

WebServices **AIRS Dossier**

Digitech

Version 2.1, 08/08/2024

Sommaire

1. Introduction	6
2. REST	7
2.1. Document	7
2.1.1. Client	7
2.1.2. Authentification	7
2.1.3. Méthodes	7
2.1.3.1. POST /document	7
2.1.3.2. GET /document/{docId}	9
2.1.3.3. PATCH /document/{docId}	10
2.1.3.4. DELETE /document/{docId}	12
2.1.3.5. DELETE /document/remove/{docId}	13
2.2. AuthorityList	13
2.2.1. Client	13
2.2.2. Authentification	13
2.2.3. Méthodes	14
2.2.3.1. GET /authorityList	14
2.2.3.2. GET /authorityList/{fieldCode}	14
2.3. Baskets	15
2.3.1. Client	16
2.3.2. Authentification	16
2.3.3. Méthodes	16
2.3.3.1. GET /baskets	16
2.3.3.2. GET /baskets/orgId/{orgId}	17
2.3.3.3. GET /baskets/flowCode/{flowCode}	17
2.3.3.4. GET /baskets/docCount/{basketName}	18
2.3.3.5. GET /baskets/docLink/{basketName}	19
2.4. Search	20
2.4.1. Client	20
2.4.2. Authentification	20
2.4.3. Méthodes	21
2.4.3.1. GET /search/docCount	21
2.4.3.2. GET /search/docLink	21
2.4.3.3. GET /search/results	23
2.5. Attachment	25
2.5.1. Client	26
2.5.2. Authentification	26
2.5.3. Méthodes	26
2.5.3.1. POST /attachment/{docId}	26

2.5.3.2.	GET /attachment/{docId}	27
2.5.3.3.	GET /attachment/{docId}/{attachmentId}	28
2.5.3.4.	DELETE /attachment/{docId}	29
2.5.3.5.	GET /attachment/download/{attachmentId}	30
2.6.	DossierContext	31
2.6.1.	Client	31
2.6.2.	Authentification	31
2.6.3.	Méthodes	31
2.6.3.1.	GET /dossierContext/flows	31
2.6.3.2.	GET /dossierContext/contentType/{ctyCode}	32
2.6.3.3.	GET /dossierContext/mandatoryFields/{ctyCode}	33
2.7.	ExtendDocument	34
2.7.1.	Client	34
2.7.2.	Authentification	34
2.7.3.	Méthodes	34
2.7.3.1.	DELETE /extendDocument/deleteDocuments	34
2.7.3.2.	DELETE /deleteDocumentsType/{deleteType}	36
2.7.3.3.	GET /extendDocument/{docId}	37
2.7.3.4.	POST /extendDocument	38
2.7.3.5.	PATCH /extendDocument	40
2.8.	ExtendSearch	42
2.8.1.	Client	42
2.8.2.	Authentification	42
2.8.3.	Méthodes	43
2.8.3.1.	GET search4FieldsAndAttachments	43
2.9.	Workbook	45
2.9.1.	Client	45
2.9.2.	Authentification	45
2.9.3.	Méthodes	45
2.9.3.1.	POST /workbook	45
2.9.3.2.	GET /workbook/{id}	46
2.9.3.3.	GET /workbook	47
2.9.3.4.	PATCH /workbook	48
2.9.3.5.	DELETE /workbook/{id}	49
2.9.3.6.	PATCH /workbook/{id}/add	50
2.9.3.7.	PATCH /workbook/{id}/remove	51
3.	SOAP	52
3.1.	Document	52
3.1.1.	Client	52
3.1.2.	Méthodes	52
3.2.	AuthorityList	59

3.2.1. Client	59
3.2.2. Méthodes	59
3.2.3. Authentification	59
3.3. Baskets	61
3.3.1. Client	61
3.3.2. Méthodes	61
3.4. Search	72
3.4.1. Client	72
3.4.2. Méthodes	72
3.5. Attachment	78
3.5.1. Client	78
3.5.2. Méthodes	78
3.6. DossierContext	83
3.6.1. Client	84
3.6.2. Méthodes	84
3.7. ExtendDocument	86
3.7.1. Client	87
3.7.2. Méthodes	87
3.8. ExtendSearch	94
3.8.1. Client	95
3.8.2. Authentification	95
3.8.3. Méthodes	95
4. Annexes	98
4.1. modèles globaux	98
4.2. Document modèles	98
4.3. Authority List modèles	99
4.4. Search modèles	99
4.5. DossierContext modèles	100
4.6. ExtendDocument modèles	101
4.7. Attachment modèles	105
4.8. ExtendedSearch modèles	106
4.9. Workbook modèles	107

Table 1. Notes de suivi

Date	Version	Etat	Objet	Rédacteur	Valideur
22/06/2020	1.0	Validé	Création du document	NFE	
12/05/2021	1.1	Validé	Modification du document, ajout compatibilité 7.1	NFE	
14/09/2022	2.0	Validé	Ajout WS v8.0	NFE / JMU	NFE
08/08/2024	2.1	Validé	Ajout WSAttachment# download	NFE	

1. Introduction

AIRS Dossier propose des **Web Services** de type **soap** (depuis sa version 4) et **rest** (à partir de la version 7.1) permettant à d'autres applications ou à des intégrateurs de s'interfacer avec l'application AIRS Dossier.

La liste des **Web Services** disponibles est accessible à l'adresse

REST

```
http(s)://<hostname>:<port>/<context>/rest/services
```

SOAP

```
http(s)://<hostname>:<port>/<context>/services
```



La plupart des **Web Services** sont également accessibles en **REST**.
Si disponibles, il est conseillé d'utiliser ces **APIs**

2. REST



Définitions openapi

- La définition **openapi** peut être récupérée à l'adresse <http://<hostname>:<port>/<context>/rest/api-docs>
- une interface **swagger** (permettant également de tester les [Web Services](#)) est disponible à l'adresse

```
http://<hostname>:<port>/<context>/rest/api-docs
```



Type de paramètres

Les paramètres transmis aux appels *http* des web services seront d'un des types suivants :

query

Paramètre ajouté à l'url (sous forme `&*<Code>=<Valeur>*`)

path

Paramètre ajouté à la construction de l'url

body

Paramètre ajouté en tant que *body*

multipart

Multiple données envoyées, de différents types, en un seul appel [http](#).

2.1. Document



Ce [Web Service](#) n'est disponible qu'à partir de la version 8 de [AIRS Dossier](#)

2.1.1. Client

La classe `com.digitech.dossier.rest.ws.IDocumentClient` de la librairie `com.digitech.airs:dossier-rest-client:1.2.1` abstrait l'utilisation de cette **API**.

2.1.2. Authentification

Ce [Web Service](#) requiert une **authentification basique**.

2.1.3. Méthodes

2.1.3.1. **POST** /document

Cette méthode permet de créer un document

Table 2. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
documentData	multipart	Ensemble de données du document	Oui	<pre>{ "contentType": "CR", "secretLevel": 0, "fields": [{ "code": "CR", "field": "Title"}], "data": ["CR"], "filename": "test", "flowCode": "CR", "scriptCode": "test" }</pre>
fileData	multipart	Liste de fichiers	Non	

cf le chapitre [Section 4.2, “Document modèles”](#) pour une description de la structure documentData (Se référer à WSCCommonDocumentType).



Le type de contenu du corps de la requête doit être "multipart/form-data". Pour ajouter plusieurs fichiers au document, il suffit d'envoyer plusieurs fois le paramètre fileData dans le corps de la requête.

appel curl

```
curl --location --request POST '<host>:<port>/<context>/rest/document' \
--header 'Authorization: Basic Yi5iaWxsb246cGFzc3dvcmQ=' \
--form 'documentData="{
  \"contentType\": \"CR\",
  \"fields\": [
    {
      \"code\": \"D_MODIF\",
      \"field\": \"04/02/2021 16:58:35\"
    },
    {
      \"code\": \"CR_DES\",
      \"field\": \"Test DOC\"
    },
    {
      \"code\": \"CR_REDACTEUR\",
      \"field\": \"1\"
    },
    {
      \"code\": \"CR_DATE\",
      \"field\": \"01/01/2022 0:00:00\"
    },
    {
      \"code\": \"CR_RESUME\",
      \"field\": \"TEST DOC\"
    }
  ]
}"'
```



```

    }
  ]
};type=application/json' \
--form 'fileData=@"/C:/TEST1.pdf"' \
--form 'fileData=@"/C:/TEST2.pdf"'

```

appel requête

```
http://<host>:<port>/<context>/rest/document
```

Table 3. Réponses

Code Http	Description
200	Opération réussie (cf. Retour, Identifiant du document)
401	Authentification non fournie
500	Erreur interne au serveur/service

Retour, Identifiant du document

```
6733
```

2.1.3.2. [GET](#) /document/{docId}

Cette méthode permet de récupérer les métadonnées d'un document à partir de son identifiant

Table 4. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
docId	path	Identifiant du document	Oui	4959

appel curl

```

curl -X 'GET' \
  '<host>:<port>/<context>/rest/document/4959' \
  -H 'accept: application/json'

```

appel requête

```
http://<host>:<port>/<context>/rest/document/4959
```

Table 5. Réponses

Code Http	Description
200	Opération réussie (cf. Structure de retour)
401	Authentification non fournie

Code Http	Description
404	Document non trouvé
500	Erreur interne au serveur/service

Structure de retour

```
{
  "WSDocumentType": {
    "contentType": "CR",
    "secretLevel": 10,
    "fields": [
      {
        "code": "D_MODIF",
        "field": "12/07/2022 15:32:47"
      },
      {
        "code": "CR_THEME",
        "field": "Réunion Transverse"
      },
      {
        "code": "D_CREAT",
        "field": "24/08/2021 11:02:30"
      },
      {
        "code": "CR_DES",
        "field": "test html"
      },
      {
        "code": "CR_REDACTEUR",
        "field": "ADM Dossier"
      },
      {
        "code": "CR_DATE",
        "field": "24/08/2021 0:00:00"
      },
      {
        "code": "CR_RESUME",
        "field": "test html"
      }
    ]
  }
}
```

cf le chapitre [Section 4.2, “Document modèles”](#) pour une description de la structure de retour.

2.1.3.3. **PATCH**/document/{docId}

Cette méthode permet de mettre à jour les métadonnées d’un document

Table 6. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
docId	path	Identifiant document	du Oui	4959
documentData	multipart	Ensemble de données du document	Oui	<pre>{ "contentType": "CR", "secretLevel": 0, "fields": [{ "code": "CR", "field": "Title"}], "data": ["CR"], "filename": "test", "flowCode": "CR", "scriptCode": "test" }</pre>
fileData	multipart	Liste de fichiers	Non	

cf le chapitre [Section 4.2, “Document modèles”](#) pour une description de la structure documentData.



Le type de contenu du corps de la requête doit être "multipart/form-data". Pour ajouter plusieurs fichiers au document, il suffit d'envoyer plusieurs fois le paramètre fileData dans le corps de la requête.

appel curl

```
curl -X 'PATCH' \
  '<host>:<port>/<context>/rest/document/4959' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
    "fields": [
      {
        "code": "CR_DES",
        "field": "test html"
      },
      {
        "code": "CR_RESUME",
        "field": "test html"
      }
    ]
  }'
```

appel requête

```
http://<host>:<port>/<context>/rest/document/4959
```

Table 7. Réponses

Code Http	Description
200	Opération réussie (cf. Exemple de retour)
401	Authentification non fournie
404	Document non trouvé
500	Erreur interne au serveur/service

Exemple de retour

4959

2.1.3.4. **DELETE** /document/{docId}

Cette méthode permet de supprimer logiquement un document.



Les documents enfants seront aussi supprimés.

Table 8. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
docId	path	Identifiant du document	Oui	4959

appel curl

```
curl -X 'DELETE' \
'<host>:<port>/<context>/rest/document/4959' \
-H 'accept: text/plain' \
```

appel requête

```
http://<host>:<port>/<context>/rest/document/4959
```

Table 9. Réponses

Code Http	Description
200	Opération réussie (cf. Liste d'identifiants des documents supprimés)
401	Authentification non fournie
404	Document non trouvé
500	Erreur interne au serveur/service

Liste d'identifiants des documents supprimés

[4959, 2845]

2.1.3.5. **DELETE** /document/remove/{docId}

Cette méthode permet de supprimer physiquement un document.



Les documents enfants seront aussi supprimés.

Table 10. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
docId	path	Identifiant du document	Oui	4959

appel curl

```
curl -X 'DELETE' \
  '<host>:<port>/<context>/rest/document/remove/4959' \
  -H 'accept: text/plain' \
```

appel requête

```
http://<host>:<port>/<context>/rest/document/remove/4959
```

Table 11. Réponses

Code Http	Description
200	Opération réussie (cf. Liste d'identifiants des documents supprimés)
401	Authentification non fournie
404	Document non trouvé
500	Erreur interne au serveur/service

Liste d'identifiants des documents supprimés

```
[4959, 2845]
```

2.2. AuthorityList



Ce Web Service n'est disponible qu'à partir de la version 8 de AIRS Dossier

2.2.1. Client

La classe `com.digitech.dossier.rest.ws.IAuthorityListClient` de la librairie `com.digitech.airs:dossier-rest-client:1.2.1` abstrait l'utilisation de cette API.

2.2.2. Authentification

Ce Web Service requiert une **authentification basique**.

2.2.3. Méthodes

2.2.3.1. GET /authorityList

Cette méthode permet de récupérer tous les codes de liste d'autorité disponibles. Ces codes peuvent ensuite être utilisés au sein de la méthode suivante afin de récupérer les différents termes d'une liste d'autorité.



L'utilisateur connecté doit posséder le droit **DOSSIER_ADM_AIRS_ADMIN**.

appel curl

```
curl -X 'GET' \  
  'http://<host>:<port>/<context>/rest/authorityList' \  
  -H 'accept: application/json'
```

appel requête

```
http://<host>:<port>/<context>/rest/authorityList
```

Table 12. Réponses

Code Http	Description
200	Opération réussie (cf. Structure de retour)
401	Authentification non fournie
403	L'utilisateur connecté ne possède pas les droits suffisants
500	Erreur interne au serveur/service

Structure de retour

```
{  
  "ArrayList": [  
    "DEX_TYPE_DOSSIER",  
    "AL_DOC_LINK",  
    "PRO_APP_FLUX_COU",  
    "PRO_AFF_TYPE",  
    "PRO_DOC_CATEGORIE",  
    "PRO_DOC_CLASSEMENT"  
  ]  
}
```

2.2.3.2. GET /authorityList/{fieldCode}

Cette méthode permet de récupérer les termes d'une liste d'autorité à partir de son code (**fieldCode**)

Table 13. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
fieldCode	path	Code identifiant la liste d'autorité	Oui	PRO_APP_FLUX_COU

appel curl

```
curl -X 'GET' \
  'http://<host>:<port>/<context>/rest/authorityList/PRO_APP_FLUX_COU' \
  -H 'accept: application/json'
```

appel requête

```
http://<host>:<port>/<context>/rest/authorityList/PRO_APP_FLUX_COU
```

Table 14. Réponses

Code Http	Description
200	Opération réussie (cf. Structure de retour)
401	Authentification non fournie
404	Liste d'autorité non trouvée
500	Erreur interne au serveur/service

Structure de retour

```
{
  "ArrayList": [
    {
      "id": 3,
      "code": "NO",
      "label": "Non"
    },
    {
      "id": 4,
      "code": "YES",
      "label": "Oui"
    }
  ]
}
```

cf le chapitre [Section 4.3, “Authority List modèles”](#) pour une description de la structure de retour.

2.3. Baskets



Ce Web Service n'est disponible qu'à partir de la version 8 de AIRS Dossier

2.3.1. Client

La classe `com.digitech.dossier.rest.ws.IBasketsClient` de la librairie `com.digitech.airs:dossier-rest-client:1.2.1` abstrait l'utilisation de cette API.

2.3.2. Authentification

Ce Web Service requiert une **authentification basique**.

2.3.3. Méthodes

2.3.3.1. GET /baskets

Récupère la liste des noms des bannettes paramétrées pour l'application.

appel curl

```
curl -X 'GET' \  
  'http://<host>:<port>/<context>/rest/baskets' \  
  -H 'accept: application/json'
```

appel requête

```
http://<host>:<port>/<context>/rest/baskets
```

Table 15. Réponses

Code Http	Description
200	Opération réussie (cf. Structure de retour)
401	Authentification non fournie
500	Erreur interne au serveur/service

Structure de retour

```
{  
  "ArrayList": [  
    "Documents charte officielle",  
    "Documents du service intégration",  
    "Mes projets au CLI",  
    "Mes projets dirigés",  
    "xml_configuration_basket_global_documents",  
    "Derniers compte rendus",  
    "Mes projets en cours"  
  ]  
}
```


2.3.3.2. GET /baskets/orgId/{orgId}

Cette méthode permet de récupérer la liste des noms des bannettes à partir d'un identifiant d'organisation (**orgId**).

Table 16. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
orgId	path	Identifiant de l'organisation	Oui	24
flowcode	query	Code du flux	Non	PRO

appel curl

```
curl -X 'GET' \
'http://<host>:<port>/<context>/rest/baskets/orgId/24?flowCode=PRO' \
-H 'accept: application/json'
```

appel requête

```
http://<host>:<port>/<context>/rest/baskets/orgId/24?flowCode=PRO
```

Table 17. Réponses

Code Http	Description
200	Opération réussie (cf. Structure de retour)
401	Authentification non fournie
404	Organisation non trouvée
500	Erreur interne au serveur/service

Structure de retour

```
{
  "ArrayList": [
    "Mes projets au SCLI",
    "Mes projets dirigés",
    "Mes projets en cours"
  ]
}
```

2.3.3.3. GET /baskets/flowCode/{flowCode}

Cette méthode permet de récupérer la liste des noms des bannettes à partir d'un code identifiant de flux (**flowCode**).

