

Interface parapheur v1.825

Table des matières

1	Objectif	2
2	Démarrage du workflow	3
2.1	Authentification et identification de l'utilisateur	3
2.2	Dépôt automatisé avec un déclenchement manuel du workflow	3
2.3	Dépôt automatisé avec un workflow prédéfini	5
2.4	Dépôt automatisé avec des métadonnées	5
2.5	Ajout de pièces complémentaires au document principal	6
3	Fin de circuit	7
3.1	Notification de l'application métier	7
3.2	Tester le statut du workflow	7
3.3	Récupération des informations fournies par les acteurs du workflow	8
3.4	Téléchargement du document signé	9
4	Autres API disponibles	10
4.1	Liste	10

1 Objectif

L'objectif de ce document est de fournir les appels de webservices pour piloter le parapheur à chaque étape du workflow ; de l'initialisation jusqu'à la fin du circuit. Ces appels de webservice permettent de déposer un document sur le parapheur et de déclencher automatiquement ou manuellement le circuit de visa ou de signature approprié au type de document en fonction des informations disponibles et de l'organisation dans les services.

Ce document fourni les principaux appels de webservices nécessaires pour ;

- Déposer le document à viser
- Déclencher le circuit de visa avec ou sans signature électronique
- Vérifier l'état d'avancement du workflow
- Télécharger le document signé
- Gérer pièces complémentaires au document principal à signer ou viser
- Gérer des métadonnées pour le routage
- Echanges d'annotations ou de métadonnées entre le parapheur et l'application métier
- Archiver le workflow si nécessaire

2 Démarrage du workflow

2.1 Authentification et identification du rôle de l'utilisateur

Les appels d'API JSON se font à l'adresse suivante : <https://<host><:port>/cdmsserveur/startElasticXML>

Les champs ci-dessous doivent être renseignés pour identifier l'utilisateur et le propriétaire des documents associés déposés ou recherchés ;

Paramètre	Description	Obligatoire	Valeur
nomBase	Nom de l'environnement	Obligatoire	
User	Login de connexion de l'utilisateur	Obligatoire	
encryptedPassword	Mot de passe crypté de l'utilisateur de connexion	Obligatoire	(user + password + "MailObserver1").hashCode()+""
Mailid	Adresse mail de l'utilisateur	Obligatoire	
secuUsers	Utilisateur pour les recherches	Facultatif	
mailOwner	Adresse mail du propriétaire	Obligatoire si domaineOwner non renseigné	
domaineOwner	Organisation du propriétaire du document	Obligatoire si mailOwner non renseigné	

2.2 Dépôt automatisé avec un déclenchement manuel du workflow

Dans certains cas difficilement automatisables, l'intervention d'un agent pour sélectionner les acteurs, les étapes et le circuit approprié est nécessaire. Dans ce cas, il faut utiliser un appel d'API sur le modèle suivant pour automatiser à minima le dépôt du document principal à viser ;

- Exemple de syntaxe JSON pour l'Api put() à alimenter sur une seule ligne pour le dépôt sans renseigner les informations concernant les workflows

```
{
  "elasticDocName": "devis2018_28.pdf",
  "fileContent": "iVB+GgCaVNoSv8CK0lkjkkjjkjjXU...FTkSuQmCC",
  "elasticTaille": "702",
  "elasticContentType": "application/pdf",
  "getasbase64": true,
  "elasticCommand": "put()",
  "elasticType": "documents",
  "elasticDocType": "documents",
  "secuUsers": "DOMINIQUE",
  "mailid": "dd@gmail.com",
  "user": "DOMINIQUE",
  "encryptedPassword": "1703850244",
  "nomBase": "NEOGED",
  "mailOwner": "dd@gmail.com"
}
```

Champs obligatoires à renseigner ;

- elasticDocName:"devis2018_28.pdf" = nom unique du fichier
- fileContent = fichier encodé en base64
- elasticTaille= taille du fichier en octets
- elasticContentType=Mime type du fichier
- elasticDocType= « documents » par défaut à mettre en cohérence avec les catégories personnalisées

La réponse retourne l'identifiant unique du document, s'il n'y pas d'erreur;

