



Metasign-server

Description des interfaces REST



Version : 0.5
Date de document: 08/10/2019
Ref. Doc.: MSIGN-SRV-REST-01

Statut

Rédaction	KDA
Validation	[Responsable]
Classification	Publique
Etat du document	Temporaire
Version actuelle	0.5
Référence	MSIGN-SRV-REST-01
Version Produit applicable	2.4.0

Diffusion

Nom	Entreprise

Historique des révisions

Date	Version	Commentaires
28/02/2018	0.1	Création du document
19/03/2018	0.2	Relecture du document
23/10/2018	0.3	Mise à jour du document
19/07/19	0.4	Correction du document
08/10/19	0.5	Port incorrect dans les exemples

Sommaire

1	Introduction.....	6
1.1	Présentation du contexte.....	6
1.2	Références documentaires.....	6
1.3	Glossaire.....	7
2	Généralités	8
3	Authentification	9
4	Les méthodes de récupération d'informations	10
4.1	Récupération de l'ensemble des méthodes disponibles	10
4.2	Récupération des informations de l'utilisateur	11
5	Les méthodes pour les opérations de signature	14
5.1	Dépôt de document sur le serveur.....	14
5.2	Récupération de document sur le serveur.....	15
5.3	Signature d'un document	17
5.4	Augmentation d'une signature	19
5.5	Vérification d'une signature	21
5.6	Récupération des identifiants des signataires	23
6	Les méthodes pour les opérations d'administration pour la signature	25
6.1	Dépôt de politique de signature	26
6.2	Liste des politiques de signature	28
6.3	Suppression d'une politique de signature	29
6.4	Récupération d'une politique de signature	30
6.5	Dépôt d'un profil de clé de signature.....	32
6.6	Mise à jour d'un profil de clé de signature	34
6.7	Liste des profils de clé de signature	35
6.8	Suppression d'un profil de clé de signature.....	37
6.9	Récupération d'un profil de clé de signature.....	38
6.10	Dépôt d'un profil de signature.....	39
6.11	Mise à jour d'un profil de signature.....	41
6.12	Liste des profils de signature	43
6.13	Suppression d'un profil de clé de signature.....	44
6.14	Récupération d'un profil de signature.....	45
6.15	Dépôt d'un key store	47

6.16	Mise à jour d'un secret de signature	48
6.17	Dépôt d'un certificat pour une clé de signature	49
6.18	Requête d'un certificat pour une clé de signature	51
6.19	Génération d'une CSR pour une clé de signature	52
6.20	Suppression d'une clé de signature.....	54
6.21	Activation d'une clé de signature.....	55
6.22	Création d'une clé de signature à partir d'un profil de clé	56
6.23	Création d'une clé de signature à partir d'une paire de clés	58
6.24	Création d'une clé de signature à partir d'un PKCS12.....	59
7	Les méthodes pour les opérations d'administration	62
7.1	Création d'un utilisateur.....	62
7.2	Mise à jour d'un utilisateur	64
7.3	Suppression d'un utilisateur.....	66
7.4	Liste des utilisateurs.....	67
7.5	Consultation d'un utilisateur	71
7.6	Création d'un groupe	73
7.7	Ajout d'utilisateurs à un groupe	75
7.8	Suppression d'utilisateurs dans un groupe.....	76
7.9	Suppression d'un groupe.....	77
7.10	Enrôlement d'un utilisateur FIDO.....	79
7.11	Dépôt d'un token FIDO	80
7.12	Liste des tokens FIDO	82
7.13	Suppression d'un token FIDO.....	83
7.14	Dépôt d'un key store	84
8	Description des structures complexes	86
8.1	Structure DocContent	86
8.2	Structure ActivationSecret.....	86
8.3	Structure SignatureProfile	87
8.4	Structure Document	91
8.5	Structure SignatureOptionalInfos	92
8.6	Structure SignatureProductionPlace	94
8.7	Structure CommitmentType.....	95
8.8	Structure ObjectIdentifierType	95
8.9	Structure SignatureReport.....	96
8.10	Structure SignVerifReport.....	97

8.11	Structure InfoFile	97
8.12	Structure SignaturePolicy	98
8.13	Structure ConstraintChain	111
8.14	Structure SigPolInfo	112
8.15	Structure SkGenerationProfileSpec	112
8.16	Structure ServerId2NameInfo	115
8.17	Structure CertificateRequestParameters	115
8.18	Structure RequestCertificateResult	115
8.19	Structure ActivationData	116
8.20	Structure ActivationResult	116
8.21	Structure SkGenerationProfileRef	117
8.22	Structure PasswordActivationSecret.....	117
8.23	Structure Role	117
8.24	Structure Credential	118
8.25	Structure Group	119
8.26	Structure User	120
8.27	Structure Application	120
8.28	Structure CertificateInfo.....	121
8.29	Structure ServerId	121
8.30	Structure UsersToGroupAssociation	121
8.31	UserResponse	121

1 Introduction

1.1 Présentation du contexte

Ce document présente les interfaces externes en mode service web « REST » du service de signature. Ce service permet la génération de signatures électroniques via des web services REST reposant sur le serveur de signature.

Ce document décrit les opérations exposées par MetaSIGN-Server et les types des arguments utilisés en entrée et en retour pour ces opérations.

Les chapitres suivants décrivent chacune des opérations suivantes :

- Dépôt d'un document,
- Récupération d'un document déposé,
- Signature d'un document,
- Augmentation d'une signature,
- Vérification d'une signature,
- Récupération des identifiants des signataires (accessible uniquement aux utilisateurs SignManager).

1.2 Références documentaires

1.2.1 Références internes

Référence	Titre	Auteur
MSIGN-SRV-GDE-02	Définition des interfaces	VKA
MSIGN-PS-02	Description des politiques de signature au format XML	FPU

1.2.2 Références externes

Référence	Titre	Auteur

1.3 Glossaire

Acronymes	Définition
WS	Web Service
WSDL	Web Service Description Language
XSD	XML Schema Definition
XML	Extensible Markup Language (langage de balisage extensible)
URL	Uniform Resource Locator (localisateur uniforme de ressource)
SCEP	Simple Certificate Enrollment Protocol (protocole simple d'enregistrement de certificat)
JSON	JavaScript Object Notation (JSON) est un format de données textuelles dérivé de la notation des objets du langage JavaScript.

2 Généralités

Toutes les requêtes exécutées sur l'interface REST de metasign-server retournent l'un des codes de réponse suivant :

Codes de succès :

Code HTTP	Description
200 – OK	Code de succès. L'opération s'est déroulée sans erreur.

Codes d'erreur côté client

Code HTTP	Description
400 – Bad Request	La requête envoyée par le client du web service est malformée ou invalide
401 – Unauthorized	Problème lors de l'authentification du client
404 – Not Found	La ressource demandée n'existe pas.
429 – Too Many Requests	La requête est rejetée à cause de la limite de ressources disponible.

Codes d'erreur côté serveur

Code HTTP	Description
500 – Server error	L'appel de la ressource est valide mais une erreur interne s'est produite pendant le traitement de la requête
501 – Not Implemented	La requête demandée n'est pas implémentée.
503 – Server Unavailable	Le serveur n'est pas capable pour le moment de traiter la requête due à une surcharge temporaire ou à des conditions de maintenance.

En cas de retour, la réponse contient un champ *returnStatus* décrivant un statut de retour :

Statut	Description
MSIGN_SRV_STATUS_SUCCESS	Succès de l'opération
MSIGN_SRV_STATUS_INTERNAL_ERROR	Erreur interne lors du traitement
MSIGN_SRV_STATUS_REJECTED_REQUEST	Requête invalide

En cas de statut en erreur, un message d'erreur sera retourné dans le champ *errorInfo*.

3 Authentification

L'authentification auprès du serveur s'effectue avec un certificat client.

Ce certificat permet :

- De vérifier les droits d'accès au serveur,
- D'identifier l'appelant des services REST.

Dans le cas où le certificat fourni est celui d'un administrateur de signature (SignManager), il peut effectuer une opération en délégation pour un utilisateur (signataire) dont il devra spécifier l'identifiant selon l'opération demandée.

Dans le cas où le certificat fourni est celui d'un signataire (Signer), le paramètre pour la délégation ne doit pas être fourni ou doit être l'identifiant du signataire à qui appartient le certificat d'authentification.

Note : Le certificat d'authentification n'est pas le certificat de signature.

4 Les méthodes de récupération d'informations

Les méthodes de l'API pour la récupération d'informations sont décrites dans le tableau ci-dessous :

URL	Méthode	Description
/signserver/info	GET	récupération des informations sur les méthodes disponibles
/signserver/info/users	GET	Récupération des informations sur l'utilisateur associé au certificat d'authentification donné

L'URL est préfixée par :

`https://<server_name>:8443/servlets/<instance>/com.bull.security.signserver.rest/signserver/`. `serveur_name` correspond au nom du serveur et `<instance>` correspond au nom de l'instance du framework sur lequel le Signature Server est installé.

4.1 Récupération de l'ensemble des méthodes disponibles

Cette méthode permet de récupérer l'ensemble des méthodes disponibles.

4.1.1 Format d'entrée

Méthode HTTP	URL pattern
GET	/signserver/info

Exemple de requête :

GET `https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/info`

4.1.2 Format de sortie

Paramètre	Présence	Type	Description
<i>methods</i>	Obligatoire	Array of String	Liste des méthodes disponibles.

Exemple de réponse :

HTTP/1.1 200 OK

```
{  
  "methods": [  
    "info",
```

```
"sigOps/signDoc",  
  
"sigOps/retDocSig",  
  
"sigOps/augSig",  
  
"sigOps/depDocSig",  
  
"sigOps/verifSig",  
  
"sigOps/listSignerId"  
  
]  
  
}
```

4.2 Récupération des informations de l'utilisateur

Cette méthode permet de récupérer des informations sur l'utilisateur ayant fourni son certificat d'authentification.

Elle est accessible pour tous les types d'utilisateurs du serveur de signature.

