique du document.

Réponse :

En cas d'opération réussie, la réponse sera les métadonnées sous format XML.

2.5.8. **delete**

Paramètre de la méthode « delete » :

- documentId : L'identifiant unique du document.

Cette méthode permet de supprimer un document ET son contexte (métadonnées, historique, ...) après un Refus de signature ou un Refus de validation

Note : la suppression est irréversible.

Erreurs retournées :

- "Erreur lors de l'authentification, Utilisateur non reconnu": utilisation d'un certificat non valide
- "Erreur lors de l'authentification, accès refusé": utilisateur non autorisé à effectuer l'opération
- "Erreur lors de la suppression. Le passage en suppression est impossible pour ce fichier": le document n'est pas en état d'être supprimé



2.5.9. **downloadAcknowledgement**

Paramètre de la méthode « downloadAcknowledgement » :

- documentId : L'identifiant unique du document.

Cette méthode permet de récupérer l'acquittement Hélios d'un document PES

Erreurs retournées :

- "Erreur lors de l'authentification, Utilisateur non reconnu": utilisation d'un certificat non valide
- "Erreur lors de l'authentification, accès refusé": utilisateur non autorisé à effectuer l'opération
- "Erreur lors de la récupération du contenu du document": Erreur lors de la récupération du contenu du document
- "Erreur lors la construction du document pour envoi"

2.5.10. **listRemainingAcknowledgements**

Paramètre de la méthode « listRemainingAcknowledgements » :

- siren : Le siren de l'abonné

Cette méthode permet de récupérer la liste des documents ayant des acquittements non traités par utilisation des Web Services SOAP.

Erreurs retournées :

- "Erreur lors de l'authentification, Utilisateur non reconnu": utilisation d'un certificat non valide
- "Erreur: l'abonne est inconnu ou l'utilisateur n'a pas les permissions pour y réaliser l'action demandée"
- "Erreur lors de la récupération des informations concernant les acquittements": le traitement ou la récupération des informations des acquittements a échoué.

2.5.11. **downloadChorus**

Paramètre de la méthode « downloadChorus » :

- documentId : L'identifiant unique du document.

Cette méthode permet de récupérer un fichier Chorus.



2.5.12. listRemainingChorus

Paramètre de la méthode « listRemainingChorus » :

- siren : Le siren de l'abonné

Cette méthode renvoie les identifiants des fichiers Chorus non récupérés via les Web Services SOAP. Ces identifiants sont à utilisés avec la méthode downloadChorus.

2.5.13. downloadPesRetour

Paramètre de la méthode « downloadPesRetour » :

- documentId : L'identifiant unique du document.

Cette méthode permet de récupérer un PES Retour.

2.5.14. listRemainingPesRetour

Paramètre de la méthode « listRemainingPesRetour » :

- siren : Le siren de l'abonné

Cette méthode renvoie les identifiants des PES Retour non récupérés via les Web Services SOAP. Ces identifiants sont à utilisés avec la méthode downloadPesRetour.

2.5.15. getRefusalMessage

Paramètre de la méthode « getRefusalMessage » :

- documentId : L'identifiant unique du document.

Cette méthode permet de récupérer le message de Refus sur le document s'il a été refusé, où le message de rejet si le document a été rejeté.

Note : le WebService renvoi une exception contenant le message « Le document n'a pas été rejeté ou « refusé » si le document n'a été ni refusé ni rejeté.

2.5.16. getSmsUrl

Note : cette méthode est en état DEPRECATED et ne doit plus être intégrée à compter du 01/11/2015. Merci de bien vouloir utiliser la méthode « getMultilidsSmsUrl ».

Paramètres de la méthode « getSmsUrl » :

- nodeRefId : L'identifiant unique du document.
- url : La chaine contenant l'URL vers laquelle sera redirigé l'utilisateur en fin d'activité sur l'application Boucle SMS.



Cette méthode permet de récupérer l'URL à laquelle est publié le service de signature sur l'application Boucle OTP.

Note :

- Si le deuxième argument est renseigné avec une chaîne de caractères vides, aucune redirection ne sera faite sur l'application OTP.
- Dans le cas d'une redirection, le paramètre GET « return » est ajouté à l'URL renseignée. La valeur de ce paramètre dépend du retour :
 - **REFUSED** : Si il y a redirection suite à un refus de signature.
 - **SIGNED** : redirection suite à une signature.
 - **ERROR** : redirection suite à une erreur.
 - **NONE** : redirection suite à un clic sur le bouton de redirection sans avoir signé ni refusé le document.

2.5.17. `getMultildsSmsUrl`

Paramètres de la méthode « `getMultildsSmsUrl` » :

- `nodeRefId` : Une liste d'identifiants uniques de documents.
- `url` : La chaîne contenant l'URL vers laquelle sera redirigé l'utilisateur en fin d'activité sur l'application OTP.

Cette méthode permet de récupérer une seule URL à laquelle est publié le service de signature des documents indiqués en paramètres sur l'application OTP.

Quand est indiqué plusieurs « `nodeRefId` », l'URL renvoyé en retour permet d'afficher sur une seule page l'ensemble des documents à signer permettant ainsi une signature par lot.

Note :

- Si le deuxième argument est renseigné avec une chaîne de caractères vides, aucune redirection ne sera faite sur l'application OTP.
- Dans le cas d'une redirection, le paramètre GET « return » est ajouté à l'URL renseignée. La valeur de ce paramètre dépend du retour :
 - **REFUSED** : Si il y a redirection suite à un refus de signature.
 - **SIGNED** : redirection suite à une signature.
 - **ERROR** : redirection suite à une erreur.
 - **NONE** : redirection suite à un clic sur le bouton de redirection sans avoir signé ni refusé le document.



2.5.18. uploadBSMS

Paramètres de la méthode « uploadBSMS » :

- label : le libellé associé au document
- comment : un commentaire associé au document
- subscriberId : l'identifiant unique du client déposant le document
- circuitId : le circuit dans lequel est déposé le document
- referrer : l'URL vers laquelle l'utilisateur sera redirigé en fin de traitement de l'application OTP
- dataFileVO : le document à déposer
- metaFileVO : le fichier xml contenant les métadonnées (au minimum les 3 métadonnées au format texte suivantes, dans l'ordre : le nom + prénom, l'email et le numéro de téléphone)

Cette méthode permet de réaliser l'opération « upload » dans le cadre de l'utilisation de l'application « OTP ». Le fichier *dataFileVO* sera dans un premier temps déposé sur le parapheur, définira les métadonnées du fichier à partir de *metaFileVO*.

Cette méthode retourne une liste de chaînes de caractères. Le premier élément de cette liste est l'identifiant unique du document accessible sur le Parapheur. Le deuxième élément est l'url à laquelle le document pourra être signé par l'application « OTP ».

Note : L'argument *metaFileVO* doit être un fichier de type xml.

Erreurs retournées :

- "le circuit dans lequel vous souhaitez uploader le fichier ne contient pas d'étape de signature pas envoi de SMS" : le paramétrage du circuit doit être vérifié
- "Merci de renseigner les métadonnées du (des) fichier(s)."
- "Vous n'avez pas la permission de réaliser cette opération." : l'utilisateur n'a pas les droits d'accès au document
- "Un des fichiers n'est pas à l'étape de signature par utilisation de la boucle SMS"

2.5.19. otpUpload

Paramètres de la méthode « otpUpload » :

- label : Le libellé associé au document (optionnel mais balise même vide doit être présente dans la requête) ;
- comment : Un commentaire associé au document ;
- siren : L'identifiant unique du client déposant le document ;
- circuitId : Le circuit dans lequel est déposé le document ;



- redirectionURL: l'URL vers laquelle l'utilisateur sera redirigé en fin de traitement de la signature OTP
- document : Le document à déposer
- otpData : Le fichier XML contenant les métadonnées.