<pre>1 { 2 "data": [], 3 "messageErreur": "", 4 "last": false, 5 "codeRetour": "OK", 6 "maxSize": 0, 7 "documentID": "AWlhkPpJRXSN3W2R3RB1", 8 "nonStoredList": "" 9 }</pre>	<p>Identifiant présent dans le champ</p> <p>« documentID »</p>
--	--

La réponse JSON retourne le message suivant en présence d'un doublon non autorisé et d'un document déjà présent avec le même nom dans un workflow de validation

<pre>1 { 2 "data": [], 3 "messageErreur": "Le document BDC2018-37.png est verrouillé", 4 "last": false, 5 "codeRetour": "OK", 6 "maxSize": 0, 7 "documentID": "AWLxMTa_RXSN3W2R3RE0", 8 "nonStoredList": "" 9 }</pre>	
---	--

2.3 Dépôt automatisé avec un workflow prédéfini

	<p>Dans le cas simple d'un circuit générique et avec une liste connue de visas ou de signature</p> <p>Pour déclencher un circuit précis en même temps que le dépôt du document sur le parapheur, il faut ajouter les informations JSON suivantes ;</p> <p>"startWorkflow":true, "workflowid":1</p> <p>Sans indiquer d'identifiant de workflow, le routage peut être effectué automatiquement à partir des métadonnées fournies.</p>
---	--

- Exemple pour le déclenchement d'un workflow précis à chaque dépôt ;

```
{
  "elasticDocName":"devis2018_28.pdf",
  "startWorkflow":true,
  "workflowid":1,
  "fileContent":"iVB+GgCaVNoSv8CK0lklkjjlkkjjjXU...FTkSuQmCC",
  "elasticTaille":"702",
  "elasticContentType":"application/pdf",
  "getasbase64":true,
  "elasticCommand":"put()",
  "elasticType":"documents",
  "elasticDocType":"documents",
  "secuUsers":"DOMINIQUE",
  "mailid":"dd@gmail.com",
  "user":"DOMINIQUE",
  "encryptedPassword":"1703850244",
  "nomBase":"NEOGE",
  "mailOwner":"dd@gmail.com"
}
```

2.4 Dépôt automatisé avec des métadonnées

Un circuit peut se déclencher automatiquement au dépôt d'un document si ses métadonnées respectent le filtre métadonnées du circuit

<p>Nom du processus : <input type="text" value="circuit 1"/></p> <p>Type de workflow : <input type="text" value="Factures"/></p> <p>Catégorie : <input type="text" value="Facture"/></p> <p>Filtre métadonnées : <input type="text" value="Montant TTC<5000;"/></p> <p>Délai d'alerte : <input type="text" value="5"/></p> <p>Archivage auto <input type="checkbox"/></p> <p>Démarrage automatique <input checked="" type="checkbox"/></p> <p>Adresse url d'appel du Webservice : <input type="text"/></p>	<p>Description du processus : <input type="text" value="Circuit 1"/></p> <p>Type : <input type="text" value="Demabox"/></p> <p><input type="checkbox"/> Fiche navette</p> <p>Métadonnées à afficher : <input type="text"/></p> <p>Délai de 2ème relance : <input type="text"/></p> <p>Alerte sur délai <input type="checkbox"/></p> <p>Relance globale <input type="checkbox"/></p>
<p>Syntaxe à ajouter au champ JSON lors du dépôt du document principal pour déclencher</p> <p>"startWorkflow":true,"elasticDocComment" : "<Type=BDC><Catégorie=Autre><Montant TTC=8000>"</p>	

- Exemple de dépôt avec des métadonnées ;

```
{ "elasticDocName": "Facture_2018_40.pdf", "elasticCommand": "put()", "elasticDocType": "Demabox",
  "elasticType": "documents", "startWorkflow": true, "elasticDocComment":
  "<Type=Demabox><Catégorie=Facture><Montant
  TTC=122200>", "getasbase64": true, "fileContent": "iVBORw0KGgoAAAANSU...AAAAAAGAh+AecMIJ4J9
  MGEAAAAAEIFTkSuQmCC", "elasticTaille": "702", "elasticContentType": "application/pdf", "secuUsers": "D
  OMINIQUE", "mailid": "dd@gmail.com", "user": "DOMINIQUE", "encryptedPassword": "1703850244", "nom
  Base": "NEOGED", "mailOwner": "dd@gmail.com" }
```

2.5 Ajout de pièces complémentaires au document principal

Le document principal à signer ou à viser n'a pas toujours toutes les informations nécessaires pour la validation des différents acteurs du workflow. Il est possible d'associer des annexes pour compléter ce dernier et permettre la validation par les différents acteurs à toutes les étapes de validation.

