

Web services REST



Depuis la version 1.11.2 des web services, sous la forme d'API REST, sont disponibles. Esup-Signature propose une documentation automatique disponible sur votre instance à l'adresse "https://<votre adresse>/swagger-ui.html"

La documentation est aussi consultable (mais non testable...) sur le site de démonstration à cette adresse :

<https://esup-signature-demo.univ-rouen.fr/swagger-ui.html>

- [Sécurité des web services](#)
- [Tester les web services](#)
 - [Curl](#)
 - [Swagger](#)
 - [Postman](#)
- [Fonctionnement général](#)
- [Le paramètre stepsJsonString](#)
- [Exemples](#)
 - [Démarrer un formulaire](#)
 - [Envoyer un document dans un circuit existant](#)

Sécurité des web services

L'accès aux web services est filtré par IP. Pour autoriser l'accès il faut modifier la configuration en ajoutant l'IP de la machine concernée au niveau du paramètre `ws-access-authorize-ips` ici :

[Configuration#security.1](#)

Tester les web services

Curl

Les web services d'esup-signature étant au format REST, il est possible de les tester à l'aide de commandes **curl**. Des exemples sont proposés dans cette documentation ainsi que dans la documentation swagger. De plus, il est possible de tester les web services directement depuis l'interface swagger. (Dans esup-signature Admin APIs Doc)

Swagger

Il est possible de tester les web services directement depuis votre interface swagger <https://<adresse esup signature>/swagger-ui.html> . Pour cela il faut être admin et il faut modifier la valeur `supported-submit-methods` dans `application.yml`

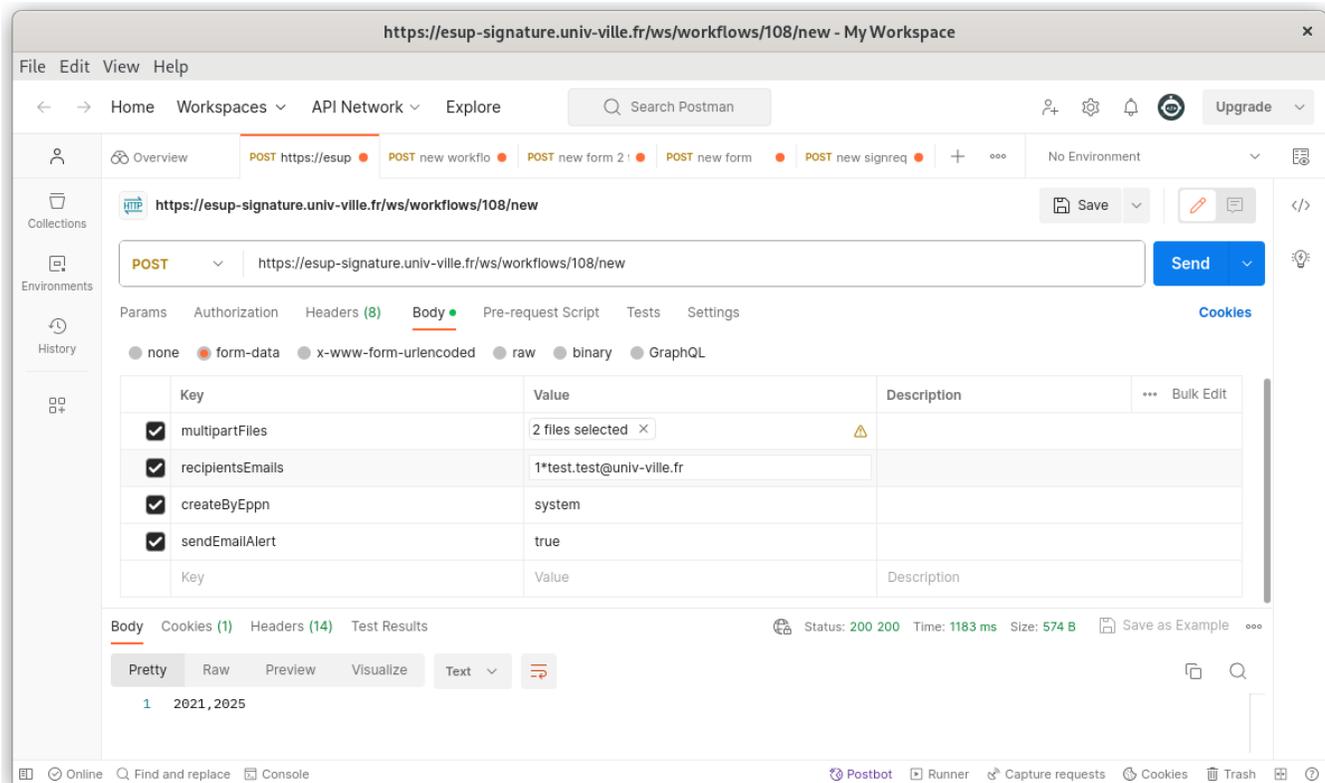
```
supported-submit-methods: ["get", "put", "post"]
```