Cette méthode permet de réaliser l'opération « upload » dans le cadre d'un circuit commençant par une signature OTP. Le *document* sera dans un premier temps déposé sur le parapheur, puis les informations OTP du document seront extraites de *otpData*.

Cette méthode retourne une liste de chaînes de caractères. Le premier élément de cette liste est l'identifiant unique du document accessible sur le Parapheur. Le deuxième élément est l'url publique à laquelle le document pourra être signé.

Note : L'argument *otpData* doit être un fichier de type XML respectant le modèle suivant :

```
<?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^{ère}, 1 pour la 2^{ème}, 2 pour la 3^{ème}, ..., n-1 pour la n^{ème}).

Note :

- La balise « filename » correspondant au fichier de métadonnée doit tout comme pour le mode « API » « Webdav » avoir une dénomination spécifique tel que « METAS_XXX.xml » sans le nommage du fichier de métadonnées sous cette forme, le dépôt sera rejeté.



2.5.20. **getSMSHistory**

Paramètres de la méthode « getSMSHistory » :

- siren : l'identifiant unique du client déposant le document ;
- circuitId : le circuit dans lequel est déposé le document ;
- date : le mois ou l'année de recherche
- status : un filtre sur le statut du SMS
- limit : le nombre maximum de fichier à récupérer

La méthode permet de lister l'ensemble des tentatives d'envoi de SMS réalisées depuis la plateforme.

Le champ « date » doit être une chaîne de caractère de la forme « yyyy-MM » avec les contraintes ou informations suivantes :

- l'année doit être supérieure ou égale à 2015 (composée de 4 chiffres)
- le mois doit être compris entre 00 et 12 (la valeur « 00 » permettant de lister les entrées sur toute l'année sélectionnée).

Le champ « status » doit contenir une des valeurs suivantes :

- pending
- waiting
- done
- not_pending
- not_waiting
- not_done

Note :

- Si le champ « status » est vide ou contient une autre chaîne de caractères, aucun filtre ne sera appliqué sur les résultats.
- Le champ « limit » est le nombre maximum d'objets retourné dans la réponse. Pour ne pas limiter le nombre d'objets à récupérer, renseigner ce champ avec la valeur 0.



2.5.21. exportUsersData

Paramètres de la méthode « exportUsersData » :

- siren : Le siren de la collectivité ;

La méthode 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.

Attention : L'utilisation de cette méthode requiert un compte administrateur. Il est nécessaire d'ajouter le header « Authorization : BASIC <base64(<login>:<mot_de_passe>)> » dans la requête.

2.5.22. search

La méthode "Search" permet d'effectuer une recherche sur le Parapheur. Il s'agit de la même recherche que sur le portail. L'utilisateur doit être admin sur le siren pour avoir l'autorisation d'effectuer la recherche.

Paramètres de la méthode « search » :

- siren (obligatoire): Le siren de la collectivité ;
- circuitId (facultatif) - 255 carac. max : Le circuit dans lequel le document doit être recherché ;
- label (facultatif) - 255 carac. max : Le dossier dans lequel le document a été déposé ;
- state (facultatif) - 255 carac. max : Ne retournera que les documents ayant cet état. Si vous renseignez "signed", alors les fichiers qui sont à signer seront retournés, mais les documents qui ont été signé, mais qui sont désormais classés, ne seront pas retournés ;
- uploadBy (facultatif) - 255 carac. max : Ne retournera que les documents déposés par cet utilisateur (email) ;
- uploadAfter (facultatif) - date (DD/MM/YYYY) : Ne retournera que les documents déposés après cette date ;
- uploadBefore (facultatif) - date (DD/MM/YYYY) : Ne retournera que les documents déposés avant cette date ;
- validateBy (facultatif) - 255 carac. Max : Ne retournera que les documents validés par cet utilisateur (email) ;
- validateAfter (facultatif) - date (DD/MM/YYYY) : Ne retournera que les documents validés après cette date ;
- validateBefore (facultatif) - date (DD/MM/YYYY) : Ne retournera que les documents validés avant cette date ;



- signBy (facultatif) - 255 carac. Max : Ne retournera que les documents signés par cet utilisateur (email) ;
- signAfter (facultatif) - date (DD/MM/YYYY) : Ne retournera que les documents signés après cette date ;
- signBefore (facultatif) - date (DD/MM/YYYY) : Ne retournera que les documents signés avant cette date ;
- meta (facultatif) – json : Ne retournera que les documents correspondant aux métadonnées renseignées.

Exemple de requête

```
<rs:search>
  <siren>XXXXXXXXXX</siren>
  <circuitId>bureautique</circuitId>
  <label>Dossier</label>
  <state>SentForSign</state>
  <uploadBy>agent@gmail.com</uploadBy>
  <uploadBefore>31/12/2022</uploadBefore>
  <uploadAfter>31/12/2020</uploadAfter>
  <validateBy>visa@gmail.com</validateBy>
  <validateBefore>31/12/2022</validateBefore>
  <validateAfter>31/12/2020</validateAfter>
  <signBy>sign@gmail.com</signBy>
  <signBefore>31/12/2022</signBefore>
  <signAfter>31/12/2020</signAfter>
  <meta>
  <parameters>
    <entry>
      <key>Meta 1</key>
      <value>777</value>
    </entry>
    <entry>
      <key>Meta 2</key>
      <value>666</value>
    </entry>
  </parameters>
</meta>
</rs:search>
```



Liste des états disponibles :

- Prepared : Le document est déposé sur FAST Parapheur ;
- SentForSign : Le document a été envoyé pour signature ;
- Signed: Le signataire a signé le document ;
- Refused : Le signataire a refusé le document ;
- SendForVisa : Le document a été envoyé pour validation d'un valideur ;
- Approved : Le valideur a validé le document ;
- Disapproved : Le valideur a rejeté le document (le rejet est l'inverse de la validation) ;
- SentToHELIOS : (Ne concerne que les circuits PES) Le document est envoyé chez HELIOS
- Acknowledged : (Ne concerne que les circuits PES) Le document a reçu l'acquittement d'HELIOS
- OTPvalidated (Ne concerne que les circuits OTP) : Le document est signé via OTP
- Filed : Le document est en fin de vie, il n'attend aucune action et n'apparaît plus sur le tableau de bord
- Archived : Le document a été archivé dans le coffre-fort électronique.

Réponse

En cas d'opération réussie, la réponse retournera un JSON avec la liste des « document_id » des fichiers correspondant à la recherche.

2.5.23. UploadOnDemand ou Circuits/Upload (Circuit à la volée)

Paramètre de la méthode « circuits/upload » :

- doc : Le document à déposer ;
- annexes (optionnel) (en cours de développement) : Les annexes du document ;
- siren : Le siren du compte ;
- circuit : la définition des étapes du circuit comme ceci :

<circuit>

<type> BUREAUTIQUE_PDF</type>

<steps>

<step>

<type>visa</type>

<members>monsieurA@cg00.fr</members>



```
</step>
<step>
<type>signature</type>
<members>madameB@cg00.fr</members>
</step>
</steps>
```

```
</circuit>
```

- email_destinataire (optionnel): Les emails des personnes à qui envoyer le flux une fois signées, séparé par une virgule. ATTENTION : la propriété s'appelle actuellement destinataire et va être renommé prochainement.
- agents (optionnel) : La liste des agents à rajouter au circuit, séparé par une virgule. La personne déposant le document est systématiquement agent, inutile de le rajouter ici.

Les types de circuit autorisés sont :

- BUREAUTIQUE_PDF
- BUREAUTIQUE_XML
- HELIOS_AVEC_TELETRANSMISSION
- HELIOS_SANS_TELETRANSMISSION

Les types de step autorisés sont :

- Signature
- Visa
- ServerSignature
- OTPSignature

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.

Exemple de la structure des erreurs remontées

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

Liste des erreurs :