- Informations nécessaire pour l'Api « addPj() » pour ajouter les pièces complémentaires
- "mainId" : Identifiant du document principal déposé pour validation
- "elasticDocName" : Nom du document à associer
- "elasticId" : Identifiant du document à associer
- "contentType", "elasticType", "elasticDocType" du document à associer

- Exemple d'appel d'Api addPj()

```
{ "elasticCommand": "addPj()", "mainId": "AWlwnTyGRX5N3W2R3REI",
  "elasticDocName": "Balance_format_pdf(1).pdf", "elasticId": "AWINPqGfPhAIf5SSze7n", "contentType": "a
  pplication/pdf", "elasticType": "documents",
  "elasticDocType": "document", "secuUsers": "DOMINIQUE", "mailid": "dd@gmail.com", "user": "DOMINIQUE",
  "encryptedPassword": "1703850244", "nomBase": "NEOGED", "mailOwner": "dd@gmail.com" }
```



3 Fin de circuit

3.1 Notification de l'application métier

Comme indiqué ci-dessous, le parapeur peut appeler une adresse URL quand le circuit est terminé pour notifier l'application métier cible avec l'identifiant du document validé ou refusé en paramètre ; 2 paramètres `elasticid` et `status`

3.2 Tester le statut du workflow

Le parapeur permet de connaître à tout moment l'état d'avancement de son circuit de workflow;

La recherche peut être effectuée à partir de l'identifiant unique du document fournit pendant le dépôt ;

La recherche peut être également effectuée à partir du nom du document ;

```
{
  "elasticCommand": "search(_id=AWlN3W2R3RAX)",
  "elasticFields": "elasticid,docsnom,workflowstatus",
  "elasticRequest": "",
  "elasticType": "documents",
  "mainDomain": "*",
  "mailid": "dd@gmail.com",
  "user": "DOMINIQUE",
  "encryptedPassword": "1703850244",
  "nomBase": "NEOGED",
  "mailOwner": "dd@mail.com"
}
```

Les principaux statuts à connaître

« Workflowstatus »	Document
9	Refusé
10	Archivé
11	Fin de circuit, à archiver
12	En attente de visa
13	En attente de signature

Date	Nom	Statut	Nom du processus	Commentaire
18/01/2019 16:40	TELINO - Devis - Test.pdf	A traiter	exemple mantes	[Soumis]
18/01/2019 15:35	TELINO - Devis - Copie.pdf	A traiter	exemple mantes	[Soumis]
18/01/2019 15:05	TELINO - Devis.pdf	A archiver	exemple mantes	[Terminé]
24/12/2018 11:46	Composants SMARTGED.docx	En cours	CS -> DAF -> DGS-> M	[Soumis] coi
13/12/2018 09:42	planning.xlsx V2	Refusé	CS -> DAF -> DGS-> M	[Refusé] Pas
11/12/2018 14:52	Capture d'écran 2018-12-11 à ...	En cours	CS -> DAF -> DGS-> M	[Soumis]
30/11/2018 10:19	detail paiement.xlsx	En cours	CS -> DAF -> DGS-> M	[Validé] J'ins
30/11/2018 10:13	plan d'action smart collab.xlsx	Refusé	CS -> DAF -> DGS-> M	[Refusé] No

3.3 Récupération des informations fournies par les acteurs du workflow

Des notes et des métadonnées peuvent être renseignées pendant les validations ou les rejets. Ces informations peuvent être indispensables et seulement connues par certains acteurs du workflow (par exemple la date réelle d'une livraison) et ces acteurs n'ont pas toujours accès à toutes les applications métier concernées.