Table 18. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
flowcode	path	Code du flux	Oui	PRO

appel curl

```
curl -X 'GET' \
  'http://<host>:<port>/<context>/rest/baskets/flowCode/PRO' \
  -H 'accept: application/json'
```

appel requête

```
http://<host>:<port>/<context>/rest/baskets/flowCode/PRO
```

Table 19. Réponses

Code Http	Description
200	Opération réussie (cf. Structure de retour)
401	Authentification non fournie
404	Flux non trouvé
500	Erreur interne au serveur/service

Structure de retour

```
{
  "ArrayList": [
    "Mes projets au CLI",
    "Mes projets dirigés",
    "Mes projets en cours"
  ]
}
```

2.3.3.4. **GET** /baskets/docCount/{basketName}

Cette méthode permet d'obtenir le nombre de documents que contient la bannette de travail identifiée par son nom (**basketName**).

Table 20. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
basketName	path	Nom de la bannette	Oui	Mes projets dirigés
orgId	query	Identifiant de l'organisation	Non	24
flowcode	query	Code du flux	Non	PRO

appel curl

```
curl -X 'GET' \
'http://<host>:<port>/<context>/rest/baskets/docCount/Derniers%20compte%20rendus?orgId=24&flowCode=PRO' \
-H 'accept: application/json'
```

appel requête

```
http://<host>:<port>/<context>/rest/baskets/docCount/Derniers%20compte%20rendus?orgId=24&flowCode=PRO
```

Table 21. Réponses

Code Http	Description
200	Opération réussie (cf. Integer)
401	Authentification non fournie
404	Bannette non trouvée
500	Erreur interne au serveur/service

Integer

415

2.3.3.5. [GET](#) /baskets/docLink/{basketName}

Cette méthode permet d'obtenir les liens URL vers les documents de la bannette de travail identifiée par son nom (**basketName**)

Table 22. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
basketName	path	Nom de la bannette	Oui	Mes projets dirigés
orgId	query	Identifiant de l'organisation	Non	24
flowcode	query	Code du flux	Non	PRO

appel curl

```
curl -X 'GET' \
'http://<host>:<port>/<context>/rest/baskets/docLink/Derniers%20compte%20rendus?orgId=24&flowCode=PRO' \
-H 'accept: application/json'
```

appel requête

```
http://<host>:<port>/<context>/rest/baskets/docLink/Derniers%20compte%20rendus?orgId=24&flowCode=PRO
```

Table 23. Réponses

Code Http	Description
200	Opération réussie (cf. Structure de retour)
401	Authentification non fournie
404	Bannette non trouvée
500	Erreur interne au serveur/service

Structure de retour

```
{
  "ArrayList": [
    "faces/redirect.jsp?authentication=HPa4o3rdP3jo%2FTvySbkDVZMK%2FYuCpRx%2BjOSv5Tp0t9z%2BdIpmS1bB6pgF4ZT9SjaaM5yiNjYtgZ4kuNmMit%2F2LyaaTta6zfYjRsEnfVqCIaXfQqrdxfjVXQNcpILl0f2I1L%2F%2Bovj2AsDD9r3x127k36wMF8MTOW3K8NMG5ouxH8k%3D&outcome=gotoDocumentUnitaire&docId=6868&flowCode=CR",
    "faces/redirect.jsp?authentication=qJoexW7AAhobyxEaEGCpYq77hjkdaayibx%2B5CiCHmX3JD5PbX%2FNtkSQG%2B38CNrNezmUjKGQc7uBjazIzQZwMAeQcgv7dhrpeJH64jAtkBbM37j279eWg2lcQltrqWGF2wqPXrsIejkUGwMqa4S3AQR1CSGHitW9owSwqmuFWXwM%3D&outcome=gotoDocumentUnitaire&docId=6867&flowCode=CR",
    ...
  ]
}
```



Les URL doivent être ajoutés à l'adresse <http://<host>:<port>/<context>> pour pouvoir être utilisées.

2.4. Search



Ce [Web Service](#) n'est disponible qu'à partir de la version 8 de [AIRS Dossier](#)

2.4.1. Client

La classe `com.digitech.dossier.rest.ws.ISearchClient` de la librairie `com.digitech.airs:dossier-rest-client:1.2.1` abstrait l'utilisation de cette **API**.

2.4.2. Authentification

Ce [Web Service](#) requiert une **authentification basique**.

2.4.3. Méthodes

2.4.3.1. [GET](#) /search/docCount

Cette méthode renvoie le nombre de documents remontés par la requête AIRS

Table 24. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
airsRequest	query	Requête AIRS	Oui	D_CREAT>01/01/2018
ctyCodes	query	Liste des types de contenu dans lesquels la recherche est exécutée	Oui	CR
flowCode	query	Code du flux dans lequel la recherche est effectuée	Non	CR

appel curl

```
curl -X 'GET' \
'http://<host>:<port>/<context>/rest/search/docCount?airsRequest=D_CREAT%3E01%2F01%2F2018&ctyCodes=CR&flowCode=CR' \
-H 'accept: application/json'
```

appel requête

```
http://<host>:<port>/<context>/rest/search/docCount?airsRequest=D_CREAT%3E01%2F01%2F2018&ctyCodes=CR&flowCode=CR
```

Table 25. Réponses

Code Http	Description
200	Opération réussie (cf. Integer)
401	Authentification non fournie
500	Erreur interne au serveur/service

Integer

34

2.4.3.2. [GET](#) /search/docLink

Cette méthode permet d'obtenir les liens URL vers les documents renvoyés par la requête AIRS.

Table 26. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
airsRequ est	query	Requête AIRS	Oui	D_CREAT>01/01/2018
ctyCodes	query	Liste des types de contenu dans lesquels la recherche est exécutée	Oui	CR
flowCode	query	Code du flux dans lequel la recherche est effectuée	Non	CR

appel curl

```
curl -X 'GET' \
'http://<host>:<port>/<context>/rest/search/docLink?airsRequest=D_CREAT%3E01%2F01%2F2018&ctyCodes=CR&flowCode=CR' \
-H 'accept: application/json'
```

appel requête

```
http://<host>:<port>/<context>/rest/search/docLink?airsRequest=D_CREAT%3E01%2F01%2F2018&ctyCodes=CR&flowCode=CR
```

Table 27. Réponses

Code Http	Description
200	Opération réussie (cf. Structure de retour)
401	Authentification non fournie
500	Erreur interne au serveur/service

Structure de retour

```
{
  "ArrayList": [
    "faces/redirect.jsp?authentication=HPa4o3rdP3jo%2FTvySbkDVZMK%2FYuCpRx%2BjOSv5Tp0t9z%2BdIpmS1bBGpgF4ZT9SjaaM5yiNjYtgZ4kuNmMit%2F2LyaaTta6zfYjRsEnfVqCIaXfQqrdxfjVXQNcpILl0f2I1L%2F%2Bovj2AsDD9r3x127k36wMF8MTOW3K8NMG5ouxH8k%3D&outcome=gotoDocumentUnitaire&docId=6868&flowCode=CR",
    "faces/redirect.jsp?authentication=qJoexW7AAhobyxEaEGCpYq77hjkdaayibx%2B5CiCHmX3JD5PbX%2FNtkSQG%2B38CNrNezmUjKGQc7uBjazIzQZwMAeQcgv7dhrpeJH64jAtkBbM37j279eWg2lcQltrqWGF2wqPXrsIejkUGwMqa4S3AQR1CSGHitW9owSwqmuFWXwM%3D&outcome=gotoDocumentUnitaire&docId=6867&flowCode=CR",
    ...
  ]
}
```



Les URL doivent être ajoutés à l'adresse <http://<host>:<port>/<context>> pour pouvoir être

utilisées.

2.4.3.3. GET /search/results

Cette méthode permet d'obtenir les métadonnées des documents renvoyés par la requête AIRS.

Table 28. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
airsRequest	query	Requête AIRS	Oui	D_CREAT>01/01/2018
contentTypeCodes	query	Liste des types de contenu dans lesquels la recherche est exécutée	Oui	CR
flowCode	query	Code du flux dans lequel la recherche est effectuée	Non	CR

appel curl

```
curl -X 'GET' \
'http://<host>:<port>/<context>/rest/search/results?airsRequest=D_CREAT%3E01%2F01%2F2018&contentTypeCodes=CR&flowCode=CR' \
-H 'accept: application/json'
```

appel requête

```
http://<host>:<port>/<context>/rest/search/results?airsRequest=D_CREAT%3E01%2F01%2F2018&contentTypeCodes=CR&flowCode=CR
```

Table 29. Réponses

Code Http	Description
200	Opération réussie (cf. WSSearchResultType)
401	Authentification non fournie
500	Erreur interne au serveur/service

WSSearchResultType

```
{
  "ArrayList": [
    {
      "refAirsId": 4372,
      "fields": [
        {
          "code": "D_MODIF",
          "field": "04/02/2021 17:04:36"
        }
      ]
    }
  ]
}
```

```

    },
    {
      "code": "CR_THEME",
      "field": "172"
    },
    {
      "code": "D_CREAT",
      "field": "13/01/2020 16:51:20"
    },
    {
      "code": "CR_DES",
      "field": "CR réu. transverse du lundi 13 janvier 2020-1"
    },
    {
      "code": "CR_REDACTEUR",
      "field": "7"
    },
    {
      "code": "CR_DATE",
      "field": "01/01/2022 0:00:00"
    },
    {
      "code": "CR_RESUME",
      "field": "CR réu. transverse du lundi 13 janvier 2020"
    },
    {
      "code": "T_PRIOR",
      "field": ""
    },
    {
      "code": "MULTI",
      "field": ""
    },
    {
      "code": "CORRES",
      "field": ""
    },
    {
      "code": "MASQUE_NUM",
      "field": ""
    }
  ]
},
{
  "refAirsId": 4473,
  "fields": [
    {
      "code": "D_MODIF",
      "field": "31/01/2020 14:20:19"
    },
    {

```



```

    "code": "CR_THEME",
    "field": "174"
  },
  {
    "code": "D_CREAT",
    "field": "31/01/2020 14:20:19"
  },
  {
    "code": "CR_DES",
    "field": "Café Sideral 3"
  },
  {
    "code": "CR_REDACTEUR",
    "field": "11"
  },
  {
    "code": "CR_DATE",
    "field": "31/01/2020 0:00:00"
  },
  {
    "code": "CR_RESUME",
    "field": "Compte rendu du 3ème café SIDERAL"
  },
  {
    "code": "T_PRIOR",
    "field": ""
  },
  {
    "code": "MULTI",
    "field": ""
  },
  {
    "code": "CORRES",
    "field": ""
  },
  {
    "code": "MASQUE_NUM",
    "field": ""
  }
]
},
...
]
}

```

cf le chapitre [Section 4.4, “Search modèles”](#) pour une description de la structure de retour.

2.5. Attachment



Ce **Web Service** n'est disponible qu'à partir de la version **8** de **AIRS Dossier**

2.5.1. Client

La classe `com.digitech.dossier.rest.ws.IAttachmentClient` de la librairie `com.digitech.airs:dossier-rest-client:1.2.1` abstrait l'utilisation de cette **API**.

2.5.2. Authentification

Ce **Web Service** requiert une **authentification basique**.

2.5.3. Méthodes

2.5.3.1. **POST** /attachment/{docId}

Cette méthode permet d'ajouter ou de mettre à jour une pièce jointe

Table 30. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
attachmentData	multipart	Ensemble de données de la pièce jointe	Oui	<pre>{ "airsType": "ORIGINAL", "label": "Test PJ" }</pre>
fileData	multipart	fichier de la pièce jointe	Oui	

cf le chapitre [Section 4.7, "Attachment modèles"](#) pour une description de la structure attachmentData (Se référer à `WSCommonAttachmentUpdateType`).



Le type de contenu du corps de la requête doit être "multipart/form-data".

appel curl

```
curl --location --request POST '<host>:<port>/<context>/rest/attachment/5072' \
--header 'Authorization: Basic Yi5iaWxsb246cGFzc3dvcmQ=' \
--header 'Cookie: JSESSIONID=6528D38B0114048987669D4EF48BF9F1' \
--form 'attachmentData="{
  "label": "PJ de test"}"
;type=application/json' \
--form 'fileData=@"/C:/PJ.pdf"'
```

appel requête

```
http://<host>:<port>/<context>/rest/attachment/5072
```

Table 31. Réponses

Code Http	Description
200	Opération réussie (cf. Retour, Identifiant de la pièce jointe créée)
401	Authentification non fournie
500	Erreur interne au serveur/service

Retour, Identifiant de la pièce jointe créée

6733

2.5.3.2. [GET](#) /attachment/{docId}

Cette méthode permet de récupérer la liste des informations des des pièces jointes d'un document AIRS.

Table 32. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
docId	path	Identifiant du document AIRS	Oui	4959

appel curl

```
curl -X 'GET' \
  '<host>:<port>/<context>/rest/attachment/5072' \
  -H 'accept: application/json'
```

appel requête

http://<host>:<port>/<context>/rest/attachment/5072

Table 33. Réponses

Code Http	Description
200	Opération réussie (cf. Liste de WSAttachmentType)
401	Authentification non fournie
404	Document non trouvé
500	Erreur interne au serveur/service

Liste de WSAttachmentType

```
{
  "ArrayList": [
    {
      "id": 5220,
      "airsType": "ORIGINAL",
      "label": "viroflay7àé.pdf",
    }
  ]
}
```

```

    "fileName": "viroflay7àé.pdf"
  },
  {
    "id": 6703,
    "airsType": "ORIGINAL",
    "label": "PJ de test",
    "fileName": "PJ.pdf"
  }
]
}

```

cf le chapitre [Section 4.2, “Document modèles”](#) pour une description de la structure de retour.

2.5.3.3. **GET** /attachment/{docId}/{attachmentId}

Cette méthode permet de récupérer une pièce jointe ainsi que son document AIRS.

Table 34. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
docId	path	Identifiant du document	Oui	4959
attachmentId	path	Identifiant de la pièce jointe	Oui	6325

appel curl

```

curl -X 'GET' \
  '<host>:<port>/<context>/rest/attachment/5072/6703' \
  -H 'accept: application/json'

```

appel requête

```
http://<host>:<port>/<context>/rest/attachment/5072/6703
```

Table 35. Réponses

Code Http	Description
200	Opération réussie (cf. WSDocumentType)
401	Authentification non fournie
404	Document non trouvé
500	Erreur interne au serveur/service

WSDocumentType

```

{
  "WSDocumentType": {

```

```

"data": "JVBERi0xLjUNCiW1tbW...",
"filename": "PJ.pdf",
"contentType": "CR",
"secretLevel": 10,
"fields": [
  {
    "code": "D_MODIF",
    "field": "14/11/2022 11:33:29"
  },
  {
    "code": "D_CREAT",
    "field": "21/10/2021 15:52:21"
  },
  {
    "code": "CR_DATE",
    "field": "14/11/2022 0:00:00"
  },
  {
    "code": "CR_RESUME",
    "field": "new CR updated (UnitTest: 2022-11-14T11:33:00)"
  }
]
}
}

```

cf le chapitre [Section 4.2, “Document modèles”](#) pour une description de la structure de retour.

2.5.3.4. **DELETE** /attachment/{docId}

Cette méthode permet de supprimer une pièce jointe d’un document AIRS.

Table 36. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
docId	path	Identifiant du document	Oui	4959
attachme ntId	path	Identifiant de la pièce jointe	Oui	6325

appel curl

```

curl -X 'DELETE' \
  '<host>:<port>/<context>/rest/attachment/5072/6703' \
  -H 'accept: application/json' \

```

appel requête

```

http://<host>:<port>/<context>/rest/attachment/5072/6703

```

Table 37. Réponses

Code Http	Description
200	Opération réussie (cf. Liste d'identifiants des pièces jointes supprimées)
401	Authentification non fournie
404	Document non trouvé
500	Erreur interne au serveur/service

Liste d'identifiants des pièces jointes supprimées

```
{
  "ArrayList": [
    6703
  ]
}
```

2.5.3.5. **GET** /attachment/download/{attachmentId}

Cette méthode permet de télécharger le binaire (contenu) d'une pièce jointe.

Table 38. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
attachmentId	path	Identifiant de la pièce jointe	Oui	6325

appel curl

```
curl -X 'GET' \
  '<host>:<port>/<context>/rest/attachment/download/6703' \
  -H 'accept: application/octet-stream' \
```

appel requête

```
http://<host>:<port>/<context>/rest/attachment/download/6703
```

Table 39. Réponses

Code Http	Description
200	Opération réussie. Byte[] en réponse.
401	Authentification non fournie
404	Pièce jointe non trouvé
500	Erreur interne au serveur/service

2.6. DossierContext



Ce [Web Service](#) n'est disponible qu'à partir de la version 8 de [AIRS Dossier](#)

2.6.1. Client

La classe `com.digitech.dossier.rest.ws.IDossierContextClient` de la librairie `com.digitech.airs:dossier-rest-client:1.2.1` abstrait l'utilisation de cette **API**.

2.6.2. Authentification

Ce [Web Service](#) requiert une **authentification basique**.

2.6.3. Méthodes

2.6.3.1. [GET](#) /dossierContext/flows

Cette méthode renvoie la liste des flux paramétrés au sein de l'application.

appel curl

```
curl -X 'GET' \
  'http://<host>:<port>/<context>/rest/dossierContext/flows' \
  -H 'accept: application/json'
```

appel requête

```
http://<host>:<port>/<context>/rest/dossierContext/flows
```

Table 40. Réponses

Code Http	Description
200	Opération réussie (cf. Liste des flux)
401	Authentification non fournie
500	Erreur interne au serveur/service

Liste des flux

```
{
  "ArrayList": [
    {
      "code": "DOCUMENTATION",
      "label": "Flow created in simplified mode"
    },
    {
      "code": "REF",
      "label": "Flow created in simplified mode"
    }
  ]
}
```

```

    },
    {
      "code": "CPA",
      "label": "Flow created in simplified mode"
    },
    {
      "code": "PRO",
      "label": "Flux Projet"
    },
    {
      "code": "CR",
      "label": "Flow created in simplified mode"
    }
  ]
}

```

cf le chapitre [Section 4.5, “DossierContext modèles”](#) pour une description de la structure de retour.