```
UNAUTHORIZED(100, "Unauthorized"),
```



```
SUBSCRIBER_NOT_FOUND(111, "Subscriber not found"),  
INVALID_CIRCUIT_TYPE(117, "Invalid circuit type"),  
INVALID_STEP_TYPE(118, "Invalid step type"),  
INVALID_STEP_MEMBER(119, "Invalid step member"),  
UPLOAD_DOC_ERROR(120, "Upload doc error"),  
CIRCUIT_DEF_ERROR(121, "Circuit definition error");  
INVALID_STEP_MEMBER_AGENT(122, "The agent cannot be a step member");
```

2.5.24. Détails de la méthode de synchronisation des utilisateurs

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.

Paramètres de la méthode « synchronizeUsers » :

- siren : L'identifiant unique du client
- users : un json contenant tous les utilisateurs de la collectivité.

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 json MAIS est dans le parapheur : il est supprimé du parapheur.

Le json doit correspondre au format suivant (tous les champs sont obligatoires) :

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

Le web service renvoie, s'il y en a, une map des utilisateurs en erreur associés aux détails des erreurs s'ils sont disponibles.

Attention : L'utilisation de cette méthode requiert un compte administrateur. Il suffit d'ajouter le header « Authorization : BASIC <base64(<login>:<mot_de_passe>)> » dans la requête.

Soit d'utiliser un certificat dont l'utilisateur correspondant est administrateur.



2.6. Détail des Méthodes des Web Services REST

2.6.1. 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>

<https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/{{docId}}/download?fdc=false&acquit=true>

Paramètres

id (obligatoire) - 10 carac.

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.

2.6.2. 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

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.

content (obligatoire) - Fichier (multipart/form-data)

Le document à déposer sur le portail.

label - 255 carac. 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.

emailDestinataire - 255 caractères Max

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.

2.6.3. UploadOnDemand ou Circuits/Upload (Circuit à la volée)

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 caractères.

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

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 caractères max

Les emails des utilisateurs, séparés 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 concernent ni la signature, ni la validation et ajouter des commentaires et annexes.

comment - 255 caractères 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 adresses 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
- serverSignature
- OTPSignature

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.
102	Unauthorized	Votre Utilisateur n'est pas autorisé à effectuer des actions 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.



2.6.4. 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
- Digiposte
- Maileva

(PUT)

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.

meta (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="TEXT001" value="John"/>  
  <meta-data name="TEXT002" value="Doe"/>  
  <meta-data name="TEXT003" value="0606060606"/>  
  <meta-data name="NUM001" value="45812"/>  
</meta-data-list>
```

Exemple de fichier metadata pour Digiposte

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

```
<meta-data-list>  
  <!-- l'identifiant du compte -->  
  <meta-data name="DIGIPOSTE_CLIENT_ID" value="001"/>  
  <!-- Le type de document. FDIPL : Diplôme -->  
  <meta-data name="DIGIPOSTE_CONTENT_TYPE" value="FDIPL"/>
```



```
<!-- La durée de validité du document, chez Digiposte, en année -->  
<meta-data name="DIGIPOSTE_VALIDITY" value="10"/>  
</meta-data-list>
```

Exemple de fichier metadata pour Maileva

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
<meta-data-list>  
  <!-- Expéditeur - Mail -->  
  <meta-data name="MAILEVA_SENDER_MAIL" value="FAST"/>  
  <!-- Expéditeur - Nom de société -->  
  <meta-data name="MAILEVA_SENDER_ADDR1" value="FAST"/>  
  <!-- Expéditeur - Statut Civil -->  
  <meta-data name="MAILEVA_SENDER_ADDR2" value="statut OK"/>  
  <!-- Expéditeur - Détail résidence-->  
  <meta-data name="MAILEVA_SENDER_ADDR3" value="residence les  
  lilas"/>  
  <!-- Expéditeur - Numéro et rue -->  
  <meta-data name="MAILEVA_SENDER_ADDR4" value="120-122 rue  
  Réaumur"/>  
  <!-- Expéditeur - Détail postage -->  
  <meta-data name="MAILEVA_SENDER_ADDR5" value="Boite postal 40"/>  
  <!-- Expéditeur - Code postal et ville -->  
  <meta-data name="MAILEVA_SENDER_ADDR6" value="75002 Paris"/>  
  <!-- Destinataire - Nom de société -->  
  <meta-data name="MAILEVA_RECEIVER_ADDR1" value="DOCAPOSTE"/>  
  <!-- Destinataire - Statut Civil -->  
  <meta-data name="MAILEVA_RECEIVER_ADDR2" value="statut KO"/>  
  <!-- Destinataire - Détail résidence-->  
  <meta-data name="MAILEVA_RECEIVER_ADDR3" value="residence les  
  Saules pleureurs"/>  
  <!-- Destinataire - Numéro et rue -->  
  <meta-data name="MAILEVA_RECEIVER_ADDR4" value="Place de la  
  république"/>  
  <!-- Destinataire - Détail postage -->  
  <meta-data name="MAILEVA_RECEIVER_ADDR5" value="Boite postale 2"/>  
  <!-- Destinataire - Code postal et ville -->  
  <meta-data name="MAILEVA_RECEIVER_ADDR6" value="75000 Ivry"/>  
</meta-data-list>
```



2.6.5. uploadOTPInformation

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

(PUT)

<https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/otp/<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.

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 1ère, 1 pour la 2ème, 2 pour la 3ème, ..., n-1 pour la nième)

Note :

- Le nom du fichier correspondant au fichier de métadonnée doit tout comme pour le mode « API » « Webdav » avoir une dénomination spécifique tel que « METAS_XXX.xml » sans le nommage du fichier de métadonnées sous cette forme, le dépôt sera rejeté.

2.6.6. 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

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.

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
- Acquiescement Hélios : (Ne concerne que les circuits PES) Le document a reçu l'acquiescement 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, 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 ;

2.6.7. getFdc

La méthode “getFdc” permet de récupérer la fiche de circulation d’un document.

(GET)

<https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/v2/{{docId}}/getFdc>

Paramètres

docId (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.

Réponse

En cas d’opération réussie, la réponse sera la fiche de circulation sous format PDF.

2.6.8. getmetas

La méthode “getMetas” permet de récupérer les métadonnées d’un document.

(GET)

<https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/{{docId}}/metas>

Paramètres

docId (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.

Réponse

En cas d’opération réussie, la réponse sera les métadonnées sous format XML.



2.6.9. 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>

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.

2.6.10. downloadAcknowledgement

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

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.

2.6.11. downloadPesRetour

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

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.

2.6.12. downloadChorus

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

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.

2.6.13. listRemainingAcknowledgements

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]

2.6.14. listRemainingChorus

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 carac.

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]

2.6.15. listRemainingPesRetour

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>/pes-retour/remaining>

Paramètres

siren (obligatoire) - Entier, 9 carac.

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]

2.6.16. 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

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 "download"



2.6.17. getSmsUrl

La méthode "getSMSURL" est utilisé 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. A 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>

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=VlwFd9b6SJRtFJnFDh0bpYr8dBSFGewoEJNCAUttcw="
}
```

2.6.18. 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

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.

document (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 1ère, 1 pour la 2ème, 2 pour la 3ème, ..., n-1 pour la nème)