Elle permet aussi de vérifier l'authentification d'un utilisateur.

4.2.1 Format d'entrée

Méthode HTTP	URL pattern
GET	/signserver/info/users

Exemple de requête :

```
GET https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/info/users
```

4.2.2 Format de sortie

Paramètre	Présence	Type	Description
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération
<i>id</i>	Optionnel	Integer	Identifiant de l'utilisateur en cas de succès
<i>response</i>	Optionnel	UserResponse	Informations sur l'utilisateur en cas de succès

Exemple de réponse :

```
HTTP/1.1 200 OK
```

```
{
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "id": 704,
  "response": {
    "name": "Test Signer for SOAPUI",
    "certificatesInfo": [
      {
        "authMethod": "password",
        "certificate": "MII...ooGWxw=="
      }
    ],
    "credentials": [
      {
        "credentialKey": "password"
      },
      {
        "credentialKey": "x509Certificate_zVS1opyxmH2UVDC+/TswbCX+P0g=",
        "x509Certificate": "MIIF...jo"
      }
    ],
    "roles": [
      {
        "roleNS":
"http://www.bull.security.com/signserver/1.3.0/signserver.ecore#Signer"
      }
    ],
    "trustedApps": [
      {
        "name":
"Super_Admin_Application_CN=SuperAdmin-
instance1,OU=admin,OU=instance1,OU=MetaPKI,O=BULL,C=FR",
        "id": "12"
      }
    ]
  }
}
```

```
    }  
  ]  
}  
}
```

5 Les méthodes pour les opérations de signature

Les méthodes de l'API pour les opérations de signature sont décrites dans le tableau ci-dessous :

URL	Méthode	Description
/signserver/sigOps/depDocSig	POST	Dépôt d'un document ou d'une signature sur le serveur
/signserver/sigOps/retDocSig	POST	Récupération d'un document ou d'une signature sur le serveur
/signserver/sigOps/signDoc	POST	Génération d'une signature
/signserver/sigOps/augSig	POST	Augmentation d'une signature
/signserver/sigOps/verifSig	POST	Vérification d'une signature
/signserver/sigOps/listSignerId	POST	Liste des identifiants des signataires

Ces opérations s'effectuent avec une authentification de l'appelant.

L'URL est préfixée par :

`https://<server_name>:8443/servlets/<instance>/com.bull.security.signserver.rest/signserver/sigOps.`

server_name correspond au nom du serveur et <instance> correspond au nom de l'instance du framework sur lequel le Signature Server est installé.

5.1 Dépôt de document sur le serveur

Cette méthode permet de déposer des documents ou signatures sur le serveur.

5.1.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/sigOps/depDocSig

Exemple de requête :

POST `https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/sigOps/depDocSig`

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.

<i>bytes</i>	Obligatoire	String	Contenu du document encodé en base64.
--------------	-------------	--------	---------------------------------------

Exemple :

```
{
  "bytes": "JVBERi0xLjQKJcOkw7zDtsO...YwCiU1RU9GCg=="
}
```

5.1.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>docId</i>	Optionnel	String	Identifiant du document en cas de succès de l'opération
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{
  "type": "DepDocSigResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "docId": "4470"
}
```

5.2 Récupération de document sur le serveur

Cette méthode permet de récupérer des documents ou signatures sur le serveur.

5.2.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/sigOps/retDocSig

Exemple de requête :

POST https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/sigOps/retDocSig

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>id</i>	Obligatoire	String	Identifiant du document à récupérer.

Exemple :

```
{  
  
    "id": "3514"  
  
}
```

5.2.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>docContent</i>	Optionnel	DocContent	Contenu du document en cas de succès de l'opération
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{  
  
    "type": "DocContentResponse",  
  
    "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",  
  
    "docContent": {  
  
        "bytes": "JVBERi0xLjQKJcOkw7zDtsOfCjI...JSVFT0YK"  
  
    }  
  
}
```

5.3 Signature d'un document

Cette méthode permet de générer une signature pour un document et un utilisateur.

5.3.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/sigOps/signDoc

Exemple de requête :

POST https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/sigOps/signDoc

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>deferredMode</i>	Optionnel	Boolean	Mode de retour de la signature du document (par défaut, <i>false</i>) Si la valeur est " <i>true</i> " alors la réponse de la requête contiendra l'identifiant de la signature Si la valeur est " <i>false</i> " alors la réponse de la requête contiendra la signature
<i>includeVerificationReport</i>	Optionnel	Boolean	Génération d'un rapport de vérification de la signature si la valeur est " <i>true</i> " Par défaut <i>false</i>
<i>secret</i>	Obligatoire	ActivationSecret	Secret d'activation de la clé du signataire.
<i>signatureKeyId</i>	Obligatoire	String	Identifiant de la clé de signature utilisée pour signer le document
<i>signatureProfile</i>	Obligatoire	SignatureProfile	Profil de signature à utiliser pour signer le document
<i>optionalInfos</i>	Optionnel	SignatureOptionalInfos	Informations optionnelles pouvant être ajoutées à la signature
<i>document</i>	Obligatoire	Document	Document à signer

Exemple :

```
{
  "inDelegationOf": "461",
  "secret": {
    "password": "secret",
    "type": "PassphraseActivationSecret"
  },
  "signatureKeyId": "SignatureKeyForSignatureTests",
  "signatureProfile": {
    "profile": {
      "signaturePolicyOid": "1.0.9.4.2015",
      "attachment": "ENVELOPED",
      "format": "PADES_BES",
      "augmentation": "NONE",
      "signatureAlgoId": "sha256withrsa",
      "archive": "false"
    }
  },
  "document": {
    "docID": "3514",
    "mimeType": "application/pdf"
  },
  "deferredMode" : false,
  "includeVerificationReport" : true
}
```

5.3.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>signatureReport</i>	Optionnel	SignatureReport	Rapport de vérification de la signature en cas de succès et si le paramètre d'entrée

			<i>includeVerificationReport</i> est positionné à true
<i>signatureContent</i>	Optionnel	DocContent	Contenu de la signature en cas de succès de l'opération
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{
  "type": "SignDocResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "signatureContent": {
    "bytes": "JVBERi0xLjQKJcOkw7zDtsOfCjI...JSVFT0YK"
  }
}
```

5.4 Augmentation d'une signature

Cette méthode permet d'augmenter une signature.

5.4.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/sigOps/augSig

Exemple de requête :

POST https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/sigOps/augSig

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera

			réalisée en cas d'opération en délégation.
<i>deferredMode</i>	Optionnel	Boolean	Mode de retour de la signature du document (par défaut, <i>false</i>) Si la valeur est " <i>true</i> " alors la réponse de la requête contiendra l'identifiant de la signature Si la valeur est " <i>false</i> " alors la réponse de la requête contiendra la signature
<i>signature</i>	Obligatoire	Document	Signature à augmenter
<i>document</i>	Optionnel	Document	Document original qui a été signé (uniquement requis pour les signatures détachées)
<i>signatureProfile</i>	Obligatoire	SignatureProfile	Profil de signature à utiliser pour augmenter la signature
<i>optionalInfos</i>	Optionnel	SignatureOptionalInfos	Informations optionnelles pouvant être ajoutées à la signature

Exemple :

```
{
  "signature": {
    "docID": "4297"
  },
  "signatureProfile": {
    "profile": {
      "verificationPolicyOid" : "1.0.9.4.2015",
      "attachment": "ENVELOPED",
      "format": "PADES_BES",
      "augmentation": "LTV",
      "signerRole": {
        "role": "Test Signer",
        "organization": "BULL S.A.S."
      },
      "signatureAlgoId": "shalwithrsa",
      "archive": "false"
    }
  }
}
```

```
    },  
    "deferredMode": "false",  
    "inDelegationOf": "461"  
}
```

5.4.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>signatureReport</i>	Obligatoire	SignatureReport	Rapport de vérification de la signature en cas de succès
<i>signatureContent</i>	Optionnel	DocContent	Contenu de la signature en cas de succès de l'opération
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{  
  "type": "SignDocResponse",  
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",  
  "signatureReport": {  
    ...  
  }  
  "signatureContent": {  
    "bytes": "JVBERi0xLjQKJcOkw7zDtsOfCjI...JSVFT0YK"  
  }  
}
```

5.5 Vérification d'une signature

Cette méthode permet de vérifier une signature.

5.5.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/sigOps/verifSig

Exemple de requête :

POST https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/sigOps/verifSig

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>signatureFormat</i>	Obligatoire	String	Format de la signature à vérifier : <ul style="list-style-type: none"> • CADES • XADES • PADES
<i>signature</i>	Obligatoire	Document	Signature à vérifier
<i>document</i>	Optionnel	Document	Document original qui a été signé (uniquement requis pour les signatures détachées)
<i>policyID</i>	Optionnel	String	OID de la politique de vérification. Obligatoire pour vérifier des signatures BES
<i>verificationLevel</i>	Optionnel	String	Niveau d'augmentation minimal auquel la signature doit être conforme : <ul style="list-style-type: none"> • B • T • LT • LTA

Exemple :

```
{
  "signatureFormat": "PADES",
  "signature": {
    "docID": "4383"
  },
  "policyID": "1.0.9.4.2015"
```

}

5.5.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>report</i>	Obligatoire	SignVerifReport	Rapport de vérification de la signature en cas de succès
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

5.6 Récupération des identifiants des signataires

Cette méthode permet de récupérer les identifiants des signataires.

Elle n'est accessible que pour les administrateurs.

5.6.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/sigOps/listSignerId

Exemple de requête :

POST https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/sigOps/listSignerId

Body :

Paramètre	Présence	Type	Description
<i>signerName</i>	Obligatoire	String	Nom du signataire dont on souhaite récupérer l'identifiant.