2.6.3.2. [GET](#) /dossierContext/contentType/{ctyCode}

Cette méthode permet de récupérer la liste des informations d’un type de contenu.

Table 41. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
ctyCode	path	Code du type de contenu	Oui	DOC_CATEGORIE
flowCode	query	Code du flux dans lequel le type de contenu est paramétré	Non	DOCUMENTATION

appel curl

```

curl -X 'GET' \

'http://<host>:<port>/<context>/rest/dossierContext/contentType/DOC_CATEGORIE?flowCode=DOCUMENTATION' \
-H 'accept: application/json'

```

appel requête

```

http://<host>:<port>/<context>/rest/dossierContext/contentType/DOC_CATEGORIE?flowCode=DOCUMENTATION

```

Table 42. Réponses

Code Http	Description
200	Opération réussie (cf. WSContentTypeInfosType)
401	Authentification non fournie

Code Http	Description
404	Type de contenu non trouvé
500	Erreur interne au serveur/service

WSContentTypeInfosType

```
{
  "WSContentTypeInfosType": {
    "code": "DOC_CATEGORIE",
    "id": 15,
    "description": "Catégorie",
    "linksInfosList": [
      {
        "childContentType": {
          "code": "DOC_DOCUMENT",
          "id": 16,
          "description": "Document",
          "linksInfosList": []
        },
        "linkedFieldInfos": [
          {
            "fieldChildcode": "DOC_CAT_CODE",
            "fieldFathercode": "DOC_CAT_CODE"
          }
        ]
      }
    ]
  }
}
```

cf le chapitre [Section 4.5, “DossierContext modèles”](#) pour une description de la structure de retour.

2.6.3.3. **GET** /dossierContext/mandatoryFields/{ctyCode}

Cette méthode permet de récupérer la liste des champs obligatoires d'un type de contenu.

Table 43. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
ctyCode	path	Code du type de contenu	Oui	DOC_CATEGORIE
flowCode	query	Code du flux dans lequel le type de contenu est paramétré	Non	DOCUMENTATION

appel curl

```
curl -X 'GET' \
'http://<host>:<port>/<context>/rest/dossierContext/mandatoryFields/CR?flowCode=CR' \
```

```
-H 'accept: application/json'
```

appel requête

```
http://<host>:<port>/<context>/rest/dossierContext/mandatoryFields/CR?flowCode=CR
```

Table 44. Réponses

Code Http	Description
200	Opération réussie (cf. Structure de retour)
401	Authentification non fournie
404	Type de contenu non trouvé
500	Erreur interne au serveur/service

Structure de retour

```
{
  "ArrayList": [
    "CR_THEME",
    "CR_DES",
    "CR_REDACTEUR",
    "CR_DATE"
  ]
}
```

2.7. ExtendDocument



Ce Web Service n'est disponible qu'à partir de la version 8 de AIRS Dossier

2.7.1. Client

La classe `com.digitech.dossier.rest.ws.IExtendDocumentClient` de la librairie `com.digitech.airs:dossier-rest-client:1.2.1` abstrait l'utilisation de cette API.

2.7.2. Authentification

Ce Web Service requiert une [authentification basique](#).

2.7.3. Méthodes

2.7.3.1. **DELETE** /extendDocument/deleteDocuments

Cette méthode permet de supprimer des documents par lot.

Table 45. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
docIds	<i>query</i>	Liste des identifiants des documents à supprimer	Oui	235

appel curl

```
curl -X 'DELETE' \
'http://<host>:<port>/<context>/rest/extendDocument/deleteDocuments?docIds=1&docIds=2' \
-H 'accept: application/json'
```

appel requête

```
http://<host>:<port>/<context>/rest/extendDocument/deleteDocuments?docIds=1&docIds=2
```

Table 46. Réponses

Code Http	Description
200	Opération réussie (cf. Liste de WSDeleteReportItemType)
401	Authentification non fournie
500	Erreur interne au serveur/service

Liste de *WSDeleteReportItemType*

```
{
  "ArrayList": [
    {
      "docId": 1,
      "state": "OK",
      "children": [],
      "errMessage": ""
    },
    {
      "docId": 2,
      "state": "ERROR",
      "children": [],
      "errMessage": "com.digitel.jcorbairs.exception.ServerException: Le document de
référence 2 n'existe pas"
    }
  ]
}
```

cf le chapitre [Section 4.6, “ExtendDocument modèles”](#) pour une description de la structure de retour.

2.7.3.2. **DELETE** /deleteDocumentsType/{deleteType}

Cette méthode permet de supprimer des documents par lot en spécifiant le type de suppression.

Table 47. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
deleteType	path	Type de suppression. Valeurs possibles : Oui DOC_REMOVE_TYPE_DOC_ONLY DOC_REMOVE_TYPE_CHECK_PARENT DOC_REMOVE_TYPE_FORCE		DOC_REMOVE_TYPE_DOC_ONLY
docIds	query	Liste des identifiants des documents à supprimer	Oui	4959

appel curl

```
curl -X 'GET' \
'<host>:<port>/<context>/rest/extendDocument/deleteDocumentsType/DOC_REMOVE_TYPE_DOC_ONLY?docIds=1&docIds=2' \
-H 'accept: application/json'
```

appel requête

```
http://<host>:<port>/<context>/rest/extendDocument/deleteDocumentsType/DOC_REMOVE_TYPE_DOC_ONLY?docIds=1&docIds=2
```

Table 48. Réponses

Code Http	Description
200	Opération réussie (cf. Liste de WSDeleteReportItemType)
401	Authentification non fournie
500	Erreur interne au serveur/service

Liste de WSDeleteReportItemType

```
{
  "ArrayList": [
    {
      "docId": 1,
      "state": "OK",
      "children": [],
      "errMessage": ""
    },
    {
      "docId": 2,
      "state": "ERROR",

```

```

    "children": [],
    "errorMessage": "com.digitech.jcorbairs.exception.ServerException: Le document de
référence 2 n'existe pas"
  }
]
}

```

cf le chapitre [Section 4.6, “ExtendDocument modèles”](#) pour une description de la structure de retour.

2.7.3.3. [GET](#) /extendDocument/{docId}

Cette méthode permet de récupérer la première pièce jointe d'un document ainsi que la liste des identifiants des autres eventuelles pièces jointes.

Table 49. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
docId	path	Identifiant document	du Oui	4959

appel curl

```

curl -X 'PATCH' \
  '<host>:<port>/<context>/rest/extendDocument/5028' \
  -H 'accept: application/json'

```

appel requête

```
http://<host>:<port>/<context>/rest/extendDocument/5028
```

Table 50. Réponses

Code Http	Description
200	Opération réussie (cf. WSAttachmentListType)
401	Authentification non fournie
404	Document non trouvé
500	Erreur interne au serveur/service

WSAttachmentListType

```

{
  "WSAttachmentListType": {
    "firstAttachment": {
      "id": 5189,
      "airsType": "ORIGINAL",
      "label": "viroflay7àé.pdf",
      "fileName": "viroflay7àé.pdf",

```

```

    "data": "JVBERi0xLjMNCjEz9MNCjEgMCAvYmoNPdw..."
  },
  "items": []
}
}

```

cf le chapitre [Section 4.6, “ExtendDocument modèles”](#) pour une description de la structure de retour.

2.7.3.4. **POST** /extendDocument

Cette méthode permet créer des documents par lot.

Table 51. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
document sData	multipart	Liste de WSInsertDocumentType	de Oui	voir structure de sortie ici .
filesData	multipart	Liste de fichiers	Non	



Il est possible d'ajouter plusieurs fichiers par documents. Le paramètre **filesData** doit contenir toutes les pièces jointes devant être ajoutées pour l'ensemble des documents. Il suffit ensuite d'indiquer au sein de la structure **documentsData** via les paramètres **filesIndex** et **fileName** les index et les noms des pièces jointes qui devront être ajoutés pour le document. Par exemple si **filesData** possède 3 pièces jointes (A, B et C) et que le document nécessite les 2 premières pièce jointes (A et B), il faudra envoyer **filesIndex**=[0,1] et **fileName**=["A","B"]

appel curl

```

curl --location --request POST 'http://<host>:<port>/<context>/rest/extendDocument' \
--header 'Authorization: Basic Yi5iaWxs246cGFzc3dvcmQ=' \
--header 'Cookie: JSESSIONID=38558D0584562F70C045F65639F98D3E' \
--form 'documentsData="{
  \"currentDocument\": {
    \"contentType\": \"CR\",
    \"secretLevel\": 10,
    \"filesIndex\": [0,1],
    \"fields\": [
      {
        \"code\": \"D_MODIF\",
        \"field\": \"04/02/2021 16:58:35\"
      },
      {
        \"code\": \"D_CREAT\",
        \"field\": \"04/04/2019 18:40:22\"
      },
      {

```

```

        "code": "CR_DES",
        "field": "Test 1"
    },
    {
        "code": "CR_REDACTEUR",
        "field": "1"
    },
    {
        "code": "CR_DATE",
        "field": "01/01/2022 0:00:00"
    },
    {
        "code": "CR_RESUME",
        "field": "TEST 1"
    }
]
},
"doUpdateParent": false
}";type=application/json' \
--form 'documentsData="{
    "currentDocument": {
        "contentType": "CR",
        "secretLevel": 10,
        "filesIndex": [1,0],
        "fileName": ["2.pdf", "1.pdf"],
        "fields": [
            {
                "code": "D_MODIF",
                "field": "04/02/2021 16:58:35"
            },
            {
                "code": "D_CREAT",
                "field": "04/04/2019 18:40:22"
            },
            {
                "code": "CR_DES",
                "field": "Test 2"
            },
            {
                "code": "CR_REDACTEUR",
                "field": "1"
            },
            {
                "code": "CR_DATE",
                "field": "01/01/2022 0:00:00"
            },
            {
                "code": "CR_RESUME",
                "field": "TEST 2"
            }
        ]
    }
}
```

```

},
  "doUpdateParent": false
};type=application/json' \
--form 'filesData=@"/C:/1.pdf"' \
--form 'filesData=@"/C:/2.pdf"'

```

appel requête

```
http://<host>:<port>/<context>/rest/extendDocument
```

Table 52. Réponses

Code Http	Description
200	Opération réussie (cf. Liste de WSInsertReportItemType)
401	Authentification non fournie
500	Erreur interne au serveur/service

Liste de *WSInsertReportItemType*

```

{
  "ArrayList": [
    {
      "docId": 6935,
      "state": "OK",
      "message": "",
      "parentState": "NONE"
    },
    {
      "docId": 6936,
      "state": "OK",
      "message": "",
      "parentState": "NONE"
    }
  ]
}

```

cf le chapitre [Section 4.6, “ExtendDocument modèles”](#) pour une description de la structure *WSInsertReportItemType*.

2.7.3.5. PATCH /extendDocument

Cette méthode permet de mettre à jour des documents par lot.

Table 53. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
documente	multipart	Liste de WSExtendDocumentType	Oui	voir structure de sortie ici .
filesData	multipart	Liste de fichiers	Non	



Il est possible d'ajouter plusieurs fichiers par documents. Le paramètre **filesData** doit contenir toutes les pièces jointes devant être ajoutées pour l'ensemble des documents. Il suffit ensuite d'indiquer au sein de la structure **documentsData** via les paramètres **filesIndex** et **fileName** les index et les noms des pièces jointes qui devront être ajoutés pour le document. Par exemple si **filesData** possède 3 pièces jointes (A, B et C) et que le document nécessite les 2 premières pièces jointes (A et B), il faudra envoyer **filesIndex**=[0,1] et **fileName**=["A","B"]

cf le chapitre [Section 4.6, "ExtendDocument modèles"](#) pour une description de la structure WSExtendDocumentType.

appel curl

```
curl --location --request PATCH 'http://<host>:<port>/<context>/rest/extendDocument' \
--header 'Authorization: Basic Yi5iaWxs246cGFzc3dvcmQ=' \
--header 'Cookie: JSESSIONID=9AF46F86AAE176E787C1BCF0F1B6FDFB' \
--form 'documentsData="{
  \"contentType\": \"CR\",
  \"filesIndex\":[0],
  \"fields\": [
    {
      \"code\": \"CR_RESUME\",
      \"field\": \"TEST 4\"
    }
  ],
  \"flowCode\": \"CR\",
  \"docId\": 7363
}";type=application/json' \
--form 'documentsData="{
  \"contentType\": \"CR\",
  \"filesIndex\":[0],
  \"fileName\":[\"DIGITECH_20220214.pdf\"],
  \"fields\": [
    {
      \"code\": \"CR_RESUME\",
      \"field\": \"TEST 5\"
    }
  ],
  \"flowCode\": \"CR\",
  \"docId\": 7362
}";type=application/json' \
```

```
--form 'filesData=@"C:/DIGITECH_20220214.pdf"
```

appel requête

```
http://<host>:<port>/<context>/rest/extendDocument
```

Table 54. Réponses

Code Http	Description
200	Opération réussie (cf. Liste de WSUpdateReportItemType)
401	Authentification non fournie
500	Erreur interne au serveur/service

Liste de *WSUpdateReportItemType*

```
{
  "ArrayList": [
    {
      "docId": 6935,
      "state": "OK",
      "errMessage": ""
    },
    {
      "docId": 6936,
      "state": "OK",
      "errMessage": ""
    }
  ]
}
```

cf le chapitre [Section 4.6, “ExtendDocument modèles”](#) pour une description de la structure *WSUpdateReportItemType*.

2.8. ExtendSearch



Ce Web Service n'est disponible qu'à partir de la version 7.1 de [AIRS Dossier](#)

2.8.1. Client

La classe `com.digitech.dossier.rest.ws.IExtendedSearchClient` de la librairie `com.digitech.airs:dossier-rest-client:1.2.1` abstrait l'utilisation de cette API.

2.8.2. Authentification

Ce Web Service requiert une [authentification basique](#).

2.8.3. Méthodes

2.8.3.1. GET search4FieldsAndAttachments

Cette méthode permet à partir d'une requête **Airs** de récupérer

- des documents
- leurs *metadata* (champs)
- et les informations sur les pièces jointes de chaque document

Table 55. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
airsRequ est	query	Requête Airs	Oui	D_CREAT "28/03/2020"
ctyCodes	query	Liste des types de contenu dans lesquels la recherche est exécutée	Oui	CR
flowCode	query	Code du flux dans lequel la recherche est effectuée. Si Vide, celui-ci sera déduit de la liste des types de contenu	Oui	CR
continuat ionToken	query	Clé de <i>continuation</i> . Le nombre de résultats reçus est limité à 20 (par défaut, côté AIRS Dossier . Si vous souhaitez plus de résultats, le Web Service doit être rappelé en ajoutant cette clé.	Non	OTU4ODQ0ZTFhMDll ODViYTNjN2JiNThkO WM5YjQwNDAjIzE1O TI5MTc3MTM2MDYjIz Q1ODYjIzIw

appel curl

```
curl -X 'GET' \
'http://<host>:<port>/<context>/rest/extendSearch/search4FieldsAndAttachments?airsRequest=D_CREAT%20%3E%3D%20%2201%2F01%2F2020%22&ctyCodes=CR' \
-H 'accept: application/json'

curl -X 'GET' \
'http://<host>:<port>/<context>/rest/extendSearch/search4FieldsAndAttachments?airsRequest=D_CREAT%20%3E%3D%20%2201%2F01%2F2020%22&ctyCodes=CR&ctyCodes=PRO_CLIENT' \
-H 'accept: application/json'
```

appel requête

```
http://<host>:<port>/<context>/rest/extendSearch/search4FieldsAndAttachments?airsRequest=D_CREAT%20%3E%3D%20%2201%2F01%2F2020%22&ctyCodes=CR
```

http://<host>:<port>/<context>/rest/extendSearch/search4FieldsAndAttachments?airsRequest=D_CREATE%20%3E%3D%20%2201%2F01%2F2020%22&ctyCodes=CR&ctyCodes=PRO_CLIENT



La valeur `D_CREATE%20%3E%3D%20%2201%2F01%2F2020%22` correspond à `D_CREATE >= "01/01/2020"`

Table 56. Réponses

Code Http	Description
200	Opération réussie (cf. Structure de retour)
401	Authentification non fournie
500	Erreur interne au serveur/service

Structure de retour

```
{
  "WSExtendSearchResult": {
    "resultNumber": 0,
    "continuationToken": "string",
    "documents": [
      {
        "refAirs": 0,
        "secretLevel": 0,
        "contentType": "string",
        "fields": [
          {
            "code": "string",
            "label": "string",
            "value": "string",
            "refValue": "string"
          }
        ],
        "attachments": [
          {
            "id": 0,
            "label": "string",
            "fileName": "string",
            "airsType": "string",
            "versions": [
              {
                "id": 0,
                "label": "string",
                "fileName": "string",
                "version": 0,
                "date": "string"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```

    }
  ]
}
}

```

cf. le chapitre [Section 4.8, “ExtendedSearch modèles”](#) pour une description de la structure de retour.

2.9. Workbook



Ce [Web Service](#) n’est disponible qu’à partir de la version 8 de [AIRS Dossier](#)

2.9.1. Client

La classe `com.digitech.dossier.rest.ws.IWorkbookClient` de la librairie `com.digitech.airs:dossier-rest-client:1.2.1` abstrait l’utilisation de cette API.

2.9.2. Authentification

Ce [Web Service](#) requiert une [authentification basique](#).