Les circuits nécessitant l'envoi d'un fichier (multipart files) ne peuvent pas être testés directement via l'interface swagger

Postman

Postman est très pratique pour tester l'envoi de fichier dans un circuit (multiples fichiers acceptés). Ex :



⚠ Dans tous les cas la/les machine(s) qui exécutent les web service (directement, via commandes curl ou qui utilise swagger) doivent être déclarées dans la configuration d'esup-signature. L'accès aux web services permet d'effectuer beaucoup d'actions il est donc sécurisé par adresse IP, à configurer dans src/main/resources/application.yml au niveau du paramètres : **ws-access-authorize-ips**

Fonctionnement général

La première étape pour utiliser les web services est de créer un circuit au niveau de l'administration.

En général, c'est à l'appel du web service que l'on va configurer les signataires, il faudra donc créer autant d'étapes que l'on souhaite en laissant la liste des participants vide (sauf dans le cas où une des étapes possède un signataire fixe).

Voici comment est configurée ce type d'étape :

Étape 1

Description de l'étape:

Type de signature :

L'étape est exécutée automatiquement

Nombre maximum de participants:

Le créateur du circuit pourra modifier les participants de cette étape

Le signataire peut insérer une étape avant la suivante

Le signataire peut apposer plusieurs signatures

Tous les participants doivent signer à cette étape

Alerte PJ PJ obligatoire

Participants

Dans l'onglet paramètres du circuit, vous n'avez pas besoin de configurer d'autorisation (le circuit ne sera pas visible sur la page d'accueil). Sur cette page, vous retrouvez l'id qu'il faudra utiliser au lancement du web service (ici 108) :



Administration / Liste des circuits / Circuit : Agape - Certificats

Paramètres Étapes

WORKFLOW ID : 108

Nombre de demandes pour ce circuit : 61

Titre

Une fois l'appel web service effectué, celui-ci vous retourne les identifiants des demandes (un par document envoyer) séparés par des virgules.

Ces identifiants doivent être conservés par l'application métier afin de permettre un suivi de l'avancement par celle-ci.

Afin de cloturer une demande en fin de circuit, il y a deux possibilité :

1. L'application métier, qui a poussé le document, vient régulièrement contrôler son statut en utilisant le web service `/ws/signrequests/{id}` avec l'id de la demande concernée (le retour peut être pending, completed, refused, deleted ou fully_deleted)
2. Esup-signature appelle un web service fourni par l'application métier. Dans ce cas il faut ajouter une destination au niveau des paramètres du circuit en y mettant l'url du web service de l'application métier. À la fin du circuit, esup-signature enverra un simple GET à votre adresse suivi de : `?signRequestId=<id>&status=<statut>`

Enfin l'application métier pourra supprimer le document en faisant une requête DELETE sur l'adresse `/ws/signrequests/{id}`. Les éléments de vérification de la signature sont conservés dans esup-signature.