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=="  
}
```

Note :

- Le nom du fichier correspondant au fichier de métadonnée doit tout comme pour le mode « API » « Webdav » avoir une dénomination spécifique tel que « METAS_XXX.xml » sans le nommage du fichier de métadonnées sous cette forme, le dépôt sera rejeté.

2.6.19. 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 carac.

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.

2.6.20. Search

La méthode "Search" permet d'effectuer une recherche sur le Parapheur. Il s'agit de la même recherche que sur le portail. L'utilisateur doit avoir les droits sur le circuit en question pour avoir l'autorisation d'effectuer la recherche.

(GET)

<https://demo-parapheur.dfast.fr/parapheur-ws/rest/v1/documents/search>

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 (facultatif) - 255 carac. max

Le circuit dans lequel le document doit être recherché.

label (facultatif) - 255 carac. max

Le dossier dans lequel le document a été déposé.

state (facultatif) - 255 carac. max



Ne retournera que les documents ayant cet état. Si vous renseignez "signed", alors les fichiers qui sont à signer seront retournés, mais les documents qui ont été signé, mais qui sont désormais classés, ne seront pas retournés.

uploadBy (facultatif) - 255 carac. max

Ne retournera que les documents déposés par cet utilisateur (email).

uploadAfter (facultatif) - date (DD/MM/YYYY)

Ne retournera que les documents déposés après cette date

uploadBefore (facultatif) - date (DD/MM/YYYY)

Ne retournera que les documents déposés avant cette date

validateBy (facultatif) - 255 carac. max

Ne retournera que les documents validés par cet utilisateur (email).

validateAfter (facultatif) - date (DD/MM/YYYY)

Ne retournera que les documents validés après cette date

validateBefore (facultatif) - date (DD/MM/YYYY)

Ne retournera que les documents validés avant cette date

signBy (facultatif) - 255 carac. max

Ne retournera que les documents signés par cet utilisateur (email).

signAfter (facultatif) - date (DD/MM/YYYY)

Ne retournera que les documents signés après cette date

signBefore (facultatif) - date (DD/MM/YYYY)

Ne retournera que les documents signés avant cette date

meta (facultatif) - json

Ne retournera que les documents correspondant aux métadonnées renseignées.

Exemple de requête

```
{
  "siren": "xxxxxxxx",
  "circuit": "bureautique",
  "label": "Dossier",
  "state": "Signed",
  "uploadBy": "agent@gmail.com",
  "uploadAfter": "31/12/2021",
  "uploadBefore": "31/12/2022",
  "validateBy": "visa@gmail.com",
  "validateAfter": "31/12/2021",
  "validateBefore": "31/12/2022",
  "signBy": "sign@gmail.com",
  "signAfter": "31/12/2021",
  "signBefore": "31/12/2022",
  "meta":
  {
```