Exemple :

```
{
  "signerName": "Test Signer for SOAPUI"
}
```

5.6.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>signerId</i>	Obligatoire	String	Identifiant du signataire en cas de succès
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

HTTP/1.1 200 OK

```
{
  "type": "ListSignerIdResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "signerId": "89"
}
```

6 Les méthodes pour les opérations d'administration pour la signature

Les méthodes de l'API pour les opérations d'administration pour la signature sont décrites dans le tableau ci-dessous :

URL	Méthode	Description
Gestion des politiques de signature		
/signserver/adminSig/depositSigPolicy	POST	Dépôt d'une politique de signature
/signserver/adminSig/listSigPolicy	POST	Liste des politiques de signature
/signserver/adminSig/deleteSignaturePolicy	POST	Suppression d'une politique de signature
/signserver/adminSig/retrieveSignaturePolicy	POST	Récupération d'une politique de signature
/signserver/adminSig/depositSignKeyProfile	POST	Dépôt d'un profil de clé de signature
Gestion des profils de clé de signature		
/signserver/adminSig/updateSignKeyProfile	POST	Mise à jour d'un profil de clé de signature
/signserver/adminSig/listSignKeyProfile	POST	Liste des profils de clé de signature
/signserver/adminSig/deleteSignKeyProfile	POST	Suppression d'un profil de clé de signature
/signserver/adminSig/retrieveSignKeyProfile	POST	Récupération d'un profil de clé de signature
Gestion des profils de signature		
/signserver/adminSig/depositSigProfile	POST	Dépôt d'un profil de signature
/signserver/adminSig/updateSigProfile	POST	Mise à jour d'un profil de signature
/signserver/adminSig/listSigProfile	POST	Liste des profils de signature
/signserver/adminSig/deleteSignatureProfile	POST	Suppression d'un profil de signature
/signserver/adminSig/retrieveSignatureProfile	POST	Récupération d'un profil de signature
Gestion des clés de signature et des certificats		
/signserver/adminSig/depositKeyStore	POST	Dépôt d'un magasin de clés
/signserver/adminSig/updateSigSecret	POST	Mise à jour du secret d'activation
/signserver/adminSig/depositCert4SignKey	POST	Dépôt d'un certificat pour une clé de signature
/signserver/adminSig/requestCertForSignKey	POST	Requête de certificat pour une clé de signature (nécessite un serveur SCEP)
/signserver/adminSig/generateCSR4SignKey	POST	Génération d'une CSR pour une clé de signature
/signserver/adminSig/deleteSignatureKey	POST	Suppression d'une clé de signature
/signserver/adminSig/activateSignatureKey	POST	Activation d'une clé de signature

/signserver/adminSig/createSignKeyCertFromProfile	POST	Création d'une clé de signature pour un profil de clé donné
/signserver/adminSig/createSignKeyCertFromKeyPair	POST	Création d'une clé de signature à partir d'une paire de clés
/signserver/adminSig/createSignKeyCertFromPKCS12	POST	Création d'une clé de signature à partir d'un fichier PKCS12

Ces opérations s'effectuent avec une authentification de l'appelant.

L'URL est préfixée par :
https://<server_name>:8443/servlets/<instance>/com.bull.security.signserver.rest/signserver/adminSig.
 serveur_name correspond au nom du serveur et <instance> correspond au nom de l'instance du framework sur lequel le Signature Server est installé.

6.1 Dépôt de politique de signature

Cette méthode permet de déposer une politique de signature sur le serveur.

6.1.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/depositSigPolicy

Exemple de requête :

POST

<https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/depositSigPolicy>

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>policyName</i>	Obligatoire	String	Nom de la politique de signature
<i>infoFile</i>	Obligatoire	InfoFile	Description d'informations concernant la politique

<i>lockPolicy</i>	Obligatoire	Boolean	Indique si la politique doit être bloquée ou non lorsqu'elle a été utilisée : <ul style="list-style-type: none">• True : la politique sera bloquée et ne pourra plus être modifiée• False : la politique ne sera pas bloquée et pourra être modifiée
<i>signaturePolicy</i>	Obligatoire	SignaturePolicy	Element complexe décrivant la politique de signature

Exemple :

```
{  
  "policyName" : "Test policy oid urn:oid:1.0.9.4.2019",  
  "infoFile" :  
    {...},  
  "lockPolicy" : "false",  
  "signaturePolicy" :  
    { ...}  
}
```

6.1.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>artifactId</i>	Optionnel	String	Identifiant de la politique en cas de succès de l'opération
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{  
  "type": "ArtifactIdResponse",  
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",  
  "artifactId": "823"  
}
```

6.2 Liste des politiques de signature

Cette méthode permet de récupérer la liste des identifiants des politiques de signature sur le serveur.

6.2.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/listSigPolicy

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/listSigPolicy

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>constraints</i>	Optionnel	ConstraintChain	Paramètre pour filtrer des politiques

6.2.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>signaturePolicyInfos</i>	Optionnel	Liste de SigPolInfos	Informations sur les politiques en cas de succès de l'opération
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{  
  "type": "ListSigPolResponse",  
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",  
}
```

```
"signaturePolicyInfos": [  
  {  
    "policyOid": "1.0.9.4.2015",  
    "policyName": "Test policy oid urn:oid:1.0.9.4.2015",  
    "isLocked": false  
  },  
  {  
    "policyOid": "1.0.9.4.2019",  
    "policyName": "Test policy oid urn:oid:1.0.9.4.2019",  
    "isLocked": false  
  }  
]  
}
```

6.3 Suppression d'une politique de signature

Cette méthode permet de supprimer une politique de signature sur le serveur.

6.3.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/deleteSignaturePolicy

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/deleteSignaturePolicy

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.

<i>signaturePolicyOid</i>	Obligatoire	String	OID de la politique
---------------------------	-------------	--------	---------------------

Exemple :

```
{  
    "signaturePolicyOid" : "1.0.9.4.2019"  
}
```

6.3.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>signaturePolicyOid</i>	Optionnel	String	OID de la politique supprimée
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{  
    "type": "DeleteSignaturePolicyResponse",  
    "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",  
    "signaturePolicyOid": "1.0.9.4.2019"  
}
```

6.4 Récupération d'une politique de signature

Cette méthode permet de récupérer une politique de signature sur le serveur.

6.4.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/retrieveSignaturePolicy

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/retrieveSignaturePolicy

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>signaturePolicyOid</i>	Obligatoire	String	OID de la politique

Exemple :

```
{
    "signaturePolicyOid" : "1.0.9.4.2019"
}
```

6.4.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>signaturePolicy</i>	Optionnel	SignaturePolicy	Politique de signature
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{
    "type": "RetrieveSignaturePolicyResponse",
    "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
    "signaturePolicy": {...}
}
```

6.5 Dépôt d'un profil de clé de signature

Cette méthode permet de déposer un profil de clé de signature sur le serveur.

6.5.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/depositSignKeyProfile

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/depositSignKeyProfile

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>infoFile</i>	Obligatoire	InfoFile	Description d'informations du fichier
<i>profileContent</i>	Obligatoire	SkGenerationProfileSpec	Profil génération de la clé de signature

Exemple :

```
{
  "infoFile": {
    "descriptions": {
      "lang": "fr",
      "description": "Profil de cle RSA"
    },
    "keyWords": {
      "lang": "fr",
      "keywords": "RSA"
    }
  }
}
```

```
    },
    "profileContent": {
      "kpAlgo": {
        "name": "RSA"
      },
      "algoParameters": {
        "parameter": {
          "key": "KEY_LEN",
          "value": "2048"
        }
      },
      "name": "RSA-KeyProfile"
    }
  }
}
```

6.5.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>artifactId</i>	Optionnel	String	Identifiant du profil de génération de clé
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{
  "type": "ArtifactIdResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "artifactId": "1296"
}
```

6.6 Mise à jour d'un profil de clé de signature

Cette méthode permet de mettre à jour un profil de clé de signature sur le serveur.

6.6.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/updateSignKeyProfile

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/updateSignKeyProfile

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>newProfileContent</i>	Obligatoire	SkGenerationProfileSpec	Informations pour le nouveau profil de génération de clé
<i>profileId</i>	Obligatoire	String	Identifiant du profil de clé

Exemple :

```
{
  "newProfileContent": {
    "kpAlgo": {
      "name": "RSA"
    },
    "algoParameters": {
      "parameter": {
        "key": "KEY_LEN",
        "value": "1024"
      }
    }
  }
}
```

```
    },  
    "name": "RSA-KeyProfile"  
  },  
  "profileId": "832"  
}
```

6.6.2 Format de sortie

Paramètre	Présence	Type	Description
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{  
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS"  
}
```

6.7 Liste des profils de clé de signature

Cette méthode permet de lister les profils de clé de signature sur le serveur.

6.7.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/listSignKeyProfile

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/listSignKeyProfile

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnelle	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>constraints</i>	Optionnelle	ConstraintChain	Filtrage pour la construction de la liste

6.7.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>skgProfileInfos</i>	Optionnel	List de ServerId2NameInfo	Liste de profils de clé
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{
  "type": "ListSKGPRResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "skgProfileInfos": [
    {
      "serverId": "775",
      "name": "NewKeyProfile"
    },
    {
      "serverId": "946",
      "name": "RSA-KeyProfile"
    }
  ]
}
```

6.8 Suppression d'un profil de clé de signature

Cette méthode permet de supprimer un profil de clé de signature sur le serveur.

6.8.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/deleteSignKeyProfile

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/deleteSignKeyProfile

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnelle	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>sigKeyGenProfileId</i>	Obligatoire	String	Identifiant du profil de clé à supprimer

Exemple :

```
{  
  
  "sigKeyGenProfileId": "1296"  
}
```

6.8.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>sigKeyGenProfileId</i>	Optionnel	String	Identifiant du profil de clé supprimé
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

```
{
```

```
"type": "DeleteSigKeyGenProfileResponse",  
"returnStatus": "MSIGN_SRV_STATUS_SUCCESS",  
"sigKeyGenProfileId": "1296"  
}
```

6.9 Récupération d'un profil de clé de signature

Cette méthode permet de récupérer un profil de clé de signature sur le serveur.

6.9.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/retrieveSignKeyProfile

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/retrieveSignKeyProfile

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnelle	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>sigKeyGenProfileId</i>	Obligatoire	String	Identifiant du profil de clé à récupérer

Exemple :

