# API

Usage	Méthode	Chemin
pousser un document	POST	/api/documents/push/
liste des documents récents	GET	/api/documents/recently-added/
valider le hash d'un document	POST	/api/validation/ <document-type>/</document-type>
lister les validations de documents	GET	/api/validation/ <document-type>/?user_na meid=YYY</document-type>
autoriser un jeton OAuth2 de récupération de document	GET	/api/oauth2/get-document/authorize
obtenir un jeton OAuth2 de récupération de document	GET	/api/oauth2/get-document/token
récupérer un document avec un jeton OAuth2	GET	/api/oauth2/get-document/
pousser un document temporaire via OAuth2	POST	/api/oauth2/put-document/
faire accepter un document temporaire via OAuth2	GET	/api/oauth2/put-document/ <document-pk> /authorize/</document-pk>

## Pousser un document

## Requête

```
POST /api/documents/push/ HTTP/1.1
Content-Type: application/json
Content-Size: xxx

{
    "user_nameid": "YYYY",
    "origin": "mairie de yyy - service enfance",
    "file_b64_content": "contenu du fichier en base64",
    "file_name": "nom du fichier",
    "deletable_by_user": true,
}
```

Champ du contenu	Usage
user_nameid	l'identifiant Publik de l'utilisateur
origin	l'origine du fichier, ex.: le nom d'une entité administrative
file_b64_content	le contenu du fichier sous forme de chaîne encodée en base64
file_name	le nom du fichier
deletable_by_user	booléen, est-ce que le fichier peut-être supprimer par l'utilisateur ou n'est pas sous son contrôle

Si l'origine indiquée n'existe pas encore, la valeur sera slugifiée et utilisée pour la créer implicitement.

### Réponse

```
{
    "result": 1,
    "data": {
        "id": null
```

20 avril 2024 1/4

### Liste des documents récents

```
GET /api/documents/recently-added/ HTTP/1.1
```

Cette API renvoit au plus les dix documents les plus récents parmi ceux crées il y a moins de quatorze jours. Cette API est la seule à récupérer l'utilisateur depuis la requête (via signature Publik).

## Validation des documents

Les types de document validables sont à définir dans les settings de fargo, via la variable FARGO\_DOCUMENT\_TYPES qui est une liste de dictionnaire, ex.:

```
FARGO_DOCUMENT_TYPES = [
        'name': 'avis-d-imposition',
        'label': u'Avis d\'imposition',
        'metadata': [
            {
                 'label': u'Personne-s concernée-s',
                 'varname': 'personnes_concernees',
                 'type': 'string',
            },
            {
                 'label': u'Année',
                 'varname': 'annee',
                 'type': 'string',
            },
            {
                 'label': u'Revenu fiscal de référence',
                 'varname': 'revenu_fiscal_de_reference',
                'type': 'string',
            },
       ],
   },
```

Les champs de ce dictionnaire sont :

- name : le nom du type de document à utiliser dans les différentes API
- label : le lable qui sera utilisé dans les interfaces pour afficher ce type de document
- metadata : une liste de dictionnaires décrivant les métadonnées possibles, dont les champs sont :
  - o varname : le nom de cette métadonnée
  - o @required : optionnel, par défaut True, définit si l'attribut est requis
  - o label : l'intitulé qui sera utilisé pour afficher cette métadonnée,
  - o type : le type de la métadonnée, pour l'instant seul string est possible
  - o validation : une expression régulière de validation de la valeur (optionnel)

### Valider le hash d'un document

#### Requête

20 avril 2024 2/4

```
POST /api/validation/avis-d-imposition/ HTTP/1.1
Content-Type: application/json
Content-Size: xxx

{
    "user_nameid": "YYY",
    "origin": "slug d'une origine déclarée dans l'admin",
    "content_hash": "hash shal du document validé",
    "start": "2019-01-01",
    "end": "2019-12-31",
    "creator": "Jean Darmette",
    "created": "2019-01-02",
    "personnes_concernees": "John Doe, Jane Doe",
    "annee": "2016",
    "revenu_fiscal_de_reference": "24678",
}
```

Ici l'origine ne sera pas créer automatiquement (contrairement à l'API de push), un slug doit être utilisé.

Champ	Usage
user_nameid	identifiant Publik de l'utilisateur
origin	slug de l'origine
content_hash	hash SHA-1 du document validé, ex.: 3f786850e387550fdab836ed7e6dc881de23001b
start	date au format ISO-8601 de début de validité de la validation du document
end	date au format ISO-8601 de fin de validité de la validation du document
creator	agent/service ayant créé la validation
created	date au format ISO-8601 de création de la validation du document

Les champs suivants correspondent aux champs de métadonnées définis pour le document, ils doivent valider les

#### Réponse

```
"result": 1
"data": {
    "origin": "mairie-de-xxx-service-enfance",
    "url": "https://fargo.<domaine>.fr/api/validation/avis-d-imposition/10/",
    "id": 10,
    "content_hash": "hash shal du document validé",
    "start": "2019-01-01",
    "end": "2019-12-31",
    "creator": "Jean Darmette",
    "created": "2019-01-02",
    "personnes_concernees": "John Doe, Jane Doe",
    "annee": "2016",
    "revenu_fiscal_de_reference": "24678"
}
```

### Lister les validations de document

#### Requête

```
GET /api/validation/avis-d-imposition/?user_nameid=YYY Accept: application/json
```

#### Réponse

20 avril 2024 3/4

# Évolutions

L'API est baroque comparée à nos autres APIs, il faudrait :

- ajouter le NameID dans l'URL query-string plutôt qu'en POST
- définir creator depuis le NameID dans la query-string si disponible
- rendre Validation.created automatiquement défini à la création
- séparer les métadonnées dans un sous-champ metadata pour éviter les collisions
- utiliser "err": 0 plutôt que @"result": 1"

20 avril 2024 4/4