```
"email": "search@gmail.com",  
"montant": "451200"  
}  
}
```

Liste des états disponibles

- Prepared : Le document est déposé sur FAST Parapheur ;
- SentForSign : Le document a été envoyé pour signature ;
- Signed: Le signataire a signé le document ;
- Refused : Le signataire a refusé le document ;
- SendForVisa : Le document a été envoyé pour validation d'un valideur ;
- Approved : Le valideur a validé le document ;
- Disapproved : Le valideur a rejeté le document (le rejet est l'inverse de la validation) ;
- SentToHELIOS : (Ne concerne que les circuits PES) Le document est envoyé chez HELIOS
- Acknowledged : (Ne concerne que les circuits PES) Le document a reçu l'acquittement d'HELIOS
- OTPvalidated (Ne concerne que les circuits OTP) : Le document est signé via OTP
- Filed : Le document est en fin de vie, il n'attend aucune action et n'apparaît plus sur le tableau de bord
- Archived : Le document a été archivé dans le coffre-fort électronique.

Réponse

En cas d'opération réussie, la réponse retournera un JSON avec la liste des document_id des fichiers correspondant à la recherche.

2.6.21. Détails de la méthode de synchronisation des utilisateurs

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://demo-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.

Attention : L'utilisation de cette méthode requiert un compte administrateur. Il suffit d'ajouter le header « Authorization : BASIC <base64(<login>:<mot_de_passe>)> » dans la requête.

Soit d'utiliser un certificat dont l'utilisateur correspondant est administrateur.

2.7. Ajout d'annexe en mode connecteur Webservice et Webdav

Il est possible aujourd'hui d'uploader des annexes mais uniquement au moment de l'upload.

Pour cela, il faut envoyer un zip portant pour nom, le nom du document principal (identique dans le zip) suffixé de l'extension « .zip ».

Exemple:

J'envoie un fichier archivé « bonjour.pdf.zip » qui contient

- bonjour.pdf
- bonjour2.pdf
- hjjglshklgsd.pdf
- qfjdlvgfsdv.xml

Tous les fichiers doivent être posés à la racine (aucun répertoire ne doit être présent).

A la réception du zip, le document principal uploader au parapheur sera « bonjour.pdf » et tous les autres fichiers contenus dans le zip seront ajoutés en annexe à ce document.



Note : Liste des extensions autorisées :

xml,xhl,pdf,txt,jpeg,jpg,gif,png,bmp,html,rtf,tif,tiff,doc,docx,xls,xlsx,odt,ods,ots,zip,bin,cry,csv,exe, bat,pli,crypt,temp,kle,sig,cmp,p7m,rep,thmx,pde,pdp,p7s,db,htm,tmp,ini,pptx,pff,sh1,cle,ppt,css,j s,msg,emz,trc,pdr, pps,dat,rst,kfl,ret,p7b,mdi,docm,xlk

3. OUTIL SIGNATURE SERVEUR - WEB SERVICE (REST)

3.1. Caractéristiques du service

Le service Outil de Signature Serveur (OSS) est un service web qui permet la signature serveur de document, indépendant de plateforme Parapheur. Il existe plusieurs variations de la signature serveur :

- Signature serveur classique
- Signature serveur dont le lancement de la signature est effectué par un utilisateur disposant d'un code envoyé par SMS ou par mail ;
- Signature serveur dont le lancement de la signature est effectué par un utilisateur authentifié par certificat électronique ;
- Signature serveur avec certificat généré à la volée (à venir).

3.2. Utilisateurs

Il existe trois types de profils différents

- Admin : S'authentifie par certificat au portail d'administration. Il peut configurer les workflows du compte et avoir accès à différente page de supervisions/statistiques ;
- Agent : Il s'agit du profil ayant le droit d'appeler les Web Services d'OSS ;
- Signataire : Ils ne sont pas connus du système, mais ils peuvent accéder au page leur permettant de signer via SMS ou Mail, s'ils ont les informations nécessaires (URL de signature envoyé par mail ou via redirection client).

3.3. Accessibilité

L'URL de base pour l'environnement de recette est : <https://demo-parapheur.dfast.fr/oss>

Pour accéder aux différents Web Service, il y a plusieurs paramètres qui seront demandé et qui seront fourni par DOCAPOSTE Fast (sauf si vous avez accès au portail d'administration) :

- workflowId : L'identifiant unique du workflow utilisé. Chaque workflow défini son propre type de signature serveur, de certificat serveur et d'agent.



- metaFile : Le fichier de métadonnée associé au document. Un exemple de fichier de métadonnée peut être trouvé en Annexe. La structure dépendra de la configuration du workflow.

3.4. Les différents cas d'utilisation

3.4.1. La signature serveur

Les Web Services associés :

- signServerDoc: permet de signer un document ;
- (optionnel) downloadDoc : Permet de récupérer le document signé ;

3.4.2. La signature serveur avec Authentification par Certificat

Cas d'utilisation :

1. Monsieur X (client de l'éditeur) est sur le portail de l'éditeur et doit signer un document ;
2. L'éditeur envoie, par Web Service, le document à signer et une url (url_Editeur) à Fast.
 - a. En réponse au Web Service, Fast renvoie à l'éditeur l'identifiant unique du document et une url (url_Fast) ;
3. L'éditeur envoie Monsieur X vers l'url_Fast ;
4. Monsieur X arrive sur une page sur laquelle il doit s'authentifier par certificat. Une fois authentifié, il peut consulter son document et il peut cliquer sur Signer ;
 - a. Fast garde les informations d'authentification de Monsieur X ;
5. Monsieur X clique sur Signer. Fast signe le document avec le certificat serveur et redirige Monsieur X vers url_Editeur ;
6. Monsieur X est redirigé chez l'éditeur ;
 - a. Grâce à l'url_Editeur, l'éditeur a l'information comme quoi la signature du document a été effectué ;
7. L'éditeur récupère le document signé via Web Service ;
 - a. Dans la réponse du Web Service, l'éditeur récupère également le log d'authentification de Monsieur X.

Les Web Services associés :

- uploadSA : permet d'uploader le document sur OSS ;
- downloadDoc : Permet de récupérer le document une fois signé ;
- (optionnel) generateSAUrl : Permet de signer plusieurs documents en une fois ;
- (optionnel) docInformation : permet d'obtenir les informations liées au document.

3.4.3. La signature serveur par SMS / Mail

Les Web Services associés :



- uploadSMS : permet d'uploader le document sur OSS ;
- downloadDoc : Permet de récupérer le document une fois signé ;
- (optionnel) generateSMSUrl : Permet de signer plusieurs documents en une fois ;
- (optionnel) docInformation : permet d'obtenir les informations liées au document.

3.4.4. La signature serveur avec certificat à la volée

Cas d'utilisation :

1. Monsieur X (client de l'éditeur) est sur le portail de l'éditeur et doit signer un document;
2. L'éditeur envoie, par Web Service, le document à signer, les informations de Monsieur X (Nom, prénom, numéro de téléphone) et une url (url_Editeur) à Fast.
 - a. En réponse au Web Service, Fast renvoie à l'éditeur l'identifiant unique du document et une url (url_Fast) ;
3. L'éditeur envoie Monsieur X vers l'url_Fast ;
4. Monsieur X arrive sur une page sur laquelle il consulte son document, au même moment, un SMS avec un code est envoyé. Monsieur X renseigne le code reçu ;
5. Monsieur X clique sur Signer. Fast signe le document avec le certificat à la volée au nom de Monsieur X et redirige Monsieur X vers url_Editeur ;
6. Monsieur X est redirigé chez l'éditeur ;
 - a. Grâce à l'url_Editeur, l'éditeur a l'information comme quoi la signature du document a été effectuée ;
7. L'éditeur récupère le document signé via Web Service ;

Les Web Services associés :

- uploadSGC : Permet de déposer le document sur OSS ;
- downloadDoc : Permet de récupérer le document une fois signé ;
- docInformation : permet d'obtenir les informations liées au document

3.5. Détail des Méthodes des Web Services

3.5.1. signServerDoc

Cette méthode POST permet d'effectuer la signature serveur d'un document.

Paramètres de la méthode :

- workflowId : L'identifiant unique du workflow ;
- dataFile : Le document à signer ;
- filename : Le nom du document ;



- **metaFile** : (Optionnel) Le fichier de métadonnée du document. Uniquement si des métadonnées ont été définies dans le workflow. Selon le paramétrage, les métadonnées peuvent apparaître dans le pictogramme de signature ;
- **pictogramme** : La position du pictogramme de signature. Les coordonnées à fournir correspondent à la position (en cm) du point en bas à gauche du pictogramme.