2.9.3. Méthodes

2.9.3.1. **POST** /workbook

Cette méthode permet de créer un classeur

Table 57. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
WSWorkbookType	body	Ensemble de données du classeur	Oui	<pre> { "name": "test", "description": "description", "workbookType": "DOSSIER_WB_TYPE_STD", "status": "DOSSIER_WB_STATUS_ACTIVE", "share": "DOSSIER_WB_SHARE_PUBLIC" } </pre>

cf le chapitre [Section 4.9, “Workbook modèles”](#) pour une description de la structure WsWorkbookType.

appel curl

```

curl --location --request POST '<host>:<port>/<context>/rest/workbook' \
-H 'accept: text/plain' \
-H 'Authorization: Basic QURNOnBhc3N3b3Jk' \

```

```
-H 'Content-Type: application/json' \
-d '{
  "name": "test",
  "description": "description",
  "workbookType": "DOSSIER_WB_TYPE_STD",
  "status": "DOSSIER_WB_STATUS_ACTIVE",
  "share": "DOSSIER_WB_SHARE_PUBLIC"
}'
```

appel requête

```
http://<host>:<port>/<context>/rest/workbook
```

Table 58. Réponses

Code Http	Description
200	Opération réussie (cf. Retour, Identifiant du document)
401	Authentification non fournie
500	Erreur interne au serveur/service

Retour, Identifiant du document

```
8005
```

2.9.3.2. [GET](#) /workbook/{id}

Cette méthode permet de récupérer les métadonnées d'un document à partir de son identifiant

Table 59. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
id	path	Identifiant du classeur	Oui	4959

appel curl

```
curl -X 'GET' \
  '<host>:<port>/<context>/rest/workbook/8001' \
  --header 'Authorization: Basic bi5mZWxpeDpwYXNzd29yZA=='
```

appel requête

```
http://<host>:<port>/<context>/rest/workbook/8001
```

Table 60. Réponses

Code Http	Description
200	Opération réussie (cf. Structure de retour)
401	Authentification non fournie
403	L'utilisateur n'a pas le droit de visualiser le classeur
404	Document non trouvé
500	Erreur interne au serveur/service

Structure de retour

```
{
  "WSWorkbookType": {
    "id": 8001,
    "name": "TEST 75",
    "description": "workbook created at : 2023/07/13 09:15:14",
    "creatorId": 1,
    "workbookType": "DOSSIER_WB_TYPE_STD",
    "shareStartDate": 1689232808000,
    "status": "DOSSIER_WB_STATUS_ACTIVE",
    "share": "DOSSIER_WB_SHARE_SHARED",
    "usersRead": [
      1
    ],
    "usersReadWrite": [
      2,
      59
    ],
    "orgasRead": [],
    "orgasReadWrite": [
      6
    ],
    "documents": []
  }
}
```

cf le chapitre [Section 4.9, “Workbook modèles”](#) pour une description de la structure de retour.

2.9.3.3. [GET](#) /workbook

Cette méthode permet de récupérer tous les classeurs visibles par l'utilisateur

appel curl

```
curl -X 'GET' \
  '<host>:<port>/<context>/rest/workbook' \
  -H 'accept: application/json'
```

appel requête

```
http://<host>:<port>/<context>/rest/workbook
```

Table 61. Réponses

Code Http	Description
200	Opération réussie (cf. Liste d'Integer)
401	Authentification non fournie
500	Erreur interne au serveur/service

Liste d'Integer

```
{
  "ArrayList": [
    7968,
    7981,
    7961,
    7970,
    7973,
    7939,
    7959,
    7887,
    7972,
    7980,
    7899,
    7965,
    7969
  ]
}
```

2.9.3.4. PATCH /workbook

Cette méthode permet de modifier un classeur

Table 62. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
WSWorkbookType	body	Ensemble de données du classeur	Oui	<pre>{ "id": "8001", "name": "test modifié", "description": "description modifiée", "workbookType": "DOSSIER_WB_TYPE_STD", "status": "DOSSIER_WB_STATUS_ACTIVE", "share": "DOSSIER_WB_SHARE_PUBLIC" }</pre>

cf le chapitre [Section 4.9, “Workbook modèles”](#) pour une description de la structure WSWorkbookType.

appel curl

```
curl --location --request PATCH '<host>:<port>/<context>/rest/workbook' \
-H 'accept: text/plain' \
-H 'Authorization: Basic QURNOnBhc3N3b3Jk' \
-H 'Content-Type: application/json' \
-d '{
  "name": "test modifié",
  "description": "description modifiée",
  "workbookType": "DOSSIER_WB_TYPE_STD",
  "status": "DOSSIER_WB_STATUS_ACTIVE",
  "share": "DOSSIER_WB_SHARE_PUBLIC"
}'
```

appel requête

```
http://<host>:<port>/<context>/rest/workbook
```

Table 63. Réponses

Code Http	Description
200	Opération réussie
401	Authentification non fournie
500	Erreur interne au serveur/service

2.9.3.5. **DELETE** /workbook/{id}

Cette méthode permet de supprimer un classeur

Table 64. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
id	path	Identifiant du classeur	Oui	8001

appel curl

```
curl -X 'DELETE' \
  '<host>:<port>/<context>/rest/workbook/8001' \
  -H 'accept: */*' \
  -H 'Authorization: Basic QURN0nBhc3N3b3Jk'
```

appel requête

```
http://<host>:<port>/<context>/rest/workbook/8001
```

Table 65. Réponses

Code Http	Description
200	Opération réussie
401	Authentification non fournie
404	Classeur non trouvé
500	Erreur interne au serveur/service

2.9.3.6. **PATCH** /workbook/{id}/add

Cette méthode permet d'ajouter des documents à un classeur

Table 66. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
id	path	Identifiant du classeur	Oui	7140
docIds	body	Liste d'identifiants des documents à ajouter au classeur	Oui	[5555,6666]

appel curl

```
curl -X 'PATCH' \
  '<host>:<port>/<context>/rest/workbook/7887/add' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '[5555,6666]'
```

appel requête

```
http://<host>:<port>/<context>/rest/workbook/7887/add
```

Table 67. Réponses

Code Http	Description
200	Opération réussie
401	Authentification non fournie
404	Classeur non trouvé
500	Erreur interne au serveur/service

2.9.3.7. PATCH /workbook/{id}/remove

Cette méthode permet de retirer des documents d'un classeur

Table 68. Paramètres en entrée

Paramètre	Type	Description	Requis	Exemple
id	path	Identifiant du classeur	Oui	7140
docIds	body	Liste d'identifiants des documents à retirer du classeur	Oui	[5555,6666]

appel curl

```
curl -X 'PATCH' \
'<host>:<port>/<context>/rest/workbook/7887/remove' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '[5555,6666]'
```

appel requête

```
http://<host>:<port>/<context>/rest/workbook/7887/remove
```

Table 69. Réponses

Code Http	Description
200	Opération réussie
401	Authentification non fournie
404	Classeur non trouvé
500	Erreur interne au serveur/service

3. SOAP



Comme indiqué précédemment, la **wsdl** est accessible.

- La plupart des technologies orienté-objet (**Java**, C#, ...) supporte la génération de l'ensemble des structures des requêtes / réponses en se basant sur ce fichier. La génération de ces "stubs") facilite votre développement mais reste optionnelle.
- Cette documentation est également optionnelle, car le fichier *WSDL*, par définition, suffit pour connaître la structure d'entrée / sortie exacte d'un [Web Service](#).

3.1. Document

La **wsdl** est accessible à l'adresse suivante :

```
http(s)://<hostname>:<port>/<context>/services/WSDocument?wsdl
```

L'encodage des requêtes et des réponses est **UTF-8**.

3.1.1. Client

La classe `com.digitech.dossier.ws.IDocumentWSClient` de la librairie `com.digitech.airs:dossier-ws-stubs:1.3.0` abstrait l'utilisation de cette **API**.

3.1.2. Méthodes

create

Cette méthode permet de créer un document

Table 70. Paramètres en entrée

Paramètre	Description	Requis	Exemple
connectionData	Objet contenant les informations connexion	Oui	<pre><web:connectionData> <type:login>login</type:login> <type:password>password</type:password> </web:connectionData></pre>

Paramètre	Description	Requis	Exemple
document Data	Ensemble de données du document	Oui	<pre> <web:documentData> <type:contentType>CR</type:contentType> <type:data> cid:1245214428780</type:data> <type:fields> <web:item> <type:code>D_MODIF</type:code> <type:field>2/07/2022 15:32:47</type:field> </web:item> <web:item> <type:code>CR_DES</type:code> <type:field>Test creat</type:field> </web:item> <web:item> <type:code> CR_RESUME</type:code> <type:field>test creat</type:field> </web:item> </type:fields> </web:documentData> </pre>

cf le chapitre [Section 4.1](#), “modèles globaux” pour une description de la structure connectionData.

exemple de requête soap

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web=
"http://webservices.dossier.digitech.com" xmlns:type=
"http://type.webservices.dossier.digitech.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:create>
      <web:connectionData>
        <type:login>login</type:login>
        <type:password>password</type:password>
      </web:connectionData>
      <web:documentData>
        <type:contentType>CR</type:contentType>
        <type:data>cid:1245214428780</type:data>
        <type:fields>
          <web:item>
            <type:code>D_MODIF</type:code>
            <type:field>2/07/2022 15:32:47</type:field>
          </web:item>
          <web:item>
            <type:code>CR_DES</type:code>

```

```

        <type:field>Test creat</type:field>
      </web:item>
    </web:fields>
  </web:documentData>
</web:create>
</soapenv:Body>
</soapenv:Envelope>

```

cf le chapitre [Section 4.2, “Document modèles”](#) pour une description de la structure documentData.

Réponse soap

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <createResponse xmlns:types="http://type.webservices.dossier.digitel.com" xmlns="http://webservices.dossier.digitel.com">
      <createReturn>6734</createReturn>
    </createResponse>
  </soap:Body>
</soap:Envelope>

```

get

Cette méthode permet de récupérer les métadonnées d’un document à partir de son identifiant

Table 71. Paramètres en entrée

Paramètre	Description	Requis	Exemple
connectionData	Objet contenant les informations de connexion	Oui	<pre> <web:connectionData> <type:login>login</type:login> <type:password>password</type:password> </web:connectionData> </pre>
docId	Identifiant du document	Oui	4959

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure connectionData.

exemple de requête soap

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.dossier.digitel.com" xmlns:type="http://type.webservices.dossier.digitel.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:get>
      <web:connectionData>

```

```

        <type:login>login</type:login>
        <type:password>password</type:password>
    </web:connectionData>
    <web:docId>4959</web:docId>
</web:get>
</soapenv:Body>
</soapenv:Envelope>

```

Réponse soap

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getResponse xmlns:types="http://type.webservices.dossier.digitel.com" xmlns="http://webservices.dossier.digitel.com">
      <getReturn>
        <types:contentType>CR</types:contentType>
        <types:fields>
          <item>
            <types:code>D_MODIF</types:code>
            <types:field>12/07/2022 15:32:47</types:field>
          </item>
          <item>
            <types:code>CR_THEME</types:code>
            <types:field>Réunion Transverse</types:field>
          </item>
          <item>
            <types:code>D_CREAT</types:code>
            <types:field>24/08/2021 11:02:30</types:field>
          </item>
          <item>
            <types:code>CR_DES</types:code>
            <types:field>test html</types:field>
          </item>
          <item>
            <types:code>CR_REDACTEUR</types:code>
            <types:field>ADM Dossier</types:field>
          </item>
          <item>
            <types:code>CR_DATE</types:code>
            <types:field>24/08/2021 0:00:00</types:field>
          </item>
          <item>
            <types:code>CR_RESUME</types:code>
            <types:field>test html</types:field>
          </item>
        </types:fields>
        <types:secretLevel>10</types:secretLevel>
      </getReturn>
    </getResponse>
  </soap:Body>

```

```
</soap:Envelope>
```

cf le chapitre [Section 4.2, “Document modèles”](#) pour une description de la structure de retour. (Se référer à documentData).

update

Cette méthode permet de mettre à jour les métadonnées d’un document

Table 72. Paramètres en entrée

Paramètre	Description	Requis	Exemple
connectionData	Objet contenant les informations de connexion	Oui	<pre><web:connectionData> <type:login>login</type:login> <type:password>password</type:password> </web:connectionData></pre>
docId	Identifiant du document	Oui	4959
documentData	Ensemble de données du document	Oui	<pre><web:documentData> <type:contentType>CR</type:contentType> <type:data> cid:1245214428780</type:data> <type:fields> <web:item> <type:code>D_MODIF</type:code> <type:field>2/07/2022 15:32:47</type:field> </web:item> <web:item> <type:code>CR_DES</type:code> <type:field>Test creat</type:field> </web:item> <web:item> <type:code> CR_RESUME</type:code> <type:field>test creat</type:field> </web:item> </type:fields> </web:documentData></pre>

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure connectionData.

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web=
"http://webservices.dossier.digitel.com" xmlns:type=
"http://type.webservices.dossier.digitel.com">
```



```

<soapenv:Header/>
<soapenv:Body>
  <web:update>
    <web:connectionData>
      <type:login>login</type:login>
      <type:password>password</type:password>
    </web:connectionData>
    <web:docId>6734</web:docId>
    <web:documentData>
      <type:contentType>CR</type:contentType>
      <type:fields>
        <web:item>
          <type:code>CR_DES</type:code>
          <type:field>TEST UPDATE</type:field>
        </web:item>
      </type:fields>
    </web:documentData>
  </web:update>
</soapenv:Body>
</soapenv:Envelope>

```

Réponse soap

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <updateResponse xmlns:types="http://type.webservices.dossier.digitech.com" xmlns="http://webservices.dossier.digitech.com">
      <updateReturn>6734</updateReturn>
    </updateResponse>
  </soap:Body>
</soap:Envelope>

```

delete

Cette méthode permet de supprimer logiquement un document.

Table 73. Paramètres en entrée

Paramètre	Description	Requis	Exemple
connectionData	Objet contenant les informations de connexion	Oui	<pre> <web:connectionData> <type:login>login</type:login> <type:password>password</type:password> </web:connectionData> </pre>
docId	Identifiant du document	Oui	4959

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure connectionData.

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web=
"http://webservices.dossier.digitech.com" xmlns:type=
"http://type.webservices.dossier.digitech.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:delete>
      <web:connectionData>
        <type:login>login</type:login>
        <type:password>password</type:password>
      </web:connectionData>
      <web:docId>6734</web:docId>
    </web:delete>
  </soapenv:Body>
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <deleteResponse xmlns:types="http://type.webservices.dossier.digitech.com" xmlns=
"http://webservices.dossier.digitech.com">
      <deleteReturn>6734</deleteReturn>
    </deleteResponse>
  </soap:Body>
</soap:Envelope>
```

remove

Cette méthode permet de supprimer physiquement un document.

Table 74. Paramètres en entrée

Paramètre	Description	Requis	Exemple
connection Data	Objet contenant les informations connexion	Oui	<pre><web:connectionData> <type:login>login</type:login> <type:password>password</type:password> </web:connectionData></pre>
docId	Identifiant du document	Oui	4959

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure connectionData.

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web=
"http://webservices.dossier.digitech.com" xmlns:type=
"http://type.webservices.dossier.digitech.com">
  <soapenv:Header/>
```

```
<soapenv:Body>
  <web:remove>
    <web:connectionData>
      <type:login>login</type:login>
      <type:password>password</type:password>
    </web:connectionData>
    <web:docId>6734</web:docId>
  </web:remove>
</soapenv:Body>
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <removeResponse xmlns:types="http://type.webservices.dossier.digitech.com" xmlns="http://webservices.dossier.digitech.com">
      <deleteReturn>6734</deleteReturn>
    </removeResponse>
  </soap:Body>
</soap:Envelope>
```

3.2. AuthorityList

La **wsdl** est accessible à l'adresse suivante :

```
http(s)://<hostname>:<port>/<context>/services/WSAuthorityList?wsdl
```

L'encodage des requêtes et des réponses est **UTF-8**.

3.2.1. Client

La classe `com.digitech.dossier.ws.IAuthorityListWSClient` de la librairie **com.digitech.air:dossier-ws-stubs:1.3.0** abstrait l'utilisation de cette API.

3.2.2. Méthodes

getAuthorityListCodes

Cette méthode permet de récupérer tous les codes de liste d'autorité disponibles. Ces codes peuvent ensuite être utilisés au sein de la méthode suivante afin de récupérer les différents termes d'une liste d'autorité.

3.2.3. Authentification



Aucun paramètre d'entrée n'est requis pour l'utilisation de cette méthode.

Une **authentification basique** est cependant nécessaire.

L'utilisateur connecté doit également posséder le droit **DOSSIER_ADM_AIRS_ADMIN**.

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.dossier.digitel.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getAuthorityListCodes/>
  </soapenv:Body>
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getAuthorityListCodesResponse xmlns:types="http://type.webservices.dossier.digitel.com" xmlns="http://webservices.dossier.digitel.com">
      <code>DEX_TYPE_DOSSIER</code>
      <code>AL_DOC_LINK</code>
      <code>PRO_APP_FLUX_COU</code>
      <code>PRO_AFF_TYPE</code>
      <code>PRO_DOC_CATEGORIE</code>
    </getAuthorityListCodesResponse>
  </soap:Body>
</soap:Envelope>
```

getTerms

Cette méthode permet de récupérer les termes d'une liste d'autorité à partir de son code (fieldCode)

Table 75. Paramètres en entrée

Paramètre	Description	Requis	Exemple
connectionData	Objet contenant les informations de connexion	Oui	<pre><web:connectionData> <type:login>login</type:login> <type:password>password</type:password> </web:connectionData></pre>
fieldCode	Code identifiant la liste d'autorité	Oui	AL_DOC_LINK

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure connectionData.

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.dossier.digitel.com" xmlns:type="http://type.webservices.dossier.digitel.com">
```

```
<soapenv:Header/>
<soapenv:Body>
  <web:getTerms>
    <web:connectionData>
      <type:login>login</type:login>
      <type:password>password</type:password>
      <type:authenticationId>?</type:authenticationId>
    </web:connectionData>
    <web:fieldCode>AL_DOC_LINK</web:fieldCode>
  </web:getTerms>
</soapenv:Body>
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getTermsResponse xmlns:types="http://type.webservices.dossier.digitech.com" xmlns="http://webservices.dossier.digitech.com">
      <getTermsReturn>
        <code>pere-fils</code>
        <id>1</id>
        <label>pere-fils</label>
      </getTermsReturn>
    </getTermsResponse>
  </soap:Body>
</soap:Envelope>
```

cf le chapitre [Section 4.3, “Authority List modèles”](#) pour une description de la structure de retour. (Se référer à WSTerm).