- « comment » = liste des métadonnées du document
- « workflowlogcomment » : Annotations visa ou rejet
- Exemple d'Api pour récupérer les métadonnées

```
{
  "elasticCommand": "search(_id=AWIteVbGRX5N3W2R3REc)",
  "elasticFields": "elasticid,comment",
  "elasticRequest": "",
  "elasticType": "documents",
  "mainDomain": "*",
  "mailid": "dd@gmail.com",
  "user": "DOMINIQUE",
  "encryptedPassword": "1703850244",
  "nomBase": "NEOGED",
  "mailOwner": "dd@gmail.com"
}
```

- Réponse JSON

```
1 1
2  "data": [ [
3     "AWk1N-PmRX5N3W2R3RAM",
4     "<Type=acte><Catégorie=Autre><Montant TTC=8000>"
5   ] ],
6  "messageErreur": "",
7  "last": true,
8  "codeRetour": "OK",
9  "maxSize": 1,
10 "documentID": "",
11 "nonStoredList": ""
```

3.4 Téléchargement du document signé

A l'étape de signature, en fonction du type de document signé, la signature peut être ;

- « interne » et présente dans une nouvelle version du document de type « signature » (pour un fichier PDF, flux PESv2 ...)

- « externe » dans un fichier xml supplémentaire de type Signature

Nom	Type	Catégorie	Hiérarchie	Version	Taille
BDC2018-40.png	Demabox	Facture		V 1	702 o
BDC2018-40.png.xml	Signature			V 1	18 Ko

Le document signé ou de signature a un identifiant différent du document principal.

- Exemple de récupération de l'identifiant du fichier de signature à partir de l'identifiant du document principal déposé pour validation

```
{
  "elasticCommand": "search(_id=AW1xltM6RX5N3W2R3RFA)",
  "elasticFields": "elasticid,documents_list",
  "elasticRequest": "",
  "elasticType": "documents",
  "mainDomain": "*",
  "additionalCriteria": "workflowflag=true",
  "mailid": "dd@gmail.com",
  "secuUsers": "DOMINIQUE",
  "user": "DOMINIQUE",
  "encryptedPassword": "1703850244",
  "nomBase": "NEOGED",
  "mailOwner": "dd@gmail.com"
}
```

- Exemple de réponse JSON por récupérer l'identifiant de la signature



```
{
  "data": [
    [
      "AW1xltM6RX5N3W2R3RFA",
      "AW13RgK8dpyTMG3lp23Y;BDC2018-40.png.xml;application/xml"
    ]
  ],
  "messageErreur": "",
  "last": true,
  "codeRetour": "OK",
  "maxSize": 1,
  "documentID": "",
  "nonStoredList": ""
}
```

Le document signé, comme les autres documents autorisés, peut ainsi être téléchargé avec l'Api getDoc et l'identifiant unique du document.

- Exemple d'API de téléchargement d'un document à partir de son identifiant (en base64)

```
{
  "elasticCommand": "getDoc(_id= AW13RgK8dpyTMG3lp23Y)",
  "elasticType": "documents",
  "getasbase64": true,
  "mailid": "dd@gmail.com",
  "secuUsers": "DOMINIQUE",
  "user": "DOMINIQUE",
  "encryptedPassword": "1703850244",
  "nomBase": "NEOGED",
  "mailOwner": "dd@gmail.com"
}
```

- Réponse avec un fichier encodé en base64

```

1 {
2   "data": [[ [
3     "PD94bWwgdmVyc2lvcj0iMS4wIiBlbmNvZGluc20iVVRGLTgiPz4KPGRzOl:
4     "application/xml"
5   ]]],
6   "messageErreur": "",
7   "last": false,
8   "codeRetour": "OK",
9   "maxSize": 0,
10  "documentID": "",
11  "nonStoredList": ""
12 }

```

- Réponse si le document n'est pas trouvé ou s'il n'est plus accessible

```

1 {
2   "data": [[]],
3   "messageErreur": "",
4   "last": true,
5   "codeRetour": "OK",
6   "maxSize": 0,
7   "documentID": "",
8   "nonStoredList": ""
9 }

```

4 Autres API disponibles

4.1 Liste

"elasticCommand"	
startWorkflow()	Démarrage d'un workflow sur un document à partir de son identifiant
ArchiveWorkflow()	Demander l'archivage d'un workflow terminé à partir de son identifiant
RejectWorkflow()	Refuser l'étape en cours avec un motif à partir de son identifiant

fin