Réponse de la méthode :

- **Header** :
 - *X-FAST-document-id* : l'identifiant unique du document
- **Body** : Le document (sauf si asynchrone est à « true »)

3.5.2. uploadSA

Cette méthode POST permet d'uploader un document à destination d'une signature serveur via une authentification par certificat.

Paramètres de la méthode :

- **workflowId** : L'identifiant unique du workflow ;
- **dataFile** : Le document à signer ;
- **filename** : Le nom du document ;
- **metaFile** : Le fichier de métadonnées du document ;
- **redirectionUrl** : L'URL vers lequel FAST doit rediriger le client en cas de réussite/échec de signature.

Réponse de la méthode :

- **Body** :
 - *url* : L'URL vers lequel l'éditeur doit envoyer son client, pour avoir accès à la page de signature
 - *docId* : L'identifiant

3.5.3. uploadSMS

Cette méthode POST permet d'uploader un document à destination d'une signature serveur via SMS ou Mail.

Paramètres de la méthode :

- **workflowId** : L'identifiant unique du workflow ;
- **dataFile** : Le document à signer ;
- **filename** : Le nom du document ;



- **metaFile** : Le fichier de métadonnées du document ;
- **redirectionUrl** : L'URL vers lequel FAST doit rediriger le client en cas de réussite/échec de signature.

Réponse de la méthode :

- **Body** :
 - **url** : L'URL vers lequel l'éditeur doit envoyer son client, pour avoir accès à la page de signature
 - **docId** : L'identifiant

3.5.4. uploadSGC

Cette méthode POST permet de déposer un document à destination d'une signature serveur avec certificat généré à la volée.

Paramètres de la méthode :

- **workflowId** : L'identifiant unique du workflow ;
- **dataFile** : Le document à signer. Cette propriété peut être renseigné plusieurs fois afin d'effectuer une signature par lot ;
- **firstname** : Le prénom du client ;
- **lastname** : Le nom de famille du client ;
- **email** : l'email du client ;
- **phoneNumber** (Optionnel) : Le numéro de téléphone du client ;
- **redirectionUrl** (Optionnel): L'URL vers lequel FAST doit rediriger le client en cas de réussite/échec de signature.
- **jsonPictogram** (Optionnel) : Les coordonnées, la taille et la page d'apposition du pictogramme. au format json. Exemple : {"x": 60, "y": 60, "width": 200, "height": 100, "page": 1} Le point 0/0 correspondant au point le plus en bas à gauche du PDF.

Réponse de la méthode :

- **Body** :
 - **url** : L'URL vers lequel l'éditeur doit envoyer son client, pour avoir accès à la page de signature
 - **o docId** : L'identifiant unique du document. Permet à l'éditeur de récupérer le document une fois signé

3.5.5. downloadDoc

Cette méthode GET permet de télécharger un document qu'il soit signé ou non.



L'URL de la méthode = URL de base + /downloadDoc/ + l'identifiant unique du document.

Exemple : <https://demo-parapheur.dfast.fr/oss/downloadDoc/100256>

3.5.6. docInformation

Cette méthode GET permet de récupérer diverses informations associées à un document.

La liste des informations est :

- id : identifiant du document ;
- filename : le nom du document ;
- creationDate : La date de création du document ;
- signatureDate : La date de signature du document ;
- purgeDate : La date de la suppression du document ;
- archiveDate : La date de l'archivage du document ;
- downloadDate : La date du premier téléchargement du document (via downloadDoc) ;
- dateToPurge : La date à laquelle le fichier sera supprimé ;
- metadata : Le fichier de métadonnée du document en BASE64 ;
- redirectionURL : l'URL de redirection fourni par l'éditeur ;
- logs : Les logs d'authentification pour la signature serveur avec authentification par certificat
- filesize : la taille du fichier
- smsexpirationDate : la date d'expiration du code SMS/Mail
- SMSRemainingTriesNb : le nombre d'essai restant pour le code SMS/Mail
- Smsdate : la date d'envoi du code SMS/mail

L'URL de la méthode = URL de base + /docInformation/ + l'identifiant unique du document

Exemple : <https://demo-parapheur.dfast.fr/oss/docInformation/100256>

3.5.7. generateSAUrl

Cette méthode POST permet de faire de la signature serveur par lot pour l'authentification avec certificat.

Paramètres de la méthode :



- ids : une liste d'identifiant de document séparé par des virgule. Exemple : 100257, 100258, 100259.
- url : l'URL vers laquelle FAST doit rediriger le client après signature (chez l'éditeur)

Réponse de la méthode :

- Body :
 - *ids : la liste des identifiants*
 - *url : L'URL vers laquelle l'éditeur doit envoyer son client pour pouvoir signer ses documents*

3.5.8. generateSMSUrl

Cette méthode POST permet de faire de la signature serveur par lot via SMS ou Mail.

Paramètres de la méthode :

- ids : une liste d'identifiant de document séparé par des virgule. Exemple : 100257, 100258, 100259.
- url : l'URL vers laquelle FAST doit rediriger le client après signature (chez l'éditeur)

Réponse de la méthode :

- Body :
 - *ids : la liste des identifiants*
 - *url : L'URL vers laquelle l'éditeur doit envoyer son client pour pouvoir signer ses documents*



4. CONNECTEUR « WEBDAV »

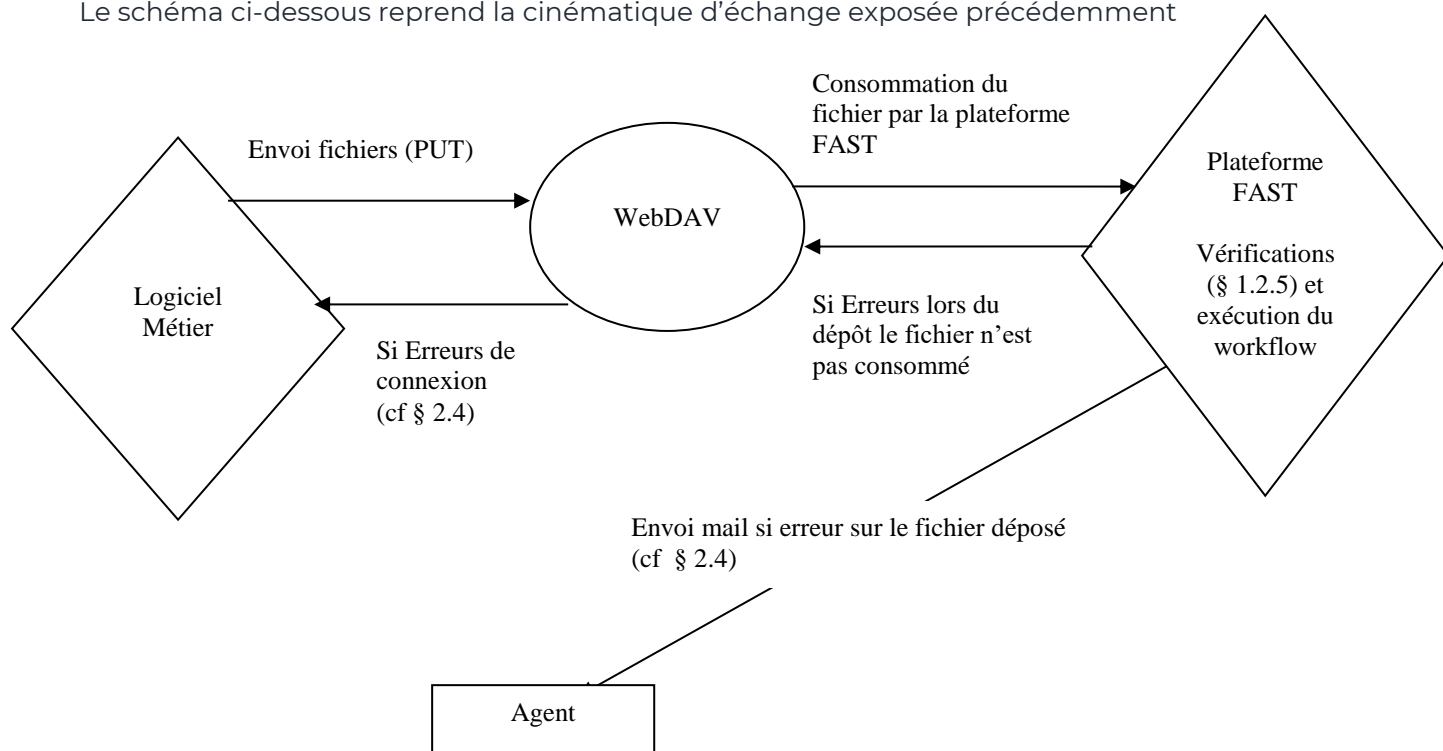
4.1. Dépôt de fichier

Le dépôt se fait dans le sous-répertoire du répertoire « in », portant l'identifiant du workflow. Des sous-dossiers peuvent être créés, lesquels seront répercutés sur le portail du parapheur en tant que 'libellés' (TesoT dans l'exemple ci-dessus).

La cinématique d'envoi d'un fichier dans le workflow 'pes' se déroule comme ceci :

- Création du sous-répertoire 'TesoT' dans le répertoire /SIREN/in/pes (optionnel)
- Dépôt du fichier dans le répertoire « WebDAV » ('/SIREN/in/pes' ou '/SIREN/in/pes/TesoT' selon ce qui a été mis en place ci-dessus), via la requête HTTP PUT ;
- Vérification du dépôt correct du fichier : code http 201 ;
- Déclenchement automatique du workflow et envoi d'une alerte mail au prochain acteur du workflow (Valideur, Signataire, ...) ;
- Poursuite des traitements et, en fin de parcours, envoi optionnel d'une alerte mail à l'acteur qui a initialement déposé le fichier ;
- En cas d'erreur suite au dépôt, FAST-Parapheur envoie une alerte mail à l'acteur qui a initialement déposé le fichier ;

Le schéma ci-dessous reprend la cinématique d'échange exposée précédemment





Note : Il incombe à l'éditeur de surveiller les répertoires partagés mis à sa disposition, et de veiller à les garder « vides » (signe que l'ensemble des fichiers ont été traités / récupérés).

Attention : bien que le code http 204 indique un dépôt réussi, ce code est relatif à une mise à jour du document. Cette mise à jour n'est pas remontée au portail. Il faut donc en cas de code 204 supprimer le document et le redéposer.

4.1.1. Ajout Métadonnées en mode Connecteur Webdav

Il est possible aujourd'hui d'uploader des métadonnées mais uniquement au moment de l'upload tout comme pour les annexes voir chapitre 2.7.

Pour cela, il faut envoyer un zip portant pour nom, le nom du document principal (identique dans le zip) suffixé de l'extension « .zip » puis du fichier de métadonnées avec pour entête « METAS_ » suivit du nom du fichier du document principal avec son extension « .pdf » avant l'extension « .xml » (exemple de fichier métas attendus présent dans les exemples fournis).

Exemple:

J'envoie un fichier archivé « `bonjour.pdf.zip` » qui contient

- `bonjour.pdf`
- `METAS_bonjour.pdf.xml`

Tous les fichiers doivent être posés à la racine (aucun répertoire ne doit être présent).

A la réception du zip, le document principal uploader au parapheur sera « `bonjour.pdf` », les métadonnées seront appliquées au document principal par la plateforme si le fichier « `xml` » est conforme aux métadonnées prévues sur la plateforme.

4.2. Organisation des répertoires

Comme indiqué précédemment, le service FAST-Parapheur permet de gérer différents workflows.

Ces workflows sont associés à un dossier (sous-répertoire) dans le répertoire WebDAV du compte, dossier présent dans le répertoire « in » et dans le répertoire « out ». Ces 2 dossiers sont créés par DOCAPOSTE FAST au moment du paramétrage du compte.

Par exemple, nous pourrions avoir l'organisation suivante :

SIREN

|--in

|--PES (envoi FAST fichiers de type PES)

|--PES-PJ (pièces justificatives de type PES)

|--Bureautique (fichiers bureautique à signer)