```
{  
    "sigKeyGenProfileId": "1296"  
}
```

6.9.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé

<i>signatureKeyGenerationProfile</i>	Optionnel	SkGenerationProfileSpec	Description du profil de clé
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

```
{
  "type": "RetrieveSkgProfileResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "signatureKeyGenerationProfile": {...}
}
```

6.10 Dépôt d'un profil de signature

Cette méthode permet de déposer un profil de signature sur le serveur.

6.10.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/depositSigProfile

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/depositSigProfile

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>infoFile</i>	Obligatoire	InfoFile	Description d'informations du fichier

<i>signatureProfile</i>	Obligatoire	Profile	Profil de génération de la signature
-------------------------	-------------	-------------------------	--------------------------------------

Exemple :

```
{
  "infoFile": {
    "descriptions": {
      "lang": "fr",
      "description": "Profil de signature de test 1"
    },
    "keyWords": {
      "lang": "fr",
      "keywords": "DEPSIGPROF-01"
    }
  },
  "signatureProfile": {
    "attachment": "ENVELOPED",
    "format": "PADES_EPES",
    "augmentation": "NONE",
    "signatureAlgoId": "shalwithrsa",
    "archive": "false"
  }
}
```

6.10.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>artifactId</i>	Optionnel	String	Identifiant du profil de génération de signature
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{  
  "type": "ArtifactIdResponse",  
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",  
  "artifactId": "1328"  
}
```

6.11 Mise à jour d'un profil de signature

Cette méthode permet de mettre à jour un profil de signature sur le serveur.

6.11.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/updateSigProfile

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/updateSigProfile

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>signatureProfile</i>	Obligatoire	Profile	Informations pour le profil de génération de signature
<i>profileId</i>	Obligatoire	String	Identifiant du profil de clé

Exemple :

```
{  
  "signatureProfile": {  
    "attachment": "DETACHED",  
    "format": "XADES_EPES",  
    "augmentation": "NONE",  
  }  
}
```

```
        "signatureAlgoId": "shalwithrsa",
        "archive": "false"
    },
    "infoFile": {
        "descriptions": {
            "lang": "fr",
            "description": "Profil de signature de test new"
        },
        "keyWords": {
            "lang": "fr",
            "keywords": "DEPSIGPROF-01"
        }
    },
    "profileId": "1335"
}
```

6.11.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>artifactId</i>	Optionnel	String	Identifiant du profil de génération de signature
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{
    "type": "ArtifactIdResponse",
    "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
    "artifactId": "1335"
}
```

6.12 Liste des profils de signature

Cette méthode permet de lister les profils de signature sur le serveur.

6.12.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/listSigProfile

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/listSigProfile

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnelle	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>constraints</i>	Optionnelle	ConstraintChain	Filtrage pour la construction de la liste

6.12.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>sigProfileInfos</i>	Optionnel	List de ServerId2NameInfo	Liste de profils de signature
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{  
  "type": "ListSigProfileResponse",  
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",  
  "sigProfileInfos": [  

```

```
{
  "serverId": "770"
},
{
  "serverId": "718"
},
{
  "serverId": "1345"
}
]
```

6.13 Suppression d'un profil de clé de signature

Cette méthode permet de supprimer un profil de clé de signature sur le serveur.

6.13.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/deleteSignatureProfile

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/deleteSignatureProfile

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnelle	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>signatureProfileId</i>	Obligatoire	String	Identifiant du profil de signature à supprimer

Exemple :

```
{  
  
    "signatureProfileId" : "1335"  
  
}
```

6.13.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>signatureProfileId</i>	Optionnel	String	Identifiant du profil de signature supprimé
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

```
{  
  
    "type": "DeleteSignatureProfileResponse",  
  
    "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",  
  
    "signatureProfileId": "1335"  
  
}
```

6.14 Récupération d'un profil de signature

Cette méthode permet de récupérer un profil de signature sur le serveur.

6.14.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/retrieveSignatureProfile

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/retrieveSignatureProfile

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnelle	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>signatureProfileId</i>	Obligatoire	String	Identifiant du profil de signature à récupérer

Exemple :

```
{  
  "signatureProfileId": "1328"  
}
```

6.14.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>signatureProfile</i>	Optionnel	Profile	Description du profil de signature
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

```
{  
  "type": "RetrieveSignatureProfileResponse",  
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",  
  "signatureProfile": {  
    "archive": false,  
    "attachment": "ENVELOPED",  
    "format": "PADES_EPES",  
    "augmentation": "NONE",  
    "signatureAlgoId": "shalwithrsa",  
    "canonicalizationAlgo": "INCLUSIVE",  
    "requireSigningTime": true,  
    "requirePlaceOfSignature": false,  
    "requireContactInfo": false  
  }  
}
```

```
}  
}
```

6.15 Dépôt d'un key store

Cette méthode permet de déposer un keystore (paire de clés) sur le serveur.

6.15.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/depositKeyStore

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/depositKeyStore

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>keyStoreType</i>	Optionnel	String	Type de keystore (Pkcs12Keystore...)
<i>keyStoreContent</i>	Optionnel	String	Contenu du keystore encodé en base64

Exemple :

```
{  
    "inDelegationOf": "1033",  
    "keyStoreType": "Pkcs12Keystore",  
    "keyStoreContent": "MIACAQMwgAYJKoZ..."  
}
```

6.15.2 Format de sortie

Paramètre	Présence	Type	Description
-----------	----------	------	-------------

<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>keystoreId</i>	Optionnel	String	Identifiant du keystore
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{
  "type": "DepositKeystoreResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "keystoreId": "610"
}
```

6.16 Mise à jour d'un secret de signature

Cette méthode permet de mettre à jour un secret de signature sur le serveur.

6.16.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/updateSigSecret

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/updateSigSecret

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
secret	Obligatoire	ActivationSecret	L'ancien secret d'activation
newsecret	Obligatoire	ActivationSecret	Le nouveau secret d'activation

Exemple :

```
{
  "secret": {
    "password": "secret",
    "type": "PassphraseActivationSecret"
  },
  "newsecret": {
    "password": "newsecret",
    "type": "PassphraseActivationSecret"
  },
  "inDelegationOf": "89"
}
```

6.16.2 Format de sortie

Paramètre	Présence	Type	Description
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS"
}
```

6.17 Dépôt d'un certificat pour une clé de signature

Cette méthode permet de déposer un certificat associé à une clé de signature sur le serveur.

6.17.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/depositCert4SignKey

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/depositCert4SignKey

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>secret</i>	Obligatoire	ActivationSecret	Le secret d'activation
<i>signatureKeyId</i>	Obligatoire	String	Identifiant de la clé de signature
<i>x509Certificate</i>	Obligatoire	String	Contenu du certificat encode en base64

Exemple :

```
{
  "secret": {
    "password": "secret",
    "type": "PassphraseActivationSecret"
  },
  "signatureKeyId": "DEFAULT_SigKey4",
  "inDelegationOf": "1033",
  "x509Certificate": "MIIF..."
}
```

6.17.2 Format de sortie

Paramètre	Présence	Type	Description
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{  
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS"  
}
```

6.18 Requête d'un certificat pour une clé de signature

Cette méthode permet de demander un certificat pour une clé de signature.

Une configuration du serveur de signature avec un serveur SCEP est nécessaire.

6.18.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/requestCertForSignKey

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/requestCertForSignKey

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>certRequestParameters</i>	Optionnel	CertificateRequestParameters	Paramètres pour la demande de certificat
<i>secret</i>	Obligatoire	ActivationSecret	Secret d'activation de la clé
<i>signatureKeyId</i>	Obligatoire	String	Identifiant de la clé de signature

Exemple :

```
{  
  
  "secret": {  
  
    "password": "secret",  
  
    "type": "PassphraseActivationSecret"  
  
  },  
  
  "certRequestParameters": {  
  
    "type": "SCEPRequestParameters"  
  
  },  
  
  "signatureKeyId": "DEFAULT_SigKey4",  
  
  "inDelegationOf": "1033"  
  
}
```

6.18.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>certResult</i>	Optionnel	RequestCertificateResult	Résultat de la requête
<i>signatureKeyId</i>	Optionnel	String	Identifiant de la clé de signature
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

6.19 Génération d'une CSR pour une clé de signature

Cette méthode permet de générer une CSR pour une clé de signature.

6.19.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/generateCSR4SignKey

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/generateCSR4SignKey

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>secret</i>	Obligatoire	ActivationSecret	Secret d'activation de la clé
<i>signatureKeyId</i>	Obligatoire	String	Identifiant de la clé de signature

Exemple :

```
{
  "secret": {
    "password": "secret",
    "type": "PassphraseActivationSecret"
  },
  "signatureKeyId": "DEFAULT_SigKey4",
  "inDelegationOf": "1033"
}
```

6.19.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>publicKey</i>	Optionnel	String	Clé publique encodée en base64
<i>signatureKeyId</i>	Optionnel	String	Identifiant de la clé de signature
<i>csr</i>	Optionnel	String	CSR encodée en base64
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple :

```
{
  "type": "CsrForSignatureKeyResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "signatureKeyId": "DEFAULT_SigKey4",
  "publicKey": "MIIBCgKCAQEA...",
  "csr": "MIICbjCCAVY..."
}
```

```
}
```

6.20 Suppression d'une clé de signature

Cette méthode permet de supprimer une clé de signature.

6.20.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/deleteSignatureKey

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/deleteSignatureKey

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>secret</i>	Obligatoire	ActivationSecret	Secret d'activation de la clé
<i>signatureKeyId</i>	Obligatoire	String	Identifiant de la clé de signature

Exemple :

```
{
  "signatureKeyId": "DEFAULT_SigKey5",
  "secret": {
    "password": "secret",
    "type": "PassphraseActivationSecret"
  },
  "inDelegationOf": "1033"
}
```

6.20.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>signatureKeyId</i>	Optionnel	String	Identifiant de la clé de signature
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple :

```
{
  "type": "DeleteSignatureKeyResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "signatureKeyId": "DEFAULT_SigKey5"
}
```

6.21 Activation d'une clé de signature

Cette méthode permet d'activer une clé de signature.

6.21.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/activateSignatureKey

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/activateSignatureKey

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>activationData</i>	Obligatoire	ActivationData	Données d'activation de la clé
<i>signatureKeyId</i>	Obligatoire	String	Identifiant de la clé de signature

Exemple :

```
{
  "activationData": {
    "type": "DASActivationData",
    "userActivationSecret": "c2VjcmV0",
    "authenticationMethod" : "BASIC_OTP"
  },
  "signatureKeyId": "Signature_Key_OTP-08",
  "inDelegationOf": "937"
}
```

6.21.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>activationResult</i>	Optionnel	ActivationResult	Résultat de l'activation
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple :

```
{
  "type": "ActivateSignatureKeyResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "activationResult": {
    "dynamicChallenge": "uOf5IKOV4qA9XBx4H8ReARZlP0RsaKc4Oqzmux49Anc=",
    "type": "DASActivationResult"
  }
}
```

6.22 Création d'une clé de signature à partir d'un profil de clé

Cette méthode permet de créer une clé de signature à partir d'un profil de clé.

6.22.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/createSignKeyCertFromProfile

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/createSignKeyCertFromProfile

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>secret</i>	Obligatoire	ActivationSecret	Secret d'activation de la clé
<i>skSpec</i>	Conditionnelle	SkGenerationProfileSpec	Description du profil de la clé de signature
<i>skRef</i>	Conditionnelle	SkGenerationProfileRef	Référence au profil de clé de signature

Un des paramètres *skSpec* ou *skRef* doit être présent.

Exemple :

```
{
  "signatureKeyId": "DEFAULT_SigKey4",
  "inDelegationOf": "1033",
  "secret": {
    "password": "secret",
    "type": "PassphraseActivationSecret"
  },
  "skRef" : {
    "skgProfileId" : "44"
  }
}
```

6.22.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>signatureKeyID</i>	Optionnel	String	Identifiant de la clé créée
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple :

```
{
  "type": "CreateSignatureKeyCertResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "signatureKeyID": "DEFAULT_SigKey5"
}
```

6.23 Création d'une clé de signature à partir d'une paire de clés

Cette méthode permet de créer une clé de signature à partir d'une paire de clés.

6.23.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/createSignKeyCertFromKeyPair

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/createSignKeyCertFromKeyPair

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>secret</i>	Obligatoire	ActivationSecret	Secret d'activation de la clé
<i>keyPairCkaId</i>	Obligatoire	String	CKAID de la paire de clé

signatureKeyId	Obligatoire	String	Identifiant de la clé de signature
----------------	-------------	--------	------------------------------------

6.23.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>signatureKeyID</i>	Optionnel	String	Identifiant de la clé créée
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple :

```
{
  "type": "CreateSignatureKeyCertResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "signatureKeyID": "DEFAULT_SigKey"
}
```

6.24 Création d'une clé de signature à partir d'un PKCS12

Cette méthode permet de créer une clé de signature à partir d'un fichier PKCS12.

6.24.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/adminSig/createSignKeyCertFromPKCS12

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/adminSig/createSignKeyCertFromPKCS12

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de

			l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>secret</i>	Obligatoire	ActivationSecret	Secret d'activation de la clé
<i>p12TransportPassword</i>	Obligatoire	PasswordActivationSecret	Mot de passe de transport pour le PKCS12
<i>keyUsage</i>	Optionnelle	KeyUsageArray	Usages de clé devant être réents dans le certificat associé à la clé
<i>p12Id</i>	Obligatoire	String	Identifiant du PKCS12 précédemment déposé à associer au signataire
<i>signatureKeyId</i>	Obligatoire	String	Identifiant de la clé de signature

Exemple :

```
{
  "secret": {
    "password": "secret",
    "type": "PassphraseActivationSecret"
  },
  "p12TransportPassword": {
    "bytes": "cGFzc3dvcmQ="
  },
  "p12Id": "1049",
  "signatureKeyId": "DEFAULT_SigKey5",
  "inDelegationOf": "1033"
}
```

6.24.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>signatureKeyId</i>	Optionnel	String	Identifiant de la clé créée
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple :

```
{
```

```
"type": "CreateSignatureKeyCertResponse",  
"returnStatus": "MSIGN_SRV_STATUS_SUCCESS",  
"signatureKeyID": "DEFAULT_SigKey5"  
}
```

7 Les méthodes pour les opérations d'administration

Les méthodes de l'API pour les opérations d'administration sont décrites dans le tableau ci-dessous :

URL	Méthode	Description
Gestion des utilisateurs		
/signserver/admin/createUser	POST	Création d'un nouvel utilisateur
/signserver/admin/updateUser	POST	Mise à jour d'un utilisateur
/signserver/admin/deleteUser	POST	Suppression d'un utilisateur
/signserver/admin/listUsers	POST	Liste des utilisateurs
/signserver/admin/consultUser	POST	Consultation d'un utilisateur
Gestion des groupes		
/signserver/admin/createGroup	POST	Création d'un groupe
/signserver/admin/addUsersToGroup	POST	Ajout d'utilisateurs au groupe
/signserver/admin/removeUsersFromGroup	POST	Suppression d'utilisateurs du groupe
/signserver/admin/deleteGroup	POST	Suppression du group
Gestion des token FIDO		
/signserver/admin/enrollFIDOInit	POST	Initialisation pour l'enrollement d'un utilisateur avec FIDO
/signserver/admin/depositFIDOToken	POST	Dépôt d'un token FIDO
/signserver/admin/listFIDOToken	POST	Liste des token FIDO
/signserver/admin/deleteFIDOToken	POST	Suppression d'un token FIDO
Gestion de magasin de clés		
/signserver/admin/depositKeystore	POST	Dépôt d'un magasin de clés

Ces opérations s'effectuent avec une authentification de l'appelant.

L'URL est préfixée par :

https://<server_name>:8443/servlets/<instance>/com.bull.security.signserver.rest/signserver/admin.

serveur_name correspond au nom du serveur et <instance> correspond au nom de l'instance du framework sur lequel le Signature Server est installé.

7.1 Création d'un utilisateur

Cette méthode permet de créer un nouvel utilisateur.

7.1.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/admin/createUser

Exemple de requête :

POST <https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/admin/createUser>

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>name</i>	Obligatoire	String	Nom de l'utilisateur
<i>type</i>	Obligatoire	String	Type de description de l'utilisateur : <ul style="list-style-type: none">• UserSpecification• UserReference• Application
<i>roles</i>	Optionnel	List< Role >	Liste des rôles de l'utilisateur
<i>credentials</i>	Obligatoire	List< Credential >	Liste des modes d'authentification de l'utilisateur (certificats, mots de passe)
<i>groups</i>	Optionnel	List< Group >	Liste des groupes auxquels appartient l'utilisateur
<i>applications</i>	Optionnel	List< Application >	Liste des applications reconnues par l'utilisateur

Exemple :

```
{
  "roles": {
    "type": "RoleSpecification",
    "roletype": "Signer",
    "secret": {
      "password": "cGFzc3dvcmQ=",
      "type": "PassphraseActivationSecret"
    }
  },
  "credentials": {
    "type": "PasswordWrapper",
    "credentialValue": "XohImNooBHFR0OVvjcyPj3NgPQ1qq73WKHhVch0VQtg=",
    "hashAlgo": "SHA-256"
  },
  "type": "UserSpecification",
```

```
"name" : "Test user new"
}
```

7.1.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>userId</i>	Optionnel	String	Identifiant de l'utilisateur en cas de succès de l'opération
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{
  "type": "CreateUserResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "userId": "1744"
}
```

7.2 Mise à jour d'un utilisateur

Cette méthode permet de mettre à jour un utilisateur.

7.2.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/admin/updateUser

Exemple de requête :

POST https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/admin/updateUser

Body :

Paramètre	Présence	Type	Description
-----------	----------	------	-------------

<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>name</i>	Obligatoire	String	Nom de l'utilisateur
<i>userOrApplicationId</i>	Obligatoire	String	Nom de l'utilisateur ou application à mettre à jour
<i>type</i>	Obligatoire	String	Type de description de l'utilisateur : <ul style="list-style-type: none">• UserSpecification• UserReference• Application
<i>roles</i>	Optionnel	List< Role >	Liste des rôles de l'utilisateur
<i>credentials</i>	Obligatoire	List< Credential >	Liste des modes d'authentification de l'utilisateur (certificats, mots de passe)
<i>groups</i>	Optionnel	List< Group >	Liste des groupes auxquels appartient l'utilisateur
<i>applications</i>	Optionnel	List< Application >	Liste des applications reconnues par l'utilisateur

Exemple :

```
{
  "type": "UserSpecification",
  "name": "Test Modified User",
  "userOrApplicationId": "89",
  "roles": {
    "type": "RoleSpecification",
    "roletype": "SignManager"
  },
  "credentials":
  [
    {
      "type": "PasswordWrapper",
      "credentialValue": "XohImNooBHFR0OVvjcyJ3NgPQ1qq73WKHhVch0VQtg=",
      "hashAlgo": "SHA-256"
    },
    {
      "type": "X509Certificate",
      "credentialValue": "wxEDAObgNVBAsT..."
    }
  ]
}
```

```
    }  
  ]  
}
```

7.2.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>updatedId</i>	Optionnel	String	Identifiant de l'utilisateur en cas de succès de l'opération
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{  
  "type": "UpdateUserResponse",  
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",  
  "updatedId": "1744"  
}
```

7.3 Suppression d'un utilisateur

Cette méthode permet de supprimer un utilisateur.

7.3.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/admin/deleteUser

Exemple de requête :

POST https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/admin/deleteUser

Body :

Paramètre	Présence	Type	Description
-----------	----------	------	-------------

<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>userId</i>	Obligatoire	String	Identifiant de l'utilisateur à supprimer

Exemple :

```
{
  "userId": "1744"
}
```

7.3.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>deletedId</i>	Optionnel	String	Identifiant de l'utilisateur supprimé en cas de succès de l'opération
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{
  "type": "DeleteResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "deletedId": "1744"
}
```

7.4 Liste des utilisateurs

Cette méthode permet de lister des utilisateurs.