3.3. Baskets

La **wsdl** est accessible à l’adresse suivante :

```
http(s)://<hostname>:<port>/<context>/services/WSBaskets?wsdl
```

L’encodage des requêtes et des réponses est **UTF-8**.

3.3.1. Client

La classe `com.digitech.dossier.ws.IBasketsWSClient` de la librairie `com.digitech.airs:dossier-ws-stubs:1.3.0` abstrait l’utilisation de cette **API**.

3.3.2. Méthodes

getBasketsList

Cette méthode permet de récupérer la liste des noms des bannettes paramétrées pour l'application.

Table 76. Paramètres en entrée

Paramètre	Description	Requis	Exemple
connectionData	Objet contenant les informations de connexion	Oui	<pre><web:connectionData> <type:login>login</type:login> <type:password>password</type:password> </web:connectionData></pre>

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure connectionData.

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.dossier.digitech.com" xmlns:type="http://type.webservices.dossier.digitech.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getBasketsList>
      <web:connectionData>
        <type:login>login</type:login>
        <type:password>password</type:password>
      </web:connectionData>
    </web:getBasketsList>
  </soapenv:Body>
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getBasketsListResponse xmlns:types="http://type.webservices.dossier.digitech.com" xmlns="http://webservices.dossier.digitech.com">
      <getBasketsListReturn>Documents charte officielle</getBasketsListReturn>
      <getBasketsListReturn>Documents du service intégration</getBasketsListReturn>
      <getBasketsListReturn>Mes projets au SCLI</getBasketsListReturn>
      <getBasketsListReturn>Mes projets dirigés</getBasketsListReturn>
      <getBasketsListReturn>
        xml_configuration_basket_global_documents</getBasketsListReturn>
      <getBasketsListReturn>Derniers compte rendus</getBasketsListReturn>
      <getBasketsListReturn>Mes projets en cours</getBasketsListReturn>
    </getBasketsListResponse>
  </soap:Body>
</soap:Envelope>
```

getBasketsListForOrganization

Cette méthode permet de récupérer la liste des noms des bannettes à partir d'un identifiant d'organisation (organizationId).

Table 77. Paramètres en entrée

Paramètre	Description	Requis	Exemple
connectionData	Objet contenant les informations de connexion	Oui	<pre><web:connectionData> <type:login>login</type:login> <type:password>password</type:password> </web:connectionData></pre>
organizationId	Identifiant de l'organisation	Oui	5
flowCode	Code du flux	Non	PRO

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure connectionData.

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.dossier.digitech.com" xmlns:type="http://type.webservices.dossier.digitech.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getBasketsListForOrganization>
      <web:connectionData>
        <type:login>login</type:login>
        <type:password>password</type:password>
      </web:connectionData>
      <web:organizationId>5</web:organizationId>
      <!--Optional:-->
      <web:flowCode>PRO</web:flowCode>
    </web:getBasketsListForOrganization>
  </soapenv:Body>
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getBasketsListForOrganizationResponse xmlns:types="http://type.webservices.dossier.digitech.com" xmlns="http://webservices.dossier.digitech.com">
      <getBasketsListForOrganizationReturn>Mes projets au
SCLI</getBasketsListForOrganizationReturn>
      <getBasketsListForOrganizationReturn>Mes projets
dirigés</getBasketsListForOrganizationReturn>
      <getBasketsListForOrganizationReturn>Mes projets en
```

```

<getBasketsListForOrganizationReturn>
  </getBasketsListForOrganizationResponse>
</soap:Body>
</soap:Envelope>

```

getBasketsListFromFlow

Cette méthode permet de récupérer la liste des noms des bannettes à partir d'un code identifiant de flux (flowCode).

Table 78. Paramètres en entrée

Paramètre	Description	Requis	Exemple
connectionData	Objet contenant les informations de connexion	Oui	<pre> <web:connectionData> <type:login>login</type:login> <type:password>password</type:password> </web:connectionData> </pre>
flowCode	Code du flux	Oui	PRO

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure connectionData.

exemple de requête soap

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.dossier.digitel.com" xmlns:type="http://type.webservices.dossier.digitel.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getBasketsListFromFlow>
      <web:connectionData>
        <type:login>login</type:login>
        <type:password>password</type:password>
      </web:connectionData>
      <web:flowCode>PRO</web:flowCode>
    </web:getBasketsListFromFlow>
  </soapenv:Body>
</soapenv:Envelope>

```

Réponse soap

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getBasketsListFromFlowResponse xmlns:types="http://type.webservices.dossier.digitel.com" xmlns="http://webservices.dossier.digitel.com">
      <getBasketsListFromFlowReturn>Mes projets au CLI</getBasketsListFromFlowReturn>
      <getBasketsListFromFlowReturn>Mes projets dirigés</getBasketsListFromFlowReturn>
      <getBasketsListFromFlowReturn>Mes projets en cours</getBasketsListFromFlowReturn>
    </getBasketsListFromFlowResponse>
  </soap:Body>
</soap:Envelope>

```



```
</soap:Body>
</soap:Envelope>
```

getDocLinkFromBasketName

Cette méthode permet d'obtenir les liens URL vers les documents de la bannette de travail identifiée par son nom (basketName).

Table 79. Paramètres en entrée

Paramètre	Description	Requis	Exemple
login	Login utilisateur	Oui	login
password	Mot de passe utilisateur	Oui	password
basketName	Nom de la bannette	Oui	Derniers compte rendus
organizationId	Identifiant de l'organisation	Non	5
flowCode	Code du flux	Non	PRO

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.dossier.digitech.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getDocLinkFromBasketName>
      <web:login>login</web:login>
      <web:password>password</web:password>
      <web:basketName>Derniers compte rendus</web:basketName>
      <!--Optional:-->
      <web:flowCode>CR</web:flowCode>
      <!--Optional:-->
      <web:organizationId>5</web:organizationId>
    </web:getDocLinkFromBasketName>
  </soapenv:Body>
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getDocLinkFromBasketNameResponse xmlns:types="http://type.webservices.dossier.digitech.com" xmlns="http://webservices.dossier.digitech.com">
      <getDocLinkFromBasketNameReturn>faces/redirect.jsp?authentication=Mv0rX2e90UFVMpqjMXsaC2Ry%2BYkDnB5snh6sVKmv0GsidZqrn00kAf71AftRBMVBWdtAvNya2EtwTWoNhWJJL1kFv6fqrva0umZI4jkJCsN7wTMZSrBIMDhNfwfow8gA0fCwX6lBMav%2Fhkb5pVEGuJdHeM3SAC5N5YoME%2BVpJo%3D&outcome=gotoDocumentUnitaire
```

```
&docId=6868&flowCode=CR</getDocLinkFromBasketNameReturn>

<getDocLinkFromBasketNameReturn>faces/redirect.jsp?authentication=1jWrCAqm8Fe0avhrNrLJb4G8v%2
FYhMh2YHGrruJ4F0GKnip8TSjTg31h4niV1a5qqNnUF%2B%2B1XaWl8b1jvgCZYOGJLMKxHE81eWCvMhxVlr0XdyG22Ai
wb6PWnxNWNcHNvF6YhcbEUsgqv2Iz4I33i%2BwQITXBtZty1P%2FNU28tJ6LY%3D&outcome=gotoDocumentUnit
aire&docId=6867&flowCode=CR</getDocLinkFromBasketNameReturn>

<getDocLinkFromBasketNameReturn>faces/redirect.jsp?authentication=IuJNqp4wbGBCPohakrVuL0zirgz
4ytKNxXkV13ovGo8Edep7JJ%2BmSjYQE7qoiDYs%2BzkRF1Z0fe9ZSTpJi7ovPRiVV7V6nyLPSbMVQNLI6g%2BtJPF0kb
SHOPW1HsdNeOHW4tTZP116LGE7rKhgdnbA3WWSEW6mCTqJfK%2BZpfqTkXk%3D&outcome=gotoDocumentUnitai
re&docId=6866&flowCode=CR</getDocLinkFromBasketNameReturn>
  </getDocLinkFromBasketNameResponse>
</soap:Body>
</soap:Envelope>
```



Les URL doivent être ajoutés à l'adresse <http://<host>:<port>/<context>> pour pouvoir être utilisées.

getDocLinkFromBasketNameTAuthId

Cette méthode permet d'obtenir les liens URL vers les documents de la bannette de travail identifiée par son nom (basketName). L'authentification se fait par un jeton (authentication).

Table 80. Paramètres en entrée

Paramètre	Description	Requis	Exemple
authentication	Paramètre d'authentification encodé en base 64	Oui	SXJpcyZBbWJyZQ==
basketName	Nom de la bannette	Oui	Derniers compte rendus
organizationId	Identifiant de l'organisation	Non	5
flowCode	Code du flux	Non	PRO

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web=
"http://webservices.dossier.digitech.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getDocLinkFromBasketNameTAuthId>
      <web:authentication>SXJpcyZBbWJyZQ==</web:authentication>
      <web:basketName>TEST</web:basketName>
      <!--Optional:-->
      <web:flowCode>CR</web:flowCode>
      <!--Optional:-->
      <web:organizationId>5</web:organizationId>
    </web:getDocLinkFromBasketNameTAuthId>
```

```
</soapenv:Body>
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getDocLinkFromBasketNameTAuthIdResponse xmlns:types=
"http://type.webservices.dossier.digitech.com" xmlns="
http://webservices.dossier.digitech.com">

<getDocLinkFromBasketNameReturn>faces/redirect.jsp?authentication=Mv0rX2e90UFVMpqjMXsaC2Ry%2B
YkDnB5snh6sVKmv0GsidZqrn00kAf71AftRBMVBWdtAvNya2EtwTWoNhWJJL1kFv6fqrVAt0umZI4jkJCsN7wTMZSrBIM
DhNfwfow8gA0fCwX6lBMav%2Fhkb5pVEGuJdHeM3SAC5N5YoME%2BVpJo%3D&outcome=gotoDocumentUnitaire
&docId=6868&flowCode=CR</getDocLinkFromBasketNameReturn>

<getDocLinkFromBasketNameReturn>faces/redirect.jsp?authentication=1jWrCAqm8Fe0avhrNr1Jb4G8v%2
FYhMh2YHGrruJ4F0GKnip8TSjTg31h4niV1a5qqNnUF%2B%2B1XaW18b1jvgCZYOGJLMKxHE81eWCvMhxVlr0XdyG22Ai
wb6PwnxNWNcHNv6YhcbEUsgqv2Iz4I33i%2BwQITXBtZty1P%2FNu28tJ6LY%3D&outcome=gotoDocumentUnit
aire&docId=6867&flowCode=CR</getDocLinkFromBasketNameReturn>

<getDocLinkFromBasketNameReturn>faces/redirect.jsp?authentication=IuJNqp4wbGBCPohakrVu10zirgz
4ytKNxXkV13ovGo8Edep7JJ%2BmSjYQE7qoiDYs%2BzkRF1Z0fe9ZSTPji7ovPRiVV7V6nyLPSbMVQNLi6g%2BtJPF0kb
SHOPW1HsdNeOhw4tTZP116LGE7rKhgdnbA3WWSEW6mCTqJfK%2BZpfqTkXk%3D&outcome=gotoDocumentUnitai
re&docId=6866&flowCode=CR</getDocLinkFromBasketNameReturn>
    </getDocLinkFromBasketNameTAuthIdResponse>
  </soap:Body>
</soap:Envelope>
```



Les URL doivent être ajoutés à l'adresse <http://<host>:<port>/<context>> pour pouvoir être utilisées.

getDocNumberFromBasketName

Cette méthode permet d'obtenir le nombre de documents que contient la bannette de travail identifiée par son nom (basketName).

Table 81. Paramètres en entrée

Paramètre	Description	Requis	Exemple
login	Login utilisateur	Oui	login
password	Mot de passe utilisateur	Oui	password
basketName	Nom de la bannette	Oui	Derniers compte rendus
flowCode	Code du flux	Non	PRO
organizationId	Identifiant de l'organisation	Non	5

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.dossier.digitech.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getDocNumberFromBasketName>
      <web:login>login</web:login>
      <web:password>password</web:password>
      <web:flowCode>CR</web:flowCode>
      <web:basketName>Derniers compte rendus</web:basketName>
    </web:getDocNumberFromBasketName>
  </soapenv:Body>
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getDocNumberFromBasketNameResponse xmlns:types="http://type.webservices.dossier.digitech.com" xmlns="http://webservices.dossier.digitech.com">
      <getDocNumberFromBasketNameReturn>415</getDocNumberFromBasketNameReturn>
    </getDocNumberFromBasketNameResponse>
  </soap:Body>
</soap:Envelope>
```

getDocNumberFromBasketNameTAuthId

Cette méthode permet d'obtenir le nombre de documents que contient la bannette de travail identifiée par son nom (basketName). L'authentification se fait par un jeton (authentication).

Table 82. Paramètres en entrée

Paramètre	Description	Requis	Exemple
authentication	Paramètre d'authentification encodé en base 64	Oui	SXJpcyZBbWJyZQ==
basketName	Nom de la bannette	Oui	Derniers compte rendus
flowCode	Code du flux	Non	PRO
organizationId	Identifiant l'organisation	de Non	5

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.dossier.digitech.com">
  <soapenv:Header/>
```

```
<soapenv:Body>
  <web:getDocNumberFromBasketNameTAuthId>
    <web:authentication>SXJpcyZBbWJyZQ==</web:authentication>
    <web:basketName>Derniers compte rendus</web:basketName>
    <!--Optional:-->
    <web:flowCode>PRO</web:flowCode>
    <!--Optional:-->
    <web:organizationId>5</web:organizationId>
  </web:getDocNumberFromBasketNameTAuthId>
</soapenv:Body>
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getDocNumberFromBasketNameTAuthIdResponse xmlns:type=
"http://type.webservices.dossier.digitech.com" xmlns=
"http://webservices.dossier.digitech.com">
      <getDocNumberFromBasketNameTAuthIdReturn>0</getDocNumberFromBasketNameTAuthIdReturn>
    </getDocNumberFromBasketNameTAuthIdResponse>
  </soap:Body>
</soap:Envelope>
```

getResultFromBasketName

Cette méthode permet d'obtenir le nombre de documents que contient la bannette de travail identifiée par son nom (basketName).

Table 83. Paramètres en entrée

Paramètre	Description	Requis	Exemple
connectionData	Objet contenant les informations de connexion	Oui	<pre><web:connectionData> <type:login>login</type:login> <type:password>password</type:password> </web:connectionData></pre>
basketName	Nom de la bannette	Oui	Derniers compte rendus
flowCode	Code du flux	Non	PRO
organizationId	Identifiant de l'organisation	Non	5

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure connectionData.

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web=
"http://webservices.dossier.digitech.com" xmlns:type=
```

```
"http://type.webservices.dossier.digitech.com">
<soapenv:Header/>
<soapenv:Body>
  <web:getResultFromBasketName>
    <web:connectionData>
      <type:login>login</type:login>
      <type:password>password</type:password>
    </web:connectionData>
    <web:basketName>Derniers compte rendus</web:basketName>
    <!--Optional:-->
    <web:flowCode>CR</web:flowCode>
    <!--Optional:-->
    <web:organizationId>5</web:organizationId>
  </web:getResultFromBasketName>
</soapenv:Body>
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getResultFromBasketNameResponse xmlns:types=
"http://type.webservices.dossier.digitech.com" xmlns=
"http://webservices.dossier.digitech.com">
      <getResultFromBasketNameReturn>415</getResultFromBasketNameReturn>
    </getResultFromBasketNameResponse>
  </soap:Body>
</soap:Envelope>
```

getResultLinkFromBasketName

Cette méthode permet d'obtenir les liens URL vers les documents de la bannette de travail identifiée par son nom (basketName).

Table 84. Paramètres en entrée

Paramètre	Description	Requis	Exemple
connectionData	Objet contenant les informations de connexion	Oui	<pre><web:connectionData> <type:login>login</type:login> <type:password>password</type:password> </web:connectionData></pre>
basketName	Nom de la bannette	Oui	Derniers compte rendus
organizationId	Identifiant de l'organisation	Non	5
flowCode	Code du flux	Non	PRO

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure `connectionData`.

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web=
"http://webservices.dossier.digitech.com" xmlns:type=
"http://type.webservices.dossier.digitech.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getResultLinkFromBasketName>
      <web:connectionData>
        <type:login>login</type:login>
        <type:password>password</type:password>
      </web:connectionData>
      <web:basketName>Derniers compte rendus</web:basketName>
      <!--Optional:-->
      <web:flowCode>CR</web:flowCode>
      <!--Optional:-->
      <web:organizationId>5</web:organizationId>
    </web:getResultLinkFromBasketName>
  </soapenv:Body>
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getResultLinkFromBasketNameResponse xmlns:types=
"http://type.webservices.dossier.digitech.com" xmlns="
http://webservices.dossier.digitech.com">

      <getResultLinkFromBasketNameReturn>faces/redirect.jsp?authentication=A2y%2Fc7NFgsfek6UDydcZ2H
wCnYHcqDZ%2FNRQTt5QkjTtk1D9MbZ4R%2FtohACItD6yVF%2BskkHER6z70v0byN3A0IbozfdAFqDzhnF8VAXEZKyMK
%2BECcUV4r0t7fXwXiyLYozyVlb80hPBNqsyL0GgjdJkTBtX41bpLVpkwyE41dRE%3D&outcome=gotoDocumentU
nitaire&docId=6868&flowCode=CR</getResultLinkFromBasketNameReturn>

      <getResultLinkFromBasketNameReturn>faces/redirect.jsp?authentication=Wlwao36%2F6F2KTHoGTAlomp
wI75HiXc2R0a3P011uz06Qndvu7uQ68HnVOBv0ur6r2tFMeU5qGXvt4KICddq6xIODtBcp447QgAQGVXqaw9PfIOLgslly
sLH%2BxjFmnDcceoo7mF08ngfiAfKlJ3NEuOW%2FM9Y0%2Fu07D0xmKYV%2F6gQ%3D&outcome=gotoDocumentU
nitaire&docId=6867&flowCode=CR</getResultLinkFromBasketNameReturn>
    </getResultLinkFromBasketNameResponse>
  </soap:Body>
</soap:Envelope>
```



Les URL doivent être ajoutés à l'adresse `http://<host>:<port>/<context>/` pour pouvoir être utilisées.