```
|--Marches (signature PDF pour réponses à AO)
|--out
  |--PES
  |--PES-PJ
  |--Bureautique
  |--Marches
```

Les dépôts dans un sous dossier de « in/PES » lanceront un workflow de type « envoi FAST », qui demandera une signature, puis transmettra le PES signé à Helios.

Les dépôts dans « Marches », quant à eux, lanceront un workflow de signature PDF par un groupe de signataires différent (configurés à l'avance), sans envoi vers la plateforme FAST.


Afin d'organiser les différents envois, le parapheur permet également la création de sous-dossiers de « libellés », dont les noms se retrouveront en catégories sur le portail FAST-Parapheur.

Prenons comme exemple l'organisation suivante :

```
SIREN
|--in
  |--PES (envoi FAST fichiers de type PES)
    |--2014-12-16
    |--facture
    |--TesoT
  |--bureautique (signature PDF pour réponses à AO)
    |--AO2010
    |--AO2011
|--out
  |--PES
    |--2014-12-16
    |-- facture
    |--TesoT
  |--bureautique
    |--AO2010
    |--AO2011
```

Après dépôt d'un PES dans chaque dossier, on obtient la vue suivante sur le service FAST-Parapheur




 Bienvenue C-Davy AGENT (C-Davy)
 [Accueil](#)
[Préférences](#)
[Aide](#)
[Déconnexion](#)

Parapheur Electronique
[Tableau de Bord](#)
[Dépôt](#)
[Recherche](#)

[bureautique](#)
[pes](#)
[pes_pj](#)

2014-12-16

Fichiers en attente d'un retour de la trésorerie: 1

<input type="checkbox"/>	Nom du fichier	Date de dépôt	Signataire	Etat
<input type="checkbox"/>	pes1Mo1.xml (838.5 ko)	16/12/2014	C-Davy SIGNATAIRE	Envoyé

[Télécharger](#)

TesoT

Retours de la trésorerie: 1

<input type="checkbox"/>	Nom du fichier	Date de dépôt	Signataire	Etat
<input type="checkbox"/>	pes1Mo2.xml (838.5 ko)	16/12/2014	C-Davy SIGNATAIRE	Acquittement

[Classer](#) [Télécharger](#)

facture

Fichiers envoyés pour signature: 1

<input type="checkbox"/>	Nom du fichier	Date de dépôt	Signataire	Etat
<input type="checkbox"/>	PR_-depenses-2014-BO-2122.xml (6.3 Mo)	16/12/2014		À signer

[Télécharger](#)

Fichiers à transmettre pour signature: 5

<input type="checkbox"/>	Nom du fichier	Date de dépôt	Signataire	Etat
<input type="checkbox"/>	pesTarek_Test0.xml (986.2 ko)	29/12/2014		Préparé
<input type="checkbox"/>	pes1Mo9.xml (819.6 ko)	29/12/2014		Préparé
<input type="checkbox"/>	pes1Mo10.xml (819.6 ko)	29/12/2014		Préparé
<input type="checkbox"/>	PR_-depenses1-2015-BO-1.xml (6.2 Mo)	14/01/2015		Préparé
<input type="checkbox"/>	pesTarek_Test1.xml (986.2 ko)	29/12/2014		Préparé

[Envoyer pour signature](#)
[Supprimer](#)
[Télécharger](#)

Ces dossiers WebDAV servent aussi à séparer les retours de FAST. En effet, ces dossiers de libellés sont répliqués dans le dossier « out », et reçoivent les acquittements des PES déposés dans le dossier « in ».

L'organisation complète des dossiers sera donc, dans le cadre de l'exemple du circuit PES :

SIREN

|--in

|--PES (envoi FAST fichiers de type PES)

|--2014-12-16

|--facture

|--TesoT

|--out

|--PES (envoi FAST fichiers de type PES)

|--2014-12-16 (qui contiendra les retours des PES déposés dans « in/2014-12-16 »)



|--facture (qui contiendra les retours des PES déposés dans « in/facture »)

|--TesoT (qui contiendra les retours des PES déposés dans « in/TesoT »)

Note 1: lors du dépôt dans le « WebDAV » d'un fichier sans libellé, un libellé égal à la date du jour est automatiquement créé (sans que le logiciel métier utilise dans ce cas de sous-répertoire pour les libellés). Il n'y a pas dans ce cas de création de sous-répertoire dans le « in » et dans le « out ».

Note 2: une fois déposée sur le « WebDAV », le document est automatiquement consommé par la plateforme sauf si souci réseau ou serveur.

Il est recommandé d'attendre un minimum de 15 minutes afin de contrôler que le document a bien été consommé, si ce n'est pas le cas, c'est qu'il y a eu une erreur lors du dépôt. Il conviendra alors de supprimer le document avant de le redéposer.

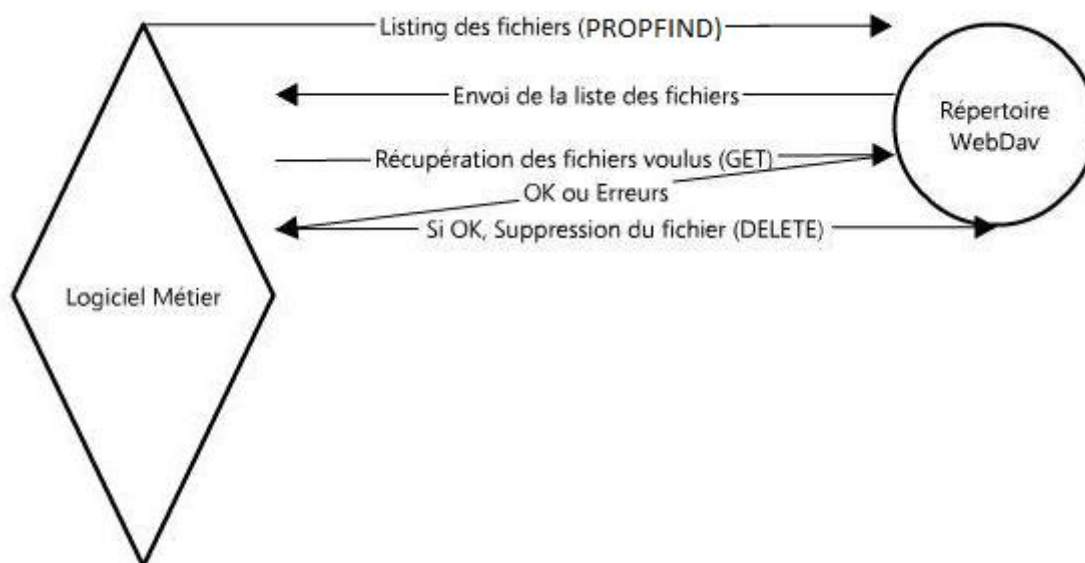
Note 3: le fait que le document soit consommé ne signifie pas que les vérifications se sont bien déroulées (cf. § 1.2.5). L'utilisateur est invité à consulter régulièrement la boîte mail paramétrée pour recevoir les alertes transmises par FAST-Parapheur.

4.3. Retours de la plateforme

Les retours de la plateforme (fichiers signés, acquittements) sont asynchrones. Après traitement, les enveloppes de réponse et les fichiers bureautiques signés sont placés dans le répertoire « WebDAV » « out » de la cinématique et il convient d'interroger la plateforme à intervalle régulier pour vérifier ce qui doit être récupéré.

Les étapes pour la récupération se déroulent comme ceci :

- Afin de vérifier la présence des retours, un listing (via la requête HTTP PROPFIND) du répertoire out et de ses sous-dossiers est nécessaire;
- Réponse de la plateforme avec la liste des fichiers présents ;
- Récupération du ou des retours à récupérer pour traitement par le logiciel métier (via la requête HTTP GET) ;
- Suppression du ou des retours une fois la récupération effective dans le logiciel métier (via la requête HTTP DELETE) ;
- Le schéma ci-dessous reprend la cinématique d'échange exposée précédemment



Note : Il incombe à l'éditeur de surveiller les répertoires partagés mis à sa disposition, et de veiller à les garder « vides » (signe que l'ensemble des fichiers ont été traités / récupérés).

Les sous-répertoires « out » ne contiennent que les retours relatifs aux fichiers déposés dans un sous-répertoire « in ». Un fichier déposé manuellement sur le portail n'aura pas de retour dans le « out ».

4.4. Gestions des erreurs suite au dépôt

Lors du dépôt en échec d'un fichier sur le répertoire « WebDAV », le serveur transmet un mail à l'acteur du dépôt, relatif au problème rencontré.

On trouvera ci-après une liste non exhaustive d'erreurs pouvant se produire et les actions à réaliser pour corriger le problème :

- Tentative de dépôt d'un fichier dont la valeur « NomFic » a déjà été utilisée : le client doit être capable de modifier le « NomFic » d'un document pour le renvoyer par la suite ;
- Tentative de dépôt d'un fichier dont le nom a déjà été utilisée précédemment (Doublon) : le logiciel métier doit être capable modifier le nom du fichier à déposer pour le renvoyer, sans oublier éventuellement de modifier le NomFic ;
- Tentative de dépôt d'une enveloppe avec une signature invalide : le logiciel métier doit être capable de reprendre le fichier, modifier son nom, le re-signer et le renvoyer, sans oublier éventuellement de modifier le NomFic ;