7.4.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/admin/listUsers

Exemple de requête :

POST https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/admin/listUsers

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>constraints</i>	Obligatoire	ConstraintChain	Paramètres pour filtrer l'utilisateur

Exemple :

```
{
  "constraints": {
    "maxResults" : 10,
    "type" : "StringConstraint",
    "constrainedElement":
"http://www.bull.security.com/Server/coreAdmin/v1.3.0/#User.name",
    "isNegated": "false",
    "typeStringConstraint": "CONTAINS",
    "value": "e"
  }
}
```

7.4.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Optionnel	String	Type de réponse renvoyé
<i>users</i>	Optionnel	Liste de User	Liste des utilisateurs en cas de succès de l'opération
<i>totalResults</i>	Optionnel	Integer	Nombre de résultats retournés en cas de succès
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{
  "users": [
    {
      "type": "ApplicationReference",
      "userOrApplicationId": "4",
      "name":
"Super_Admin_Application_CN=EFA6F0D3407950740F438B21CD9D432A,OU=instance1_entity_ADMIN
_bootstrap,OU=component,OU=instance1,OU=MetaPKI,O=BULL,C=FR",
      "roles": [
        {
          "type": "RoleReference",
          "roleNS":
"http://www.bull.security.com/Server/coreAdmin/v1.3.0/#ServerManager"
        },
        {
          "type": "RoleReference",
          "roleNS":
"http://www.bull.security.com/SignServer/extensions/v1.3.0/#SignManager"
        }
      ],
      "credentials": [
        {
          "type": "CredentialReference",
          "credentialId": {
            "credentialKey":
"x509Certificate_LNV+7QncixgstW+LhT+mLEYiM6w="
          }
        }
      ],
      "groups": [],
      "applications": []
    }
  ]
}
```

```
    },  
    {  
      "type": "ApplicationReference",  
      "userOrApplicationId": "12",  
      "name": "Super_Admin_Application_CN=SuperAdmin-  
instance1,OU=admin,OU=instance1,OU=MetaPKI,O=BULL,C=FR",  
      "roles": [  
        {  
          "type": "RoleReference",  
          "roleNS":  
"http://www.bull.security.com/Server/coreAdmin/v1.3.0/#ServerManager"  
        },  
        {  
          "type": "RoleReference",  
          "roleNS":  
"http://www.bull.security.com/SignServer/extensions/v1.3.0/#SignManager"  
        }  
      ],  
      "credentials": [  
        {  
          "type": "CredentialReference",  
          "credentialId": {  
            "credentialKey":  
"x509Certificate_mMWTcYhO2+ajWbaqu2+y0g+XE4c="  
          }  
        }  
      ],  
      "groups": [],  
      "applications": []  
    }  
  ],  
],
```

```
"totalResults": 2
}
```

7.5 Consultation d'un utilisateur

Cette méthode permet de consulter un utilisateur.

7.5.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/admin/consultUser

Exemple de requête :

POST https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/admin/consultUser

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>id</i>	Obligatoire	String	Identifiant de l'utilisateur à consulter

Exemple :

```
{
  "userId": "89"
}
```

7.5.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>name</i>	Optionnel	String	Nom de l'utilisateur en cas de succès de l'opération
<i>roles</i>	Optionnel	String	Roles de l'utilisateur

<i>credentials</i>	Optionnel	Liste de CredentialDescription	Modes d'authentification de l'utilisateur
<i>groups</i>	Optionnel	Liste de ServerId	Identifiants des groupes de l'utilisateur
<i>trustedApplications</i>	Optionnel	Liste de ServerId	Applications reconnues par l'utilisateur
<i>certificatesInfo</i>	Optionnel	Liste de CertificateInfo	Informations complémentaires sur les certificats (en cas d'authentification OTP ou FIDO)
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{
  "type": "ConsultUserResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "name": "Super_Admin_Application_CN=SuperAdmin-
instance1,OU=admin,OU=instance1,OU=MetaPKI,O=BULL,C=FR",
  "roles": [
    {
      "roleNS":
"http://www.bull.security.com/Server/coreAdmin/v1.3.0/#ServerManager"
    },
    {
      "roleNS":
"http://www.bull.security.com/SignServer/extensions/v1.3.0/#SignManager"
    }
  ],
  "credentials": [
    {
      "credentialKey": "x509Certificate_mMWTcYhO2+ajWbaqu2+y0g+XE4c=",
      "x509Certificate":
"MIIE6zCCAtOgAwIBAgIBBTANBgkqhkiG9w0BAQ0FADCBjDELMAkGA1UEBhMCRLIxDALBgNVBAoTBEJVTTEwxE
DAOBgNVBAsTB01ldGFQS0kxDzANBgNVBAsTBmNvbWl1bWljELMAkGA1UECzMCMQ0ExDzANBgNVBAMTBnRlY2hDQTE
tMCsGA1UEBRmKNGI3OGEyMzgtMmRmNS00NGI5LTlhYzQtY2Q0YTMwNzliOTZiMB4XDTE4MDQxMjEzZmJExMFoXD
TI4MDQxMjEzZmJExMFowcTELMakGA1UEBhMCRLIxDALBgNVBAoTBEJVTTEwxEADAOBgNVBAsTB01ldGFQS0kxEjA
```

```

QBgNVBAStCWLuc3RhbmNlMTEOMAwGA1UECxFYWRtaW4xHTAbBgNVBAMTFFNlcGVyQWRtaW4taW5zdGFuY2UxM
IIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEArzVLHxXuUS8DE9pfVC8cdhqZrlgF4LjK+lz5Cz7nmov
c4+cz9n52YdrDcNF+iNfSFCbBC7SyCJcuObA2EaXfk+rGSFm+doey7mvlhnoDe+AdfI107WbL0UK1K5Q5V2ZPG
Qtsk+1WBjpjibF8kmRGzTZp3vA1GJNOW/DRvi7JJ7ukMnEk0a0p4JKCYimlfJU0UvX70TDThVXotwPWz4JCj9F
NxTRfhZXViuoyJE9G5qFBcrQi8Q1021LdHRE3XB5bTQdxnmAy1t0srFS4FbTdHbWncovVW4seug4DuWftHartHn
CN0SARfCNufgxFFEQuNgrKjU4OoVLPx8Xax1T8fWQIDAQABo3IwcDAOBgNVHQ8BAf8EBAMCB4AwHwYDVR0jBBg
wFoAUzvhyyk0LKXy8hflf809yC8ZGxWswCQYDVR0TBAlwADAdBgNVHQ4EFgQUUV9OMwbjn9S9nnpnTqlbgCsU/4z
H8wEwYDVR01BAwwCgYIKwYBBQUHAWIwDQYJKoZIhvcNAQENBQADggIBAECQGS0qddIv5+m/8sdcxbzeZ9LQliu
JmM/D4ukFeTelcJiFG0iTNeWNlopZ9P+F1u7in5gCPfYDFhRRGoPr0vXk0HxBulpUGEL+g1hc0crWxgPzBKXXx
gLgaG26r9+M3EA5KVOiy9dxM3PGfe3qjC1k3p6kEyR/ee3hrowh8bDNmSt+gdqjbW5xtAxx6El1qjY0hs6jGtZ
vi0w7x95IT8A9K+XV4QyHfki3cdyT/HzwMe4whmzWlVRhwCWouEWBw27AE+S1RIm9kg6kClbVcyXy6IH8aKnJj
IbF2sWnGiNiwiG0uTjQkbfxdbRSK8HIVuy48AP+j5ejqprvWmd2PLf4VBHqaLSvbaoD5ienHDTgqiEBSfK/fdI
3kXSTChKxQJGaUM07f6+LtiU4tqKoVAft+CoXRuwYy9WHp8l4l9+FVyxDHInEHON29uk10f5V/XWadHQ06/Y25
Tfz5y00hBHwN0zNtwMQFMelWQY9ub9iK3QVZG5ahQJWGQzSqiijnJlyw4qC0BsI2BC5SPyZZTIq0PspoykSACK
ahVQXZjX3qZe0pd2Qc3PaAwB7vhfWRkdWA6Gt/27yJXMrM/fJCHZ8EPeFLKkndSlwBH5WLvrfnL3oLK9V/iK9Y
ywa8VV/3TBEJbwqtJ6zLXp8NtPxqLUgoZeWMehnZgDkFK15H/C8"

    }

  ],

  "groups": [

    {

      "serverId": "608",

      "name": "Test Grp-CREGROUP03"

    }

  ]

}

```

7.6 Création d'un groupe

Cette méthode permet de créer un groupe.

7.6.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/admin/createGroup

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/admin/createGroup

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>parameter</i>	Obligatoire	Group	Description du groupe de type GroupSpecification

Exemple :

```
{
  "parameter" : {
    "name": "Test Grp-CREGROUP03"
  }
}
```

7.6.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>groupId</i>	Optionnel	String	Identifiant du groupe créé en cas de succès de l'opération
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{
  "type": "CreateGroupResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "groupId": "760"
}
```

7.7 Ajout d'utilisateurs à un groupe

Cette méthode permet d'ajouter des utilisateurs dans un groupe.

7.7.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/admin/addUsersToGroup

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/admin/addUsersToGroup

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>parameter</i>	Obligatoire	UsersToGroupAssociation	Description de l'association entre utilisateurs et groupe

Exemple :

```
{
  "parameter": {
    "userIds": "12",
    "groupId": "608"
  }
}
```

7.7.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>groupId</i>	Optionnel	String	Identifiant du groupe en cas de succès de l'opération

<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{
  "type": "UsersGroupResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "groupId": "760"
}
```

7.8 Suppression d'utilisateurs dans un groupe

Cette méthode permet de supprimer des utilisateurs d'un groupe.