3.4. Search

La **wsdl** est accessible à l'adresse suivante :

```
http(s)://<hostname>:<port>/<context>/services/WSSearch?wsdl
```

L'encodage des requêtes et des réponses est **UTF-8**.

3.4.1. Client

La classe `com.digitech.dossier.ws.ISearchWSClient` de la librairie `com.digitech.airs:dossier-ws-stubs:1.3.0` abstrait l'utilisation de cette **API**.

3.4.2. Méthodes

getResultNumberFromAirsRequest

Cette méthode renvoie le nombre de documents remontés par la requête AIRS

Table 85. Paramètres en entrée

Paramètre	Description	Requis	Exemple
connectionData	Objet contenant les informations de connexion	Oui	<pre><web:connectionData> <type:login>login</type:login> <type:password>password</type:password> </web:connectionData></pre>
ctyCodeList	Liste des types de contenu dans lesquels la recherche est exécutée	Oui	CR
flowCode	Code du flux dans lequel la recherche est effectué	Non	CR

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure `connectionData`.

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.dossier.digitech.com" xmlns:type="http://type.webservices.dossier.digitech.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getResultNumberFromAirsRequest>
      <web:connectionData>
        <type:login>login</type:login>
        <type:password>password</type:password>
      </web:connectionData>
      <web:airsRequest>D_CREAT>01/01/2018</web:airsRequest>
    </web:getResultNumberFromAirsRequest>
  </soapenv:Body>
</soapenv:Envelope>
```



```

<!--1 or more repetitions:-->
<web:ctyCodeList>CR</web:ctyCodeList>
<!--Optional:-->
<web:flowCode>CR</web:flowCode>
</web:getResultNumberFromAirsRequest>
</soapenv:Body>
</soapenv:Envelope>

```

Réponse soap

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getResultNumberFromAirsRequestResponse xmlns:types=
"http://type.webservices.dossier.digitel.com" xmlns="
http://webservices.dossier.digitel.com">
      <getResultNumberFromAirsRequestReturn>551</getResultNumberFromAirsRequestReturn>
    </getResultNumberFromAirsRequestResponse>
  </soap:Body>
</soap:Envelope>

```

getResultLinkFromAirsRequest

Cette méthode permet d'obtenir les liens URL vers les documents renvoyés par la requête AIRS.

Table 86. Paramètres en entrée

Paramètre	Description	Requis	Exemple
connectionData	Objet contenant les informations de connexion	Oui	<pre> <web:connectionData> <type:login>login</type:login> <type:password>password</type:password> </web:connectionData> </pre>
ctyCodeList	Liste des types de contenu dans lesquels la recherche est exécutée	Oui	CR
flowCode	Code du flux dans lequel la recherche est effectué	Non	CR

cf le chapitre [Section 4.1, "modèles globaux"](#) pour une description de la structure connectionData.

exemple de requête soap

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web=
"http://webservices.dossier.digitel.com" xmlns:type=
"http://type.webservices.dossier.digitel.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getResultLinkFromAirsRequest>
      <web:connectionData>

```

```

<type:login>login</type:login>
<type:password>password</type:password>
</web:connectionData>
<web:airsRequest>D_CREAT>25/09/2022</web:airsRequest>
<!--1 or more repetitions:-->
<web:ctyCodeList>CR</web:ctyCodeList>
<!--Optional:-->
<web:flowCode>CR</web:flowCode>
</web:getResultLinkFromAirsRequest>
</soapenv:Body>
</soapenv:Envelope>

```

Réponse soap

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getResultLinkFromAirsRequestResponse xmlns:types=
"http://type.webservices.dossier.digitech.com" xmlns="
http://webservices.dossier.digitech.com">

<getResultLinkFromAirsRequestReturn>faces/redirect.jsp?authentication=Wzm9BmKpzJAWIe1KVgWQG9
CGhop%2B2VmFOvoqMivhi6PTudpEPzs7ko2A9Dk5h%2Fs5ewfb7%2F5PI8pD38NdbZpuJELzD%2Ff2UmwWuG6ooYlgS9P
PIXK1rcM1MLBh23H02Nf1CsQ5m7Y0qqpZHJraYoEz%2BkhHhb7VxFat3Mh6Tvv%2FZg%3D&outcome=gotoDocume
ntUnitaire&docId=6874&flowCode=CR</getResultLinkFromAirsRequestReturn>

<getResultLinkFromAirsRequestReturn>faces/redirect.jsp?authentication=cQtFv9no87E66GU8KUvhQU3
uLmHErCQroaKB%2B1NLt%2FKRlYn2Zo7P0OCuaCcyMXJkXAYBLxz6P0aQig6us1m9CQZhApFbSkSFucoueVeLL9LIXIUk
3upII1qLrRL4uyccZW03LkyJ3SQEQpgr4Cc3Z7rqSVzU9BsLCd1B4qDLZLM%3D&outcome=gotoDocumentUnitai
re&docId=6875&flowCode=CR</getResultLinkFromAirsRequestReturn>

<getResultLinkFromAirsRequestReturn>faces/redirect.jsp?authentication=CEZ%2BMPiGhqN3lyVj2eyJY
17JLxBI2W5b3AY6HaShREaZGrj22Mhu%2FJ4yfappK38m0ezA3%2BhST2sJl32lc6Fpu%2FE1Ed1Vizw%2Fj4CohEn3sI
mzxZvcpHzN4LFBPUgkxThc%2BLADPjdfWDiJu06WbbKEUnKPgVrGj%2BqQAjjDIpUyYQI%3D&outcome=gotoDocu
mentUnitaire&docId=6876&flowCode=CR</getResultLinkFromAirsRequestReturn>

<getResultLinkFromAirsRequestReturn>faces/redirect.jsp?authentication=oiJI5WnyFI4J52ilix4nHIb
dg6HkWkzLx6f14ZOb1y5mvCmD8jXxsfs98cgwFgdkWufxFSJIjRfgGEzJ6BsX%2FDIt0ij01hMlj2kG9uiJPgAvelj%2B
JaQE8xRj8KrfvDe%2FjpnDVXIbmWDe52BjTwBsBG8TNWBXuSnKaCdMEpqaHo%3D&outcome=gotoDocumentUnit
aire&docId=6877&flowCode=CR</getResultLinkFromAirsRequestReturn>

<getResultLinkFromAirsRequestReturn>faces/redirect.jsp?authentication=RJJBFKvToNh6m2Tlfp5w6UK
4yea7qJDYCFgCgHQXNgpFMB5kU0qc0aqjnruKFntbhoJd07gw8mIUmd40ASRbyXQn2PoFm%2F7Nhx%2FrjuaDABKoxiFv1
brntfRSwCdiHwVmlgIsEsEynsawS%2FUcfk0guLCMV9LZOEXgm4ez5jsf01KI%3D&outcome=gotoDocumentUnit
aire&docId=6878&flowCode=CR</getResultLinkFromAirsRequestReturn>

<getResultLinkFromAirsRequestReturn>faces/redirect.jsp?authentication=aZ13IimjBYgwJHUF%2FLS1
ymH7%2FsnsPnjKYdxD55MYB0h77ZGRDoaqod0KpcC38ipIjFvlg3lnCVLS8i5FTdz9gCI%2BWZcOGIHRostjyAYmAYyA9
kXeBHNWd8aebtr8oKgL4xXfcPe2efuctL53d1LDJfg8rBWQDSNpxrNIHcbmU%3D&outcome=gotoDocumentUnit
aire&docId=6879&flowCode=CR</getResultLinkFromAirsRequestReturn>

```

```
<getResultLinkFromAirsRequestReturn>faces/redirect.jsp?authentication=FIW%2B6z1f6RDxy2GPmWnvs
7FtuA6rMaTImS2L%2BHt1d7k2QGWIPsM3cMQiTxFd2k27acB7cE%2Bb2qYrNQS7ZyP5kSwe000XtxlQm4Dvp8iSzmE3Ba
XNJscwo0iNcTXLO%2B%2BX2LqltAiCWNh%2FpfqGtMruu%2FyjQecEQZnBpQrFu3htoXk%3D&outcome=gotoDocu
mentUnitaire&docId=6880&flowCode=CR</getResultLinkFromAirsRequestReturn>
</getResultLinkFromAirsRequestResponse>
</soap:Body>
</soap:Envelope>
```



Les URL doivent être ajoutés à l'adresse <http://<host>:<port>/<context>> pour pouvoir être utilisées.

getResultsFromAirsRequest

Cette méthode permet de récupérer la liste des noms des bannettes à partir d'un code identifiant de flux (flowCode).

Table 87. Paramètres en entrée

Paramètre	Description	Requis	Exemple
connectionData	Objet contenant les informations de connexion	Oui	<pre><web:connectionData> <type:login>login</type:login> <type:password>password</type:password> </web:connectionData></pre>
ctyCodeList	Liste des types de contenu dans lesquels la recherche est exécutée	Oui	CR
flowCode	Code du flux dans lequel la recherche est effectué	Non	CR

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure connectionData.

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web=
"http://webservices.dossier.digitech.com" xmlns:type=
"http://type.webservices.dossier.digitech.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getResultsFromAirsRequest>
      <web:connectionData>
        <type:login>login</type:login>
        <type:password>password</type:password>
      </web:connectionData>
      <web:airsRequest>D_CREAT>25/09/2022</web:airsRequest>
      <web:ctyCode>CR</web:ctyCode>
      <!--Optional:-->
      <web:flowCode>CR</web:flowCode>
    </web:getResultsFromAirsRequest>
```

```
</soapenv:Body>
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getResultsFromAirsRequestResponse xmlns:types="
http://type.webservices.dossier.digitech.com" xmlns="
http://webservices.dossier.digitech.com">
      <getResultsFromAirsRequest>
        <refAirsId>6874</refAirsId>
        <fields>
          <code>D_MODIF</code>
          <field>27/09/2022 7:26:57</field>
        </fields>
        <fields>
          <code>CR_THEME</code>
          <field/>
        </fields>
        <fields>
          <code>D_CREAT</code>
          <field>27/09/2022 7:26:57</field>
        </fields>
        <fields>
          <code>CR_DES</code>
          <field/>
        </fields>
        <fields>
          <code>CR_REDACTEUR</code>
          <field/>
        </fields>
        <fields>
          <code>CR_DATE</code>
          <field>27/09/2022 0:00:00</field>
        </fields>
        <fields>
          <code>CR_RESUME</code>
          <field>file1 (UnitTest: 2022-09-27T07:27:36)</field>
        </fields>
        <fields>
          <code>T_PRIOR</code>
          <field/>
        </fields>
        <fields>
          <code>MULTI</code>
          <field/>
        </fields>
        <fields>
          <code>CORRES</code>
```

```

        <field/>
    </fields>
    <fields>
        <code>MASQUE_NUM</code>
        <field/>
    </fields>
</getResultsFromAirsRequest>
<getResultsFromAirsRequest>
    <refAirsId>6875</refAirsId>
    <fields>
        <code>D_MODIF</code>
        <field>27/09/2022 7:27:01</field>
    </fields>
    <fields>
        <code>CR_THEME</code>
        <field/>
    </fields>
    <fields>
        <code>D_CREAT</code>
        <field>27/09/2022 7:27:01</field>
    </fields>
    <fields>
        <code>CR_DES</code>
        <field/>
    </fields>
    <fields>
        <code>CR_REDACTEUR</code>
        <field/>
    </fields>
    <fields>
        <code>CR_DATE</code>
        <field>27/09/2022 0:00:00</field>
    </fields>
    <fields>
        <code>CR_RESUME</code>
        <field>file2 (UnitTest: 2022-09-27T07:27:36)</field>
    </fields>
    <fields>
        <code>T_PRIOR</code>
        <field/>
    </fields>
    <fields>
        <code>MULTI</code>
        <field/>
    </fields>
    <fields>
        <code>CORRES</code>
        <field/>
    </fields>
    <fields>
        <code>MASQUE_NUM</code>

```

```

        <field/>
    </fields>
    </getResultsFromAirsRequest>
</getResultsFromAirsRequestResponse>
</soap:Body>
</soap:Envelope>

```

cf le chapitre [Section 4.4, “Search modèles”](#) pour une description de la structure de retour.

3.5. Attachment

La **wsdl** est accessible à l’adresse suivante :

```
http(s)://<hostname>:<port>/<context>/services/WSAttachment?wsdl
```

L’encodage des requêtes et des réponses est **UTF-8**.

3.5.1. Client

La classe `com.digitech.dossier.ws.IAttachmentWSClient` de la librairie **com.digitech.air:dossier-ws-stubs:1.3.0** abstrait l’utilisation de cette **API**.

3.5.2. Méthodes

addOrUpdate

Cette méthode permet de créer un document

Table 88. Paramètres en entrée

Paramètre	Description	Requis	Exemple
connectionData	Objet contenant les informations de connexion	Oui	<pre> <web:connectionData> <type:login>login</type:login> <type:password>password</type:password> </web:connectionData> </pre>
docId	Identifiant du document AIRS	Oui	5072

Paramètre	Description	Requis	Exemple
attachment Data	Ensemble de données de la pièce jointe	Oui	<pre> <web:attachmentData> <type:label>PJ de test</type:label> <type:fileName> PJ.pdf</type:fileName> <type:data>JVBERi0xLjUNCiW1tbW1D1...</type: data> </web:attachmentData> </pre>

cf le chapitre [Section 4.1](#), “modèles globaux” pour une description de la structure `connectionData`.

exemple de requête soap

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web=
"http://webservices.dossier.digitech.com" xmlns:type=
"http://type.webservices.dossier.digitech.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:addOrUpdate>
      <web:connectionData>
        <type:login>login</type:login>
        <type:password>password</type:password>
      </web:connectionData>
      <web:docId>5072</web:docId>
      <web:attachmentData>
        <type:label>PJ de test</type:label>
        <type:fileName>PJ.pdf</type:fileName>
        <type:data>JVBERi0xLjUNCiW1tbW1D1...</type:data>
      </web:attachmentData>
    </web:addOrUpdate>
  </soapenv:Body>
</soapenv:Envelope>

```

cf le chapitre [Section 4.7](#), “Attachment modèles” pour une description de la structure `attachmentData`.

Réponse soap

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <createResponse xmlns:types="http://type.webservices.dossier.digitech.com" xmlns=
"http://webservices.dossier.digitech.com">
      <createReturn>6734</createReturn>
    </createResponse>
  </soap:Body>
</soap:Envelope>

```

getAttachmentList

Cette méthode permet de récupérer liste des pièces jointes d'un document AIRS.

Table 89. Paramètres en entrée

Paramètre	Description	Requis	Exemple
connectionData	Objet contenant les informations de connexion	Oui	<pre><web:connectionData> <type:login>login</type:login> <type:password>password</type:password> </web:connectionData></pre>
docId	Identifiant du document	Oui	4959

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure connectionData.

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.dossier.digitel.com" xmlns:type="http://type.webservices.dossier.digitel.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getAttachmentList>
      <web:connectionData>
        <type:login>login</type:login>
        <type:password>password</type:password>
      </web:connectionData>
      <web:docId>5072</web:docId>
    </web:getAttachmentList>
  </soapenv:Body>
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getAttachmentListResponse xmlns:types="http://type.webservices.dossier.digitel.com" xmlns="http://webservices.dossier.digitel.com">
      <getAttachmentListReturn>
        <airsType>ORIGINAL</airsType>
        <fileName>viroflay7àé.pdf</fileName>
        <id>5220</id>
        <label>viroflay7àé.pdf</label>
      </getAttachmentListReturn>
      <getAttachmentListReturn>
        <airsType>ORIGINAL</airsType>
        <fileName>1.pdf</fileName>
        <id>6563</id>
        <label>1.pdf</label>
      </getAttachmentListReturn>
    </getAttachmentListResponse>
  </soap:Body>
</soap:Envelope>
```



```

    </getAttachmentListReturn>
    <getAttachmentListReturn>
      <airsType>ORIGINAL</airsType>
      <fileName>PJ.pdf</fileName>
      <id>6703</id>
      <label>PJ de test</label>
    </getAttachmentListReturn>
  </getAttachmentListResponse>
</soap:Body>
</soap:Envelope>

```

cf le chapitre [Section 4.7, “Attachment modèles”](#) pour une description de la structure de retour. (Se référer à WSAttachmentType).

getAttachment

Cette méthode permet de récupérer une pièce jointe à partir de son identifiant et de celui de son document AIRS.