Le paramètre stepsJsonString

Depuis la version 1.28 vous pouvez passer une liste de "WorkflowStepDto" (<https://github.com/EsupPortail/esup-signature/blob/master/src/main/java/org/esupportail/esupsignature/dto/WorkflowStepDto.java>), dans tous les web services de création, pour configurer vos étapes. Cela remplace les paramètres recipientEmails, signTypes, allSignToCompletes, etc., mais la rétrocompatibilité est maintenue.

```
[
{
  "title": "string",
  "workflowId": 0,
  "stepNumber": 0,
  "description": "string",
  "recipientsCCEmails": [
    "string"
  ],
  "recipients": [
    {
      "step": 0,
      "email": "string",
      "phone": "string",
      "name": "string",
      "firstName": "string",
      "forceSms": true
    }
  ],
  "changeable": true,
  "signLevel": 0,
  "signType": "hiddenVisa",
  "repeatable": true,
  "repeatableSignType": "hiddenVisa",
  "allSignToComplete": true,
  "userSignFirst": true,
  "multiSign": true,
  "autoSign": true,
  "forceAllSign": true,
  "comment": "string",
  "attachmentRequire": true,
  "maxRecipients": 0
}
]
```

Exemples

Démarrer un formulaire

Accès : <https://<votre adresse>/ws/forms/{id}/new>

Description : Ce web service va créer une nouvelle instance du formulaire désigné pas le paramètre "id" de l'url d'accès.

Attributs :

Attribut	Description
createByEppn	eppn du propriétaire du futur document
stepsJsonString	Si les participants de certaines étapes sont configurables, il faut saisir un tableau de WorkflowStepDto[]. Ex : [{"signType": "pdfImageStamp", "recipients": [{"email": "test.test@univ-ville.fr"}, {"email": "test2.test2@univ-ville.fr"}]}, {"signType": "certSign", "recipients": [{"email": "test.test@univ-ville.fr"}, {"email": "test2.test2@univ-ville.fr"}]}]
targetEmails	Pour que la demande soit transmise par à la fin du circuit, il est possible, d'envoyer un tableau de String contenant la liste des destinataires finaux
targetUrl	url pour la destination finale des formulaire terminés. Ex : smb://stockage.univ-ville.fr/form

Exemple de commande curl :

```
curl --location --request POST 'http://dsi-7.univ-rouen.fr/ws/signrequests/new' \
--form 'multipartFiles=@"/home/lemaida3/Documents/sample.pdf"' \
--form 'createByEppn="test@univ-ville.fr"' \
--form 'stepsJsonString="{\"signType\": \"pdfImageStamp\", \"recipients\": [{\"email\": \"david.lemaignent@univ-rouen.fr\"}, {\"email\": \"demo.esup@inv.univ-rouen.fr\"}], {\"recipients\": [{\"email\": \"david.lemaignent@univ-rouen.fr\"}, {\"email\": \"demo.esup@inv.univ-rouen.fr\"}]}"'
```

Envoyer un document dans un circuit existant

Accès : <https://<votre adresse>/ws/workwlos/{id}/new>

Description : Ce web service va déposer d'un document dans une nouvelle instance d'un circuit

Attributs :

Attribut	Description
multipartFile (obligatoire)	Multipart stream du fichier à signer
createByEppn (obligatoire)	eppn du propriétaire du futur document
stepsJsonString	Si les participants de certaines étapes sont configurables, il faut saisir un tableau de WorkflowStepDto[]. Ex : [{"signType": "pdfImageStamp", "recipients": [{"email": "test.test@univ-ville.fr"}, {"email": "test2.test2@univ-ville.fr"}]}, {"signType": "certSign", "recipients": [{"email": "test.test@univ-ville.fr"}, {"email": "test2.test2@univ-ville.fr"}]}]
targetEmails	Pour que la demande soit transmise par à la fin du circuit, il est possible, d'envoyer un tableau de String contenant la liste des destinataires finaux
targetUrl	url pour la destination finale des formulaire terminés. Ex : smb://stockage.univ-ville.fr/form

Exemple de commande curl :

```
curl --location --request POST 'https://esup-signature.univ-rouen.fr/ws/workflows/999999/new' \
--form 'createByEppn="test@univ-rouen.fr"' \
--form 'title="TITRE"' \
--form 'multipartFiles=@"/home/lemaida3/Documents/esup-signature/sample.pdf"' \
--form 'recipientsJsonString="{\"signType\": \"pdfImageStamp\", \"recipients\": [{\"email\": \"test.test@univ-ville.fr\"}, {\"email\": \"test2.test2@univ-ville.fr\"}], {\"signType\": \"certSign\", \"recipients\": [{\"email\": \"test.test@univ-ville.fr\"}, {\"email\": \"test2.test2@univ-ville.fr\"}]}"';type=application/json'
```