- Tentative de dépôt d'une enveloppe avec un formatage incorrect ou qui ne respecte pas le format PES : le logiciel métier doit être capable recréé l'enveloppe pour qu'elle puisse correspondre au standard attendu et redéposer le fichier.
- Tentative de dépôt d'un fichier qui est vide : le logiciel métier doit vérifier la conformité du fichier transmis ;



- Tentative de dépôt d'un fichier dont l'extension n'est pas autorisée : le logiciel métier doit vérifier la conformité du fichier transmis ;
- Tentative de dépôt d'un fichier PES ne contenant uniquement qu'un entête : le logiciel métier doit vérifier la conformité du fichier transmis ;
- Tentative de dépôt d'un fichier dont le contenu ou le nom d'une PJ est absent : le logiciel métier doit être capable de vérifier la conformité du fichier transmis ;
- Tentative de dépôt d'un fichier pdf protégé en écriture : le logiciel métier doit s'assurer avant envoi que le fichier n'est pas protégé en écriture ;

On trouvera ci-après une liste non exhaustive d'erreurs retournés par le « WebDAV » pouvant se produire à la connexion et les actions à réaliser par le logiciel métier :

- Tentative de connexion avec un certificat périmé
- Tentative de connexion avec un certificat révoqué
- Tentative de connexion avec un certificat émis par une AC non référencée
- Tentative de connexion en cas de coupure internet

Dans chaque cas, le logiciel métier doit afficher un message explicite et permettre une reprise des envois.

4.5. Bonnes pratiques à respecter par l'éditeur (ou le client)

- **Prévoir les mécanismes de reprise nécessaires en cas d'erreur, qu'elles proviennent de la procédure de connexion ou du niveau applicatif, et de remonter les informations nécessaires à l'utilisateur pour le guider dans son diagnostic.**
- **Surveiller les répertoires partagés mis à sa disposition, et de veiller à les garder « vides » (signe que l'ensemble des fichiers ont été traités / récupérés). Par conséquent :**
 - En cas d'erreur lors de la consommation d'un fichier, l'éditeur doit :
 - régler l'anomalie pour que le document puisse être consommé ;
 - supprimer du répertoire les fichiers bloqués, avant de redéposer les fichiers corrigés (et le cas échéant prendre contact avec le support éditeur à l'adresse <https://editeurs.dfast.fr/assistance/> pour remonter l'anomalie si l'éditeur considère que le document aurait dû être consommé)
 - Les différents retours mis à disposition de l'éditeur devront être effacés par l'éditeur, une fois que celui-ci les aura correctement récupérés

Ces deux points sont essentiels pour s'assurer qu'aucun document ne pourrait être « bloqué » pendant leur traitement. Des tests à ce sujet sont prévus dans le cahier de labellisation.

A noter : une purge a lieu automatiquement et quotidiennement pour les fichiers plus vieux de 15 jours dans les répertoires « in » et « out » (et dans les sous-dossiers associés). Les sous-dossiers du répertoire « out » vides sont



supprimés et recréés automatiquement si nécessaire.

- **Avant chaque dépôt de fichier, il faut être sûr qu'un fichier avec le même nom n'existe pas déjà, notamment dans le cas où un fichier n'aurait pas été traité par le Parapheur et que l'on souhaiterait redéposer le fichier.**

Si l'on veut redéposer son fichier sans changer le nom, il faut supprimer le fichier déjà présent, puis le redéposer. Si l'on se contente d'écraser le document (code retour http 204), le « WebDAV » ne considèrera pas qu'il s'agit d'un nouveau fichier et il ne le traitera pas.

- **Eviter absolument de créer un sous-répertoire pour chaque fichier déposé. Ce choix n'a aucun intérêt et complexifie la gestion des données sur FAST-Parapheur. Cela signifie en effet que le sous-répertoire n'est d'aucune utilité et nous recommandons alors de créer un sous-répertoire par jour, ou de ne pas en créer.**
- **Penser à mettre des « NomFic » uniques, selon une règle sûre, qui ne risquent pas d'être utilisés par d'autres éditeurs / collectivités. Nous conseillons d'utiliser le numéro de SIREN, la date et l'heure (avec les millisecondes).**
- **Eviter au possible que le nom du fichier corresponde au « NomFic » du PES.**
- **Les noms de fichiers ne doivent pas excéder les 200 caractères.**
- **Lorsque l'on fait un « PROPFIND » pour voir si un fichier est présent ou pas, il faut que le PROFIND soit fait sur les répertoires connus ET existants. Le « PROPFIND » ne doit pas être fait sur chaque fichier attendu.**
- **La requête « PROPFIND » ne doit être utilisée que toutes les 30 minutes au minimum afin d'éviter une surcharge des serveurs (il est conseillé par ailleurs d'éviter une heure pile afin de répartir les requêtes des différents clients).**

A noter : la plateforme FAST est arrêtée toutes les nuits entre 4h05 et 4h15 pour procéder aux sauvegardes

- **Il est recommandé de garder sur le serveur du client une sauvegarde temporaire des fichiers récupérés, le temps nécessaire pour rejouer éventuellement leur intégration dans le logiciel métier. Cela est important en cas de problème dans la chaîne d'intégration des acquittements qui conduirait à une reprise.**
- **L'encodage du nom des fichiers déposés sur le « WebDav » doit impérativement être en « UTF-8 », c'est important si le nom du fichier contient des caractères accentués ou des caractères spéciaux**

Il est à noter que des informations supplémentaire sont présentes dans le retour « http » qui est transmis par la plateforme lors du dépôt du fichier dans le répertoire « IN » du « Webdav » dont les messages d'erreurs tel que :



Headers dans le retour « http » :

- x-fast-docid : docId si le fichier est consommé par le parapheur (donc sans erreur)
- x-fast-error : code d'erreur (si erreur)
- x-fast-error-msg : message d'erreur plus précis si « x-fast-error » présent

en cas d'erreur, du « json » avec l'erreur se trouve dans le corps de la réponse.

4.6. Exemples de requêtes WebDAV avec cURL

PUT :

```
curl -T "Fichier.xml" -E Certificat.pem:MotDePasse AdresseDuCircuit
```

PROPFIND :

```
curl -E Certificat.pem:MotDePasse -H depth:1 -X PROPFIND AdresseDuCircuit > ServClient/liste.txt
```

GET :

```
curl -E Certificat.pem:MotDePasse AdresseDuCircuit/Retour.xml > ServClient/Retour.xml
```

DELETE :

```
curl -E Certificat.pem:MotDePasse --request DELETE AdresseDuCircuit/Retour.xml
```

Avec :

AdresseDuCircuit : <https://parapheur.dfast.fr/partage/webdav/clients/SIREN/in/circuit>

ServClient : répertoire de stockage des retours sur le serveur du client

La documentation complète est disponible sur le site : <http://curl.haxx.se/>