Table 90. Paramètres en entrée

Paramètre	Description	Requis	Exemple
connection Data	Objet contenant les informations de connexion	Oui	<pre> <web:connectionData> <type:login>login</type:login> <type:password>password</type:password> </web:connectionData> </pre>
docId	Identifiant du document AIRS	Oui	4959
attachment Id	Identifiant de la pièce jointe	Oui	1684

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure connectionData.

exemple de requête soap

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web=
"http://webservices.dossier.digitech.com" xmlns:type=
"http://type.webservices.dossier.digitech.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getAttachement>
      <web:connectionData>
        <type:login>login</type:login>
        <type:password>password</type:password>
      </web:connectionData>
      <web:docId>5072</web:docId>
      <web:attachmentId>6563</web:attachmentId>
    </web:getAttachement>
  </soapenv:Body>

```

```
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getAttachementResponse xmlns:types="http://type.webservices.dossier.digitech.com"
xmlns="http://webservices.dossier.digitech.com">
      <getAttachementReturn>
        <types:contentType>CR</types:contentType>
        <types:fields>
          <item>
            <types:code>D_MODIF</types:code>
            <types:field>14/11/2022 11:33:29</types:field>
          </item>
          <item>
            <types:code>D_CREAT</types:code>
            <types:field>21/10/2021 15:52:21</types:field>
          </item>
          <item>
            <types:code>CR_DATE</types:code>
            <types:field>14/11/2022 0:00:00</types:field>
          </item>
          <item>
            <types:code>CR_RESUME</types:code>
            <types:field>new CR updated (UnitTest: 2022-11-14T11:33:00)</types:field>
          </item>
        </types:fields>
        <types:secretLevel>10</types:secretLevel>
        <types:filename>1.pdf</types:filename>
        <types:data>JVBERi0xLjcN...</types:data>
      </getAttachementReturn>
    </getAttachementResponse>
  </soap:Body>
</soap:Envelope>
```

cf le chapitre [Section 4.2, “Document modèles”](#) pour une description de la structure de retour. (Se référer à documentData).

delete

Cette méthode permet de supprimer une pièce jointe d’un document AIRS.

Table 91. Paramètres en entrée

Paramètre	Description	Requis	Exemple
connectionData	Objet contenant les informations de connexion	Oui	<pre><web:connectionData> <type:login>login</type:login> <type:password>password</type:password> </web:connectionData></pre>
docId	Identifiant du document AIRS	Oui	4959
attachmentId	Identifiant de la pièce jointe	Oui	1684

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure connectionData.

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.dossier.digitech.com" xmlns:type="http://type.webservices.dossier.digitech.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:delete>
      <web:connectionData>
        <type:login>login</type:login>
        <type:password>password</type:password>
      </web:connectionData>
      <web:docId>5072</web:docId>
      <web:attachmentId>6563</web:attachmentId>
    </web:delete>
  </soapenv:Body>
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <deleteResponse xmlns:types="http://type.webservices.dossier.digitech.com" xmlns="http://webservices.dossier.digitech.com">
      <deleteReturn>6563</deleteReturn>
    </deleteResponse>
  </soap:Body>
</soap:Envelope>
```

3.6. DossierContext

La **wsdl** est accessible à l'adresse suivante :

```
http(s)://<hostname>:<port>/<context>/services/WSDossierContext?wsdl
```

L'encodage des requêtes et des réponses est **UTF-8**.

3.6.1. Client

La classe `com.digitech.dossier.ws.IDossierContextWSCClient` de la librairie `com.digitech.airs:dossier-ws-stubs:1.3.0` abstrait l'utilisation de cette API.

3.6.2. Méthodes

getFlows

Cette méthode renvoie la liste des flux paramétrés au sein de l'application.

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.dossier.digitech.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getFlows/>
  </soapenv:Body>
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <dos:getFlowsResponse xmlns:types="http://type.webservices.dossier.digitech.com"
xmlns:dos="http://webservices.dossier.digitech.com">
      <getFlowsReturn>
        <code>DOCUMENTATION</code>
        <label>Flow created in simplified mode</label>
      </getFlowsReturn>
      <getFlowsReturn>
        <code>REF</code>
        <label>Flow created in simplified mode</label>
      </getFlowsReturn>
      <getFlowsReturn>
        <code>CPA</code>
        <label>Flow created in simplified mode</label>
      </getFlowsReturn>
      <getFlowsReturn>
        <code>PRO</code>
        <label>Flux Projet</label>
      </getFlowsReturn>
      <getFlowsReturn>
        <code>CR</code>
        <label>Flow created in simplified mode</label>
      </getFlowsReturn>
    </getFlowsResponse>
  </soap:Body>
</soap:Envelope>
```

```

        </getFlowsReturn>
    </dos:getFlowsResponse>
</soap:Body>
</soap:Envelope>

```

getContentLinks

Cette méthode permet de récupérer la liste des informations d'un type de contenu.

exemple de requête soap

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web=
"http://webservices.dossier.digitech.com">
    <soapenv:Header/>
    <soapenv:Body>
        <web:getContentLinks>
            <web:contentTypeCode>DOC_CATEGORIE</web:contentTypeCode>
            <!--Optional:-->
            <web:flowCode>DOCUMENTATION</web:flowCode>
        </web:getContentLinks>
    </soapenv:Body>
</soapenv:Envelope>

```

Réponse soap

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <dos:getContentLinksResponse xmlns:types=
"http://type.webservices.dossier.digitech.com" xmlns:dos=
"http://webservices.dossier.digitech.com">
            <getContentLinksReturn>
                <code>DOC_CATEGORIE</code>
                <id>15</id>
                <description>Catégorie</description>
                <linksInfosList>
                    <linksInfosList>
                        <dos:childContentType>
                            <code>DOC_DOCUMENT</code>
                            <id>16</id>
                            <description>Document</description>
                            <linksInfosList/>
                        </dos:childContentType>
                        <dos:linkedFieldInfos>
                            <dos:linkedFieldInfos>
                                <dos:fieldChildcode>DOC_CAT_CODE</dos:fieldChildcode>
                                <dos:fieldFathercode>DOC_CAT_CODE</dos:fieldFathercode>
                            </dos:linkedFieldInfos>
                        </dos:linkedFieldInfos>
                    </linksInfosList>
                </linksInfosList>
            </getContentLinksReturn>
        </dos:getContentLinksResponse>
    </soap:Body>
</soap:Envelope>

```

```

</dos:getContentTypeLinksResponse>
</soap:Body>
</soap:Envelope>

```

cf le chapitre [Section 4.5, “DossierContext modèles”](#) pour une description de la structure de retour.

getMandatoryField

Cette méthode permet de récupérer la liste des champs obligatoires d’un type de contenu.

exemple de requête soap

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web=
"http://webservices.dossier.digitel.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getMandatoryField>
      <web:contentTypeCode>CR</web:contentTypeCode>
      <!--Optional:-->
      <web:flowCode>CR</web:flowCode>
    </web:getMandatoryField>
  </soapenv:Body>
</soapenv:Envelope>

```

Réponse soap

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <dos:getMandatoryFieldResponse xmlns:types=
"http://type.webservices.dossier.digitel.com" xmlns:dos=
"http://webservices.dossier.digitel.com">
      <getMandatoryFieldReturn>CR_THEME</getMandatoryFieldReturn>
      <getMandatoryFieldReturn>CR_DES</getMandatoryFieldReturn>
      <getMandatoryFieldReturn>CR_REDACTEUR</getMandatoryFieldReturn>
      <getMandatoryFieldReturn>CR_DATE</getMandatoryFieldReturn>
    </dos:getMandatoryFieldResponse>
  </soap:Body>
</soap:Envelope>

```

3.7. ExtendDocument

La **wsdl** est accessible à l’adresse suivante :

```
http(s)://<hostname>:<port>/<context>/services/extendDocument?wsdl
```

L’encodage des requêtes et des réponses est **UTF-8**.

3.7.1. Client

La classe `com.digitech.dossier.ws.IExtendDocumentWSClient` de la librairie `com.digitech.airs:dossier-ws-stubs:1.3.0` abstrait l'utilisation de cette API.

3.7.2. Méthodes

deleteDocuments

Cette méthode permet de supprimer des documents par lot.

Table 92. Paramètres en entrée

Paramètre	Description	Requis	Exemple
arg0	Objet contenant les informations de connexion.	Oui	<pre><arg0> <login>login</login> <password>password</password> </arg0></pre>
arg1	Identifiant AIRS du document a supprimer. Ce paramètre peut apparaître plusieurs fois.	Oui	<pre><arg1>157</arg1></pre>

cf le chapitre [Section 4.1, "modèles globaux"](#) pour une description de la structure `arg0` (voir `connectionData`).

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.dossier.digitech.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:deleteDocuments>
      <arg0>
        <login>login</login>
        <password>password</password>
      </arg0>
      <!--1 or more repetitions:-->
      <arg1>1</arg1>
      <arg1>2</arg1>
    </web:deleteDocuments>
  </soapenv:Body>
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <dos:deleteDocumentsResponse xmlns:types="http://type.webservices.dossier.digitech.com">
```

```

xmlns:dos="http://webservices.dossier.digitel.com/" xmlns:ns4=
"http://webservices.dossier.digitel.com">
  <dos:return>
    <ns4:docId>1</ns4:docId>
    <ns4:state>OK</ns4:state>
    <ns4:message></ns4:message>
    <ns4:children/>
  </dos:return>
  <dos:return>
    <ns4:docId>2</ns4:docId>
    <ns4:state>ERROR</ns4:state>
    <ns4:message>com.digitel.jcorbairs.exception.ServerException: Le document de
référence 2 n'existe pas</ns4:message>
    <ns4:children/>
  </dos:return>
</dos:deleteDocumentsResponse>
</soap:Body>
</soap:Envelope>

```

cf le chapitre [Section 4.6, “ExtendDocument modèles”](#) pour une description de la structure de retour (voir WSDelateReportItemType).

deleteDocumentsType

Cette méthode permet de supprimer des documents par lot en spécifiant le type de suppression.

Table 93. Paramètres en entrée

Paramètre	Description	Requis	Exemple
arg0	Objet contenant les informations de connexion.	Oui	<pre> <arg0> <login>login</login> <password>password</password> </arg0> </pre>
arg1	Identifiant AIRS du document à supprimer. Ce paramètre peut apparaître plusieurs fois.	Oui	<pre> <arg1>157</arg1> </pre>
deleteType	Type de suppression. Valeurs possibles : DOC_REMOVE_TYPE_DO C_ONLY DOC_REMOVE_TYPE_CHE CK_PARENT DOC_REMOVE_TYPE_FOR CE	Oui	DOC_REMOVE_TYPE_FORCE

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure arg0 (voir connectionData).

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web=
"http://webservices.dossier.digitel.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:deleteDocumentsType>
      <arg0>
        <login>n.felix</login>
        <password>password</password>
      </arg0>
      <!--1 or more repetitions:-->
      <arg1>1</arg1>
      <arg1>2</arg1>
      <deleteType>DOC_REMOVE_TYPE_FORCE</deleteType>
    </web:deleteDocumentsType>
  </soapenv:Body>
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <dos:deleteDocumentsTypeResponse xmlns:types=
"http://type.webservices.dossier.digitel.com" xmlns:dos=
"http://webservices.dossier.digitel.com/" xmlns:ns4="
http://webservices.dossier.digitel.com">
      <dos:return>
        <ns4:docId>1</ns4:docId>
        <ns4:state>OK</ns4:state>
        <ns4:message></ns4:message>
        <ns4:children/>
      </dos:return>
      <dos:return>
        <ns4:docId>2</ns4:docId>
        <ns4:state>ERROR</ns4:state>
        <ns4:message>com.digitel.jcorbairs.exception.ServerException: Le document de
référence 2 n'existe pas</ns4:message>
        <ns4:children/>
      </dos:return>
    </dos:deleteDocumentsTypeResponse>
  </soap:Body>
</soap:Envelope>
```

cf le chapitre [Section 4.6, “ExtendDocument modèles”](#) pour une description de la structure de retour (voir WSDelateReportItemType).

getAttachementByDocId

Cette méthode permet de récupérer la première pièce jointe d’un document ainsi que la liste des

identifiants des autres eventuelles pièces jointes.

Table 94. Paramètres en entrée

Paramètre	Description	Requis	Exemple
arg0	Objet contenant les informations de connexion.	Oui	<pre><arg0> <login>login</login> <password>password</password> </arg0></pre>
arg1	Identifiant du document	Oui	4959

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure arg0 (voir `connectionData`).

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web=
"http://webservices.dossier.digitech.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getAttachementByDocId>
      <arg0>
        <login>login</login>
        <password>password</password>
      </arg0>
      <arg1>5028</arg1>
    </web:getAttachementByDocId>
  </soapenv:Body>
</soapenv:Envelope>
```

Réponse soap

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <dos:getAttachementByDocIdResponse xmlns:types=
"http://type.webservices.dossier.digitech.com" xmlns:dos=
"http://webservices.dossier.digitech.com/" xmlns:ns4="
http://webservices.dossier.digitech.com">
      <dos:return>
        <ns4:firstAttachment>
          <ns4:id>5189</ns4:id>
          <ns4:airsType>ORIGINAL</ns4:airsType>
          <ns4:label>viroflay7àé.pdf</ns4:label>

        <ns4:data>JVBERi0xLjMNJeLjz9MNCjEgMCBvYmoNPdWvTWV0YWRhdGEgMjcgcMCBSL1BhZ2VzIDIgMCBSL1R5cGUvQ2F
0YWxvZz4...</ns4:data>
          <ns4:fileName>viroflay7àé.pdf</ns4:fileName>
        </ns4:firstAttachment>
      </dos:return>
    </dos:getAttachementByDocIdResponse>
  </soap:Body>
</soap:Envelope>
```

```

</dos:return>
</dos:getAttachementByDocIdResponse>
</soap:Body>
</soap:Envelope>

```

cf le chapitre [Section 4.2, “Document modèles”](#) pour une description de la structure de retour (voir WSAttachmentListType).

insertDocuments

Cette méthode permet créer des documents par lot.

Table 95. Paramètres en entrée

Paramètre	Description	Requis	Exemple
arg0	Objet contenant les informations de connexion	Oui	<pre> <arg0> <login>login</login> <password>password</password> </arg0> </pre>
arg1	Liste de WSInsertDocumentType. Ce paramètre peut apparaître plusieurs fois.	Oui	<pre> <arg1> <currentDocument> <contentType>CR</contentType> <secretLevel>10</secretLevel> <fields> <!--Zero or more repetitions:--> <field> <code>D_MODIF</code> <field>04/02/2021 16:58:35</field> </field> <field> <code>CR_DES</code> <field>Test DOC</field> </field> </fields> <data>cid:373014430535</data> <flowCode>CR</flowCode> </currentDocument> <doUpdateParent>>false</doUpdateParent> </arg1> </pre>

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure arg0 (voir connectionData).

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure WSInsertDocumentType.

exemple de requête soap

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web=
"http://webservices.dossier.digitech.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:insertDocuments>
      <arg0>
        <login>login</login>
        <password>password</password>
      </arg0>
      <!--1 or more repetitions:-->
      <arg1>
        <currentDocument>
          <contentType>CR</contentType>
          <secretLevel>10</secretLevel>
          <fields>
            <!--Zero or more repetitions:-->
            <field>
              <code>D_MODIF</code>
              <field>04/02/2021 16:58:35</field>
            </field>
            <field>
              <code>CR_DES</code>
              <field>Test DOC</field>
            </field>
          </fields>
          <data>cid:373014430535</data>
          <flowCode>CR</flowCode>
        </currentDocument>
        <doUpdateParent>false</doUpdateParent>
      </arg1>
    </web:insertDocuments>
  </soapenv:Body>
</soapenv:Envelope>

```

Réponse soap

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <dos:insertDocumentsResponse xmlns:types="http://type.webservices.dossier.digitech.com"
xmlns:dos="http://webservices.dossier.digitech.com/" xmlns:ns4=
"http://webservices.dossier.digitech.com">
      <dos:return>
        <ns4:docId>6976</ns4:docId>
        <ns4:state>ERROR</ns4:state>
        <ns4:message>Erreur</ns4:message>
      </dos:return>
    </dos:insertDocumentsResponse>
  </soap:Body>

```

```
</soap:Envelope>
```

cf le chapitre [Section 4.6, “ExtendDocument modèles”](#) pour une description de la structure de retour (voir WSInsertReportItemType).

updateDocuments

Cette méthode permet de mettre à jour des documents par lot.

Table 96. Paramètres en entrée

Paramètre	Description	Requis	Exemple
arg0	Objet contenant les informations de connexion	Oui	<pre><arg0> <login>login</login> <password>password</password> </arg0></pre>
arg1	Liste de WSExtendDocumentType . Ce paramètre peut apparaître plusieurs fois.	Oui	<pre><arg1> <contentType>CR</contentType> <secretLevel>10</secretLevel> <fields> <!--Zero or more repetitions:--> <field> <code>D_MODIF</code> <field>04/02/2021 16:58:35</field> </field> <field> <code>CR_DES</code> <field>Test DOC</field> </field> </fields> <flowCode>CR</flowCode> <web1:docId>5028</web1:docId> </arg1></pre>

cf le chapitre [Section 4.1, “modèles globaux”](#) pour une description de la structure arg0 (voir connectionData).

cf le chapitre [Section 4.6, “ExtendDocument modèles”](#) pour une description de la structure WSExtendDocumentType.

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web=
"http://webservices.dossier.digitel.com/" xmlns:web1=
"http://webservices.dossier.digitel.com">
  <soapenv:Header/>
  <soapenv:Body>
    <web:updateDocuments>
```

```

<arg0>
  <login>login</login>
  <password>password</password>
</arg0>
<!--1 or more repetitions:-->
<arg1>
  <contentType>CR</contentType>
  <secretLevel>10</secretLevel>
  <fields>
    <!--Zero or more repetitions:-->
    <field>
      <code>D_MODIF</code>
      <field>04/02/2021 16:58:35</field>
    </field>
    <field>
      <code>CR_DES</code>
      <field>Test DOC</field>
    </field>
  </fields>
  <flowCode>CR</flowCode>
  <web1:docId>5028</web1:docId>
</arg1>
</web:updateDocuments>
</soapenv:Body>
</soapenv:Envelope>

```

Réponse soap

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <dos:updateDocumentsResponse xmlns:types="http://type.webservices.dossier.digitech.com"
xmlns:dos="http://webservices.dossier.digitech.com/" xmlns:ns4=
"http://webservices.dossier.digitech.com">
      <dos:return>
        <ns4:docId>5028</ns4:docId>
        <ns4:state>OK</ns4:state>
        <ns4:message/>
      </dos:return>
    </dos:updateDocumentsResponse>
  </soap:Body>
</soap:Envelope>

```

cf le chapitre [Section 4.6, “ExtendDocument modèles”](#) pour une description de la structure de retour (voir WUpdateReportItemType).

3.8. ExtendSearch



Ce Web Service n'est disponible qu'à partir de la version 7.1 de AIRS Dossier

La **wsdl** est accessible à l'adresse suivante :