7.8.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/admin/removeUsersFromGroup

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/admin/removeUsersFromGroup

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>parameter</i>	Obligatoire	UsersToGroupAssociation	Description de l'association entre utilisateurs et groupe

Exemple :

```
{
```

```
"parameter": {  
    "userIds": "12",  
    "groupId": "720"  
}  
}
```

7.8.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>groupId</i>	Optionnel	String	Identifiant du groupe en cas de succès de l'opération
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{  
    "type": "UsersGroupResponse",  
    "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",  
    "groupId": "760"  
}
```

7.9 Suppression d'un groupe

Cette méthode permet de supprimer un groupe sur le serveur.

7.9.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/admin/deleteGroup

Exemple de requête :

POST

<https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/admin/deleteGroup>

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>groupId</i>	Obligatoire	String	Identifiant du groupe à supprimer

Exemple :

```
{
  "groupId": "760"
}
```

7.9.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>deletedId</i>	Optionnel	String	Identifiant du groupe supprimé en cas de succès de l'opération
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{
  "type": "DeleteResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "deletedId": "760"
}
```

7.10 Enrôlement d'un utilisateur FIDO

Cette méthode permet d'initialiser l'enrôlement d'un utilisateur FIDO.

Elle retourne les données à envoyer au token FIDO pour l'enrollement d'un utilisateur.

7.10.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/admin/enrollFIDOInit

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/admin/enrollFIDOInit

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>signerId</i>	Obligatoire	String	Identifiant de l'utilisateur à enrôler

Exemple :

```
{  
  
    "signerId": "89"  
}
```

7.10.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>challenge</i>	Optionnel	String	Challenge encode en base64 en cas de succès
<i>appId</i>	Optionnel	String	Identifiant de l'application
<i>version</i>	Optionnel	String	Version U2F
<i>registeredTokens</i>	Optionnel	List de String	Liste des tokens FIDO déjà enregistrés (encodés en base64)
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{
  "type": "EnrollFIDOInitResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "appId": "chdem.frcl.bull.fr",
  "challenge": "KfOyzQ3Eni47pDKCUB4HlEgoN+0egHtrKhX/DUMYERk=",
  "version": "U2F_V2"
}
```

7.11 Dépôt d'un token FIDO

Cette méthode permet de déposer un token FIDO sur le serveur. L'identifiant de ce token pourra être utilisé lors de la génération de signature ou lors de l'authentification auprès du serveur de signature avec une clé nécessitant une authentification FIDO.

7.11.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/admin/depositFIDOToken

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/admin/depositFIDOToken

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>signerId</i>	Obligatoire	String	Identifiant de l'utilisateur associé au token FIDO
<i>registrationData</i>	Obligatoire	String	Données d'enregistrement retournées par le token FIDO
<i>clientData</i>	Obligatoire	String	Données de signature retournées par le token FIDO

Exemple :

```
{
  "signerId": "800",
  "registrationData":
"BQSxdLxJx8olS3DS5cIHzunPF0gg69d+o8ZVCMJtpRtlfBzGuVL4YhaXk2SC2gptPTgmpZCV2vbNfAPi5gOF0
vbZQCpVLf23R37WX9hBM/hhlgeLIhW1faddMvt7no/i45JaYBlVG6th0WWRZZy68AtJUPer/mZg4uAG92hot3L
XDCUwggE8MIHkoAMCAQICCKeQEoAAEVWVclIwCgYIKoZIzj0EAwIwFzEVMBMGA1UEAxMMR251YmJ5IFBpbG90M
B4XDTEyMDgxNDE4MjkzMloXDTEzMDgxNDE4MjkzMlowMTEvMC0GA1UEAxMmUGlsb3RHbnViYnktMC40LjEtNDc
5MDEyODAwMDExNTU5NTczNTIwWTATBgqhkhjOPQIBBgqhkhjOPQMBBwNCAASNYX51yVCOZLzFZzrIKmeZ2jwUR
mgsJYxGP//fWN/S+j5sN4tT15XEpn/7QZnt14YvI6uvAg00uJEboFaZlOEbMAoGCCqGSM49BAMCA0cAMEQCIGD
NtgYenCImLRqSHZbYxwgpsjZlMd2iaIMsuDa80w36AiBjGxRZ8J5jMAVXIjsjYm39IiDuQibiNYNHZeVkcswQQ3
zBFAiAUcYmbzDmH5i6CAsmznDPBkDP3NANS26gPyrAX25Iw5AIhAIJnfWc9iRkzreb2F+Xb3i4kfnBCP9WteAS
m09OWHvhx",
  "clientData":
"eyJ0eXAiOiJuYXZpZ2F0b3IuaWQzmluaXNoRW5yb2xsbWVudCIsImNoYWxsZW5nZSI6InZxcmlM2VlhEZTFKV
XMlX2MzaTQtTGTLSUhSci0zWFZiM2F6dUE1VGlmSG8iLCJjaWRfcHVia2V5Ijp7Imt0eSI6IkVDTiwiY3J2Ijo
iUC0yNTYiLCJ4IjoishPrd2xmWFg3UTRTNU10Q0NuWlVOQnczUk16UE85dE95V2pCcVJsNHRKOCIsInkiOiJYV
mdlR0ZMSVp4MWZYZzN3TnFmZGJuNzVoaTQtXzctQnhoTWxqdZQySHQ0In0sIm9yaWdpbiI6Imh0dHA6Ly9leGF
tcGx1LmNvbSJ9"
```

7.11.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>tokenId</i>	Optionnel	String	Identifiant du token FIDO en cas de succès de l'opération
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{
  "type": "DepositFIDOTokenResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "tokenId": "774"
}
```

7.12 Liste des tokens FIDO

Cette méthode permet de lister les token FIDO pour un utilisateur donné sur le serveur.

7.12.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/admin/listFIDOToken

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/admin/listFIDOToken

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>signerId</i>	Obligatoire	String	Identifiant de l'utilisateur

Exemple :

```
{  
    "signerId": "800"  
}
```

7.12.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>tokenIdList</i>	Optionnel	Liste de String	Identifiants des token FIDO en cas de succès de l'opération
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{  
  "type": "ListFIDOTokenResponse",  
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",  
  "tokenIdList": [  
    "811"  
  ]  
}
```

7.13 Suppression d'un token FIDO

Cette méthode permet de supprimer un token FIDO sur le serveur.

7.13.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/admin/deleteFIDOToken

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/admin/deleteFIDOToken

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>tokenId</i>	Obligatoire	String	Identifiant du token FIDO à supprimer

Exemple :

```
{  
  "tokenId": "693"  
}
```

7.13.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>fidoToken</i>	Optionnel	String	Identifiant du token FIDO supprimé en cas de succès de l'opération
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{
  "type": "DeleteFIDOTokenResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "fidoToken": "774"
}
```

7.14 Dépôt d'un key store

Cette méthode permet de déposer un keystore (paire de clés) sur le serveur.

7.14.1 Format d'entrée

Méthode HTTP	URL pattern
POST	/signserver/admin/depositKeystore

Exemple de requête :

POST

https://servername:8443/servlets/instancename/com.bull.security.signserver.rest/signserver/admin/depositKeystore

Body :

Paramètre	Présence	Type	Description
<i>inDelegationOf</i>	Optionnel	String	Ce paramètre permet de préciser l'identifiant de l'utilisateur pour le compte duquel l'opération sera réalisée en cas d'opération en délégation.
<i>keyStoreType</i>	Optionnel	String	Type de keystore (Pkcs12Keystore...)

<i>keyStoreContent</i>	Optionnel	String	Contenu du keystore encodé en base64
------------------------	-----------	--------	--------------------------------------

Exemple :

```
{
  "inDelegationOf": "1033",
  "keyStoreType": "Pkcs12Keystore",
  "keyStoreContent": "MIACAQMwgAYJKoZ..."
}
```

7.14.2 Format de sortie

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de réponse renvoyé
<i>keystoreId</i>	Optionnel	String	Identifiant du keystore
<i>returnStatus</i>	Obligatoire	String	Statut de l'opération
<i>errorInfo</i>	Optionnel	String	Message d'erreur en cas d'échec de l'opération

Exemple de réponse :

HTTP/1.1 200 OK

```
{
  "type": "DepositKeystoreResponse",
  "returnStatus": "MSIGN_SRV_STATUS_SUCCESS",
  "keystoreId": "866"
}
```

8 Description des structures complexes

8.1 Structure DocContent

Cette structure permet de retourner le contenu d'un document. Ce contenu doit être encodé en « Base64 » et présenté en « US-ASCII ».

Paramètre	Présence	Type	Description
<i>bytes</i>	Obligatoire	String	Contenu du document encodé en base64

Exemple :

```
"docContent": {  
    "bytes": "JVBER...AA"  
}
```

8.2 Structure ActivationSecret

Cette structure permet de décrire un secret d'activation d'une clé.

Paramètre	Présence	Type	Description
<i>password</i>	Obligatoire	String	Mot de passe de la clé
<i>type</i>	Obligatoire	String	Type de secret d'activation : <ul style="list-style-type: none">• PassphraseActivationSecret• DASActivationSecret• FIDOActivationSecret
<i>dynamicSecret</i>	Conditionnel	String	Secret d'activation dynamique (DAS) encodé en base 64. Il doit être présent si le type de secret d'activation est DASActivationSecret
<i>signatureData</i>	Conditionnel	String	Tableau de réponses encodé en base64. Il doit être présent si le type de secret d'activation est FIDOActivationSecret
<i>clientData</i>	Conditionnel	String	Données client encodées en base64. Il doit être présent si le type de secret d'activation est FIDOActivationSecret
<i>fidoToken</i>	Conditionnel	String	Identifiant du token FIDO à utiliser. Il doit être présent si le type de secret d'activation est FIDOActivationSecret

Exemple :

```
"secret": {  
    "password": "secret",  
    "type": "PassphraseActivationSecret"
```

}

8.3 Structure SignatureProfile

Cette structure permet de décrire un profil de signature.

	Paramètre	Présence	Type	Description
Au choix	<i>profID</i>	Obligatoire	String	Identifiant du profil de signature
	<i>profile</i>	Obligatoire	Profile	Description du profil de signature

Exemples :

```
"signatureProfile": {
    "profID": "14"
}
```

Ou

```
"signatureProfile": {
    "profile": {
        "signaturePolicyOid": "1.0.9.4.2015",
        "attachment": "ENVELOPED",
        "format": "PADES_BES",
        "augmentation": "NONE",
        "signatureAlgoId": "sha256withrsa",
        "archive": "false",
    }
}
```

8.3.1 Structure Profile

Paramètre		Présence	Type	Description
<i>signaturePolicyOID</i>		Optionnel	String	OID de la politique de signature à utiliser
<i>verificationPolicyOID</i>		Optionnel	String	OID de la politique de vérification à utiliser. En cas d'absence, la politique utilisée pour la vérification sera la politique de signature.
<i>attachment</i>		Obligatoire	String	Type d'attachement de la signature : <ul style="list-style-type: none"> • DETACHED (signature détachée) • ENVELOPING (signature enveloppante) • ENVELOPED (signature enveloppée)
<i>format</i>		Obligatoire	String	Format de la signature : <ul style="list-style-type: none"> • CADES_BES • CADES_EPES • XADES_BES • XADES_EPES • PADES_BES • PADES_EPES
Au choix	<i>augmentation</i>	Obligatoire	String	Format d'augmentation de la signature : <ul style="list-style-type: none"> • NONE • T • C • X1 • X2 • XL • A • LTV
	<i>augmentationLevel</i>	Obligatoire	String	Niveau d'augmentation de la signature (EIDAS) : <ul style="list-style-type: none"> • B • T • LT • LTA
1 à n fois	<i>commitments</i>	Optionnel	CommitmentType	Liste des types d'engagements pour la signature
<i>signerRole</i>		Optionnel	SignerRole	Description du rôle du signataire
<i>signatureAlgoId</i>		Optionnel	String	Identifiant de l'algorithme de signature

<i>transformationAlgo</i>	Optionnel	TransformationType	Description des algorithmes de transformation à appliquer
<i>canonicalizationAlgo</i>	Optionnel	String	Description des algorithmes de canonicalisation à appliquer : <ul style="list-style-type: none"> • INCLUSIVE • INCLUSIVE_WITH_COMMENTS • EXCLUSIVE • EXCLUSIVE_WITH_COMMENTS
<i>requireSigningTime</i>	Optionnel	boolean	Indique si une information sur la date supposée de la signature doit être présente dans la signature
<i>requirePlaceOfSignature</i>	Optionnel	boolean	Indique si les informations sur la localisation du signataire doivent être présentes dans la signature
<i>requireContactInfo</i>	Optionnel	boolean	Indique si les informations de contact du signataire doivent être présentes dans la signature (uniquement en cas de signature PAdES)
<i>archive</i>	Optionnel	boolean	Indique si la signature doit être archivée dans le système d'archivage par le serveur de signature
<i>name</i>	Optionnel	String	Indique le nom du profile de signature
<i>signatureProductionPlace</i>	Optionnel	SignatureProductionPlace	Lieu de signature

Exemple :

```

"signatureProfile": {
  "profile": {
    "signaturePolicyOid": "1.0.9.4.2015",
    "attachment": "ENVELOPED",
    "format": "PADES_BES",
    "augmentation": "NONE",

```

```
        "signatureAlgoId": "sha256withrsa",  
        "archive": "false"  
    }  
}
```

8.3.2 Structure TransformationType

Paramètre	Présence	Type	Description
<i>transform</i>	Obligatoire	Array of TransformType	Liste de transformations

Exemple :

```
"transform" : {  
    "XPath" : {  
        "//source"  
    }  
    "Algorithm" : "XPATH"  
}
```

8.3.2.1 Structure TransformType

Paramètre	Présence	Type	Description
<i>XPath</i>	Optionnel	Array of Object	Liste de transformations XPATH
<i>Algorithm</i>	Obligatoire	String	Algorithme de transformation

8.3.3 Structure SignerRole

Paramètre	Présence	Type	Description
<i>role</i>	Optionnel	String	Rôle du signataire
<i>organization</i>	Optionnel	String	Organisation du signataire

Exemple :

```
"signerRole": {
    "role": "Test Signer",
    "organization": "BULL S.A.S."
}
```

8.4 Structure Document

Cette structure permet de définir un document.

	Paramètre	Présence	Type	Description
<i>Au choix</i>	<i>docID</i>	Obligatoire	String	Identifiant du document (du point de vue du serveur de signature)
	<i>docURL</i>	Obligatoire	String	Lien URL (http) permettant d'accéder au document
	<i>docContent</i>	Obligatoire	String	Contenu du document encodé en base64

<i>docHash</i>	Obligatoire	String	contient le Hash du document. Cette méthode est uniquement valable pour le document original ne contenant pas la signature. Cette méthode peut être utilisée lorsqu'un document est « trop volumineux » ou « trop sensible » pour être transmis directement au serveur.
<i>mimeType</i>	Conditionnel	String	Définit le type MIME du document. Ce paramètre est requis lorsque docURL est renseigné.
<i>encoding</i>	Conditionnel	String	Définit l'encodage du contenu. Ce paramètre est requis lorsque docURL est renseigné.

Exemple :

```
"document": {
    "docID": "3514",
    "mimeType": "application/pdf"
}
```

8.5 Structure SignatureOptionalInfos

Cette structure permet de définir les informations optionnelles pour la génération de la signature.

Paramètre	Présence	Type	Description
<i>deActivateAutoTS</i>	Optionnel	boolean	Désactive l'apposition automatique d'un jeton d'horodatage dans une signature PADES simple

<i>signatureProductionPlace</i>		Optionnel	SignatureProductionPlace	Informations concernant le lieu où est réalisée la signature
1 à n fois	<i>commitments</i>	Optionnel	CommitmentType	Liste des types d'engagements pour la signature
<i>contactInfo</i>		Optionnel	String	Informations pour contacter le signataire
<i>visualSignature</i>		Optionnel	VisualSignature	Informations concernant la signature visuelle (PADES uniquement)

8.5.1 Structure VisualSignature

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de visuel de signature : <ul style="list-style-type: none"> VisualSignatureFromFieldIdName VisualSignatureFromPosition
<i>logoBackground</i>	Optionnel	Document	Image du « watermark » qui se trouve en fond de la signature visuelle sur le document signé.
<i>logoLoyalty</i>	Optionnel	Document	Image de représentation de la signature manuscrite à placer sur le document signé.
<i>font</i>	Optionnel	String	Définit la police du texte qui sera affiché dans la signature visuelle. Cette police doit appartenir à la liste des polices utilisables définies par le système sur lequel le serveur de signature est exécuté

certificateObject Info	Optionnel	String	Définit les éléments récupérés du « Distinguished Name » (DN) de l'objet du certificat. Si l'utilisateur souhaite afficher le DN en entier alors il ne doit pas renseigner cette information. L'affichage des éléments se fera selon l'ordre indiqué dans la fonction. Un contrôle syntaxique par expression régulière (Majuscule et/ou minuscule) est réalisé.
displayableElement	Optionnel	List< VisualSignatureDisplayableElement >	Définit un élément à afficher
page	Conditionnel	String	Numéro de la page sur laquelle la signature visible sera insérée selon la nomenclature suivante : En partant de la première page : 0 : première page 1 : deuxième page Etc. En partant de la dernière page : -1 : dernière page -2 : avant-dernière page Etc Obligatoire uniquement si type vaut VisualSignatureFromPosition
yAxis	Conditionnel	String	Distance du rectangle par rapport au bord gauche de la page dans laquelle la signature doit être insérée. Obligatoire uniquement si type vaut VisualSignatureFromPosition
xAxis	Conditionnel	String	Distance du rectangle par rapport au bas de la page dans laquelle la signature doit être insérée Obligatoire uniquement si type vaut VisualSignatureFromPosition
signatureWidth	Conditionnel	String	Hauteur de l'encadré de la signature visuelle dans le document Obligatoire uniquement si type vaut VisualSignatureFromPosition
signatureHeight	Conditionnel	String	Hauteur de l'encadré de la signature visuelle dans le document Obligatoire uniquement si type vaut VisualSignatureFromPosition

signatureFieldName	Conditionnel	String	Nom du champ dans le document PDF représentant l'encadrement de la signature visuelle dans lequel seront positionnées les informations de la signature visuelle. Obligatoire uniquement si type vaut VisualSignatureFromFieldName
--------------------	--------------	--------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

8.5.1.1 Structure VisualSignatureDisplayableElement

Paramètre	Présence	Type	Description
<i>text</i>	Optionnel	String	Définit le texte de l'élément à afficher.
<i>Type</i>	Obligatoire	String	Type aux choix : <ul style="list-style-type: none"> VisualSignatureAliasElt VisualSignatureLocationElt VisualSignatureCommitmentTypeElt VisualSignatureDateElt VisualSignatureDNElt

8.6 Structure SignatureProductionPlace

Paramètre	Présence	Type	Description
<i>City</i>	Optionnel	String	Ville
<i>StateOrProvince</i>	Optionnel	String	Etat ou province
<i>PostalCode</i>	Optionnel	String	Code postal
<i>CountryName</i>	Optionnel	String	Pays

Exemple :

```
"signatureProductionPlace" : {  
    "City" : "Paris",  
    "StateOrProvince" : "Ile-de-France",  
    "PostalCode" : "75000",  
    "CountryName" : "FRANCE"  
}
```

8.7 Structure CommitmentType

Cette structure définit des types d'engagement.

Paramètre	Présence	Type	Description
<i>commitmentIdentifier</i>	Obligatoire	ObjectIdentifierType	Définit un identifiant d'engagement
<i>semantics</i>	Optionnel	String	Définit la sémantique du type d'engagement (ex : « Preuve d'origine »)

8.8 Structure ObjectIdentifierType

Paramètre	Présence	Type	