```
http(s)://<hostname>:<port>/<context>/services/extendSearch?wsdl
```

L'encodage des requêtes et des réponses est **UTF-8**.

3.8.1. Client

La classe `com.digitech.dossier.ws.IExtendSearchWSClient` de la librairie `com.digitech.airs:dossier-ws-stubs:1.3.0` abstrait l'utilisation de cette **API**.

3.8.2. Authentification

Ce **Web Service** requiert une **authentification basique**.

3.8.3. Méthodes

search4FieldsAndAttachments

Cette méthode permet à partir d'une requête **Airs** de récupérer

- des documents
- leurs *metadata* (champs)
- et les informations sur les pièces jointes de chaque document

Table 97. Paramètres en entrée

Paramètre	Description	Requis	Exemple
airsReques t	Requête Airs	Oui	D_CREAT >= 28/03/2020
ctyCodes	Liste des types de contenu dans lesquels la recherche est exécutée	Oui	CR
flowCode	Code du flux dans lequel la recherche est effectuée. Si Vide, celui-ci sera déduit de la liste des types de contenu	Oui	CR
continuati onToken	Clé de <i>continuation</i> . Le nombre de résultats reçus est limité à 20 (par défaut, côté AIRS Dossier . Si vous souhaitez plus de résultats, le Web Service doit être rappelé en ajoutant cette clé.	Non	OTU4ODQ0ZTFhMDIlOD ViYTNjN2JiNThkOWM5Yj QwNDAjIzE1OTI5MTc3M TM2MDYjIzQ1ODYjIzIw

exemple de requête soap

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web=
"http://webservices.dossier.digitech.com">
  <soapenv:Header/>
  <soapenv:Body>
```

```

<web:search4FieldsAndAttachments>
  <web:airsRequest>D_CREAT >= 28/03/2020</web:airsRequest>
  <!--1 or more repetitions:-->
  <web:ctyCodes>CR</web:ctyCodes>
  <web:ctyCodes>PRO_CLIENT</web:ctyCodes>
  <!--Optional:-->
  <web:flowCode/>
  <!--Optional:-->
  <web:continuationToken/>
</web:search4FieldsAndAttachments>
</soapenv:Body>
</soapenv:Envelope>

```

Réponse soap

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <dos:search4FieldsAndAttachmentsResponse
      xmlns:type="http://type.webservices.dossier.digitel.com"
      xmlns:dos="http://webservices.dossier.digitel.com">
      <dos:search4FieldsAndAttachments>
        <dos:resultNumber>35</dos:resultNumber>
        <dos:documents>
          <dos:document>
            <dos:refAirs>4539</dos:refAirs>
            <dos:secretLevel>10</dos:secretLevel>
            <dos:contentType>CR</dos:contentType>
            <dos:fields>
              <type:field>
                <code>D_MODIF</code>
                <label>Date de modification</label>
                <value>22/06/2020 15:38:10</value>
              </type:field>
              <type:field>
                <code>CR_THEME</code>
                <label>Thématique</label>
                <value>Réunion Transverse</value>
                <refValue>172</refValue>
              </type:field>
              ...
            </dos:fields>
            <dos:attachments>
              <type:attachment>
                <id>4254</id>
                <label>20_euros</label>
                <fileName>20_euros.PDF</fileName>
                <airsType>ORIGINAL</airsType>
              </type:attachment>
              <type:attachment>
                <id>4257</id>

```



```

<label>test_compat</label>
<fileName>test_compat-1.odt</fileName>
<airsType>ORIGINAL</airsType>
<type:versions>
  <type:version>
    <id>4255</id>
    <label>test_compat</label>
    <fileName>test_compat.odt</fileName>
    <version>1</version>
    <date>22/08/2022 15:32:05</date>
  </type:version>
  ...
</type:versions>
</type:attachment>
...
</dos:attachments>
</dos:document>
...
</dos:documents>
</dos:search4FieldsAndAttachments>
</dos:search4FieldsAndAttachmentsResponse>
</soap:Body>
</soap:Envelope>

```

cf le chapitre [Section 4.8, “ExtendedSearch modèles”](#) pour une description de la structure de retour.

4. Annexes

4.1. modèles globaux

connectionData (SOAP)

Table 98. Paramètres en entrée

Paramètre	Description	Observations	Type
login	login utilisateur		string
password	mot de passe		string
authenticationId	le paramètre d'authentification encodé en base 64		string

4.2. Document modèles

La réponse est retournée sous forme de chaîne de caractères, au format **XML** (**soap**) ou **json** (**rest**)

Celle-ci contient les informations suivantes :

WSDocumentType (REST)

Table 99. Paramètres en sortie

Paramètre	Description	Observations	Type
contentType	type de contenu du document		string
secretLevel	Niveau de secret		integer
data	données du document principal en bytes		string
filename	nom du fichier		string
flowCode	code du flux		string
scriptCode	code du script		string
fields > field			
code	code du champ		string
field	valeur du champ		integer

WSCommonDocumentType (REST)

Table 100. Paramètres en sortie

Paramètre	Description	Observations	Type
contentType	type de contenu du document		string
secretLevel	Niveau de secret		integer

Paramètre	Description	Observations	Type
flowCode	code du flux		<i>string</i>
scriptCode	code du script		<i>string</i>
fields > field			
code	code du champ		<i>string</i>
field	valeur du champ		<i>integer</i>

documentData (SOAP)

Table 101. Paramètres en sortie

Paramètre	Description	Observations	Type
contentType	type de contenu du document		<i>string</i>
secretLevel	Niveau de secret		<i>integer</i>
data	données du document principal en bytes		<i>string</i>
filename	nom du fichier		<i>string</i>
flowCode	code du flux		<i>string</i>
scriptCode	code du script		<i>string</i>
fields > item			
code	code du champ		<i>string</i>
field	valeur du champ		<i>integer</i>

4.3. Authority List modèles

La réponse est retournée sous forme de chaîne de caractères, au format **XML** (**soap**) ou **json** (**rest**)

Celle-ci contient les informations suivantes :

WSTerm

Table 102. Paramètres en sortie

Paramètre	Description	Observations	Type
id	identifiant du terme		<i>string</i>
code	code du terme		<i>integer</i>
label	libellé du terme		<i>string</i>

4.4. Search modèles

La réponse est retournée sous forme de chaîne de caractères, au format **XML** (**soap**) ou **json** (**rest**)

Celle-ci contient les informations suivantes :

WSSearchResultType

Table 103. Paramètres en sortie

Paramètre	Description	Observations	Type
refAirsId	Identifiant AIRS du document		<i>integer</i>
fields			
code	code du champ		<i>string</i>
field	valeur du champ		<i>string</i>

4.5. DossierContext modèles

La réponse est retournée sous forme de chaîne de caractères, au format XML (soap) ou json (rest)

Celle-ci contient les informations suivantes :

Liste de flux

Table 104. Paramètres en sortie

Paramètre	Description	Observations	Type
code	Code du flux		<i>string</i>
label	Nom du flux		<i>string</i>

WSContentTypeInfosType (REST)

Table 105. Paramètres en sortie

Paramètre	Description	Observations	Type
code	Code du type de contenu		<i>string</i>
id	Identifiant AIRS du type de contenu		<i>integer</i>
description	Nom du type de contenu		<i>string</i>
linksInfosList > childContentType			
code	Code du type de contenu fils		<i>string</i>
id	Identifiant AIRS du type de contenu fils		<i>integer</i>
description	Nom du type de contenu fils		<i>string</i>
linksInfosList	Informations du type de contenu fils	Contient une liste vide si le type de contenu fils n'a pas d'enfant	strucutre <i>linksInfos List</i>

linksInfosList > linkedFieldInfos

Paramètre	Description	Observations	Type
fieldChildcode	Code du champ utilisé par le type de contenu fils pour la liaison père-fils		<i>string</i>
fieldFathercode	Code du champ utilisé par le type de contenu père pour la liaison père-fils		<i>string</i>

4.6. ExtendDocument modèles

sample **POST** sortie

```
[
  {
    "contentType": "CR",
    "fields": [
      {
        "code": "CR_RESUME",
        "field": "TEST JMU 4"
      }
    ],
    "flowCode": "CR",
    "docId": 6935
  },
  {
    "contentType": "CR",
    "fields": [
      {
        "code": "CR_RESUME",
        "field": "TEST JMU 5"
      }
    ],
    "flowCode": "CR",
    "docId": 6936
  }
]
```

sample **PATCH** sortie

```
[
  {
    "currentDocument": {
      "contentType": "CR",
      "secretLevel": 10,
      "fields": [
        {
          "code": "D_MODIF",
          "field": "04/02/2021 16:58:35"
        }
      ]
    }
  }
]
```

```

    },
    {
      "code": "D_CREAT",
      "field": "04/04/2019 18:40:22"
    },
    {
      "code": "CR_DES",
      "field": "Test JMU1"
    },
    {
      "code": "CR_REDACTEUR",
      "field": "1"
    },
    {
      "code": "CR_DATE",
      "field": "01/01/2022 0:00:00"
    },
    {
      "code": "CR_RESUME",
      "field": "TEST JMU1"
    }
  ]
},
"doUpdateParent": false
},
{
  "currentDocument": {
    "contentType": "CR",
    "secretLevel": 10,
    "fields": [
      {
        "code": "D_MODIF",
        "field": "04/02/2021 16:58:35"
      },
      {
        "code": "D_CREAT",
        "field": "04/04/2019 18:40:22"
      },
      {
        "code": "CR_DES",
        "field": "Test JMU2"
      },
      {
        "code": "CR_REDACTEUR",
        "field": "1"
      },
      {
        "code": "CR_DATE",
        "field": "01/01/2022 0:00:00"
      },
    ]
  }
}

```

```

    "code": "CR_RESUME",
    "field": "TEST JMU2"
  }
]
},
"doUpdateParent": false
}
]
```

La réponse est retournée sous forme de chaîne de caractères, au format XML (soap) ou json (rest)

Celle-ci contient les informations suivantes :

WSDeleteReportItemType

Table 106. Paramètres en sortie

Paramètre	Description	Observations	Type
docId	Identifiant du document		integer
state	Etat de l'opération	OK ou ERROR	string
children	Liste des documents enfants supprimés	Liste d'identifiants	liste d'integer
errMessage	Message d'erreur	Vide si aucune erreur	string

WSAttachmentListType

Table 107. Paramètres en sortie

Paramètre	Description	Observations	Type
WSAttachmentListType > firstAttachment			
id	Identifiant AIRS de la pièce jointe		integer
airsType	Type de pièce jointe		string
label	Intitulé de la pièce jointe		string
fileName	Nom du fichier		string
data	Données brutes du fichier	Sous forme de bytes	string
WSAttachmentListType > items	Liste des identifiants des autres pièces jointes du document		liste d'integer

WSInsertDocumentType

Table 108. Paramètres en entrée

Paramètre	Description	Observations	Type
doUpdateParent	Nécessité de mettre à jour le document parent		boolean
WSInsertDocumentType > currentDocument			

Paramètre	Description	Observations	Type
contentType	Type de contenu du document		<i>string</i>
secretLevel	Niveau de secret		<i>integer</i>
data	Données brutes du document	En bytes	<i>string</i>
filename	Nom du fichier		<i>string</i>
flowCode	Code du flux		<i>string</i>
scriptCode	Nom du script a exécuter		<i>string</i>
filesIndex	Liste des index des pièces jointes à ajouter (voir FilesData). Le premier index correspond à 0.		<i>Liste d'integer</i>
fileName	Liste des noms de pièces jointes à ajouter. L'ordre doit être le même que dans la liste filesIndex		<i>Liste de string</i>

WSInsertDocumentType > currentDocument > fields (liste des champs)

code	code du champ	Liste d'identifiants	<i>string</i>
field	valeur du champ	Vide si aucune erreur	<i>string</i>

WSInsertDocumentType > parentDocument

contentType	Type de contenu du document		<i>string</i>
secretLevel	Niveau de secret		<i>integer</i>
data	Données brutes du document	En bytes	<i>string</i>
filename	Nom du fichier		<i>string</i>
flowCode	Code du flux		<i>string</i>
scriptCode	Nom du script a exécuter		<i>string</i>
filesIndex	Liste des index des pièces jointes à ajouter (voir FilesData). Le premier index correspond à 0.		<i>Liste d'integer</i>
fileName	Liste des noms de pièces jointes à ajouter. L'ordre doit être le même que dans la liste filesIndex		<i>Liste de string</i>

WSInsertDocumentType > parentDocument > fields (liste des champs)

code	code du champ	Liste d'identifiants	<i>string</i>
field	valeur du champ	Vide si aucune erreur	<i>string</i>

WSInsertReportItemType

Table 109. Paramètres en sortie

Paramètre	Description	Observations	Type
docId	Identifiant du document		<i>integer</i>

Paramètre	Description	Observations	Type
state	Etat de l'opération	OK ou ERROR	<i>string</i>
parentState	Etat du parent		<i>string</i>
message	Message d'erreur	Vide si aucune erreur	<i>string</i>

WSExtendDocumentType

Table 110. Paramètres en entrée

Paramètre	Description	Observations	Type
docId	Identifiant du document		<i>integer</i>
contentType	Type de contenu du document		<i>string</i>
secretLevel	Niveau de secret		<i>integer</i>
data	Données brutes du document	En bytes	<i>string</i>
filename	Nom du fichier		<i>string</i>
flowCode	Code du flux		<i>string</i>
filesIndex	Liste des index des pièces jointes à ajouter (voir FilesData). Le premier index correspond à 0.		<i>Liste d'integer</i>
filesName	Liste des noms de pièces jointes à ajouter. L'ordre doit être le même que dans la liste filesIndex		<i>Liste de string</i>
scriptCode	Nom du script à exécuter		<i>string</i>
WSExtendDocumentType > fields (liste des champs)			
code	code du champ	Liste d'identifiants	<i>string</i>
field	valeur du champ	Vide si aucune erreur	<i>string</i>

WSUpdateReportItemType

Table 111. Paramètres en sortie

Paramètre	Description	Observations	Type
docId	Identifiant du document		<i>integer</i>
state	Etat de l'opération	OK ou ERROR	<i>string</i>
errMessage	Message d'erreur	Vide si aucune erreur	<i>string</i>

4.7. Attachment modèles

La réponse est retournée sous forme de chaîne de caractères, au format **XML** (soap) ou **json** (rest)

Celle-ci contient les informations suivantes :

WSAttachmentType

Table 112. Paramètres en sortie

Paramètre	Description	Observations	Type
airsType	Type de pièce jointe		<i>string</i>
filename	nom du fichier		<i>string</i>
id	identifiant de la pièce jointe		<i>integer</i>
label	titre de la pièce jointe		<i>string</i>

attachmentData (SOAP)

Table 113. Paramètres en sortie

Paramètre	Description	Observations	Type
airsType	type de la pièce jointe		<i>string</i>
id	identifiant de la pièce jointe à mettre à jour		<i>integer</i>
data	données du document principal en bytes		<i>string</i>
filename	nom du fichier		<i>string</i>
label	label de la pièce jointe		<i>string</i>
scriptCode	code du script		<i>string</i>

WSCommonAttachmentUpdateType (REST)

Paramètre	Description	Observations	Type
airsType	type de la pièce jointe		<i>string</i>
id	identifiant de la pièce jointe à mettre à jour		<i>integer</i>
label	label de la pièce jointe		<i>string</i>
scriptCode	code du script		<i>string</i>

4.8. ExtendedSearch modèles

La réponse est retournée sous forme de chaîne de caractères, au format **XML** (**soap**) ou **json** (**rest**)

Celle-ci contient les informations suivantes :

Table 114. Paramètres en sortie

Paramètre	Description	Observations	Type
resultNumber	Nombre total de résultats		<i>string</i>
documents > document			

Paramètre	Description	Observations	Type
refAirs	Référence Airs du document		<i>integer</i>
secretLevel	Niveau de secret		<i>integer</i>
contentType	Type de contenu		<i>string</i>
continuationToken	Clé de <i>continuation</i>	Absent si l'intégralité des documents a été retourné	<i>string</i>
documents > document > fields > field			
code	Code du champ		<i>string</i>
label	Libellé du champ		<i>string</i>
value	Valeur du champ		<i>string</i>
refValue	Identifiant du champ référence Airs	Nœud absent si le champ n'est pas un champ <i>référence</i>	<i>string</i>
documents > document > attachments > attachment			
id	Référence Airs de la pièce jointe		<i>integer</i>
label	Libellé de la pièce jointe		<i>string</i>
fileName	Nom de la pièce jointe		<i>string</i>
airsType	Type de pièce jointe		<i>string</i>
documents > document > attachments > attachment > versions > version			
id	Référence Airs de la version de la pièce jointe		<i>integer</i>
label	Libellé de la version de la pièce jointe		<i>string</i>
fileName	Nom de la version de la pièce jointe		<i>string</i>
version	Référence Airs de la version de la pièce jointe		<i>integer</i>
date	Date de création de cette version de la pièce jointe	Date, au format dd/MM/yyyy H:mm:ss	<i>string</i> / <i>date</i>

4.9. Workbook modèles

La réponse est retournée sous forme de chaîne de caractères, au format **json**

Celle-ci contient les informations suivantes :

WSWorkbookType

Table 115. Paramètres en sortie

Paramètre	Description	Observations	Type
id	identifiant du classeur		<i>integer</i>
name	nom du classeur		<i>string</i>
description	description du classeur		<i>string</i>
creatorId	identifiant du créateur du classeur		<i>integer</i>
workbookType	type du classeur		<i>string</i>
shareStartDate	date de début de partage du timestamp classeur		<i>integer</i>
shareEndDate	date de fin de partage du classeur	timestamp	<i>integer</i>
status	statut du classeur	DOSSIER_WB_STATUS_ACTIVE ou DOSSIER_WB_STATUS_CLOSED	<i>string</i>
share	état de partage du classeur	DOSSIER_WB_SHARE_SHARED ou DOSSIER_WB_SHARE_PRIVATE ou DOSSIER_WB_SHARE_PUBLIC	<i>string</i>
usersRead	liste d'utilisateurs avec accès en lecture seulement		<i>[]integer</i>
usersReadWrite	liste d'utilisateurs avec accès en lecture et écriture		<i>[]integer</i>
orgasRead	liste d'organisations avec accès en lecture seulement		<i>[]integer</i>
orgasReadWrite	liste d'organisations avec accès en lecture et écriture		<i>[]integer</i>
documents	liste d'identifiants des documents du classeur		<i>[]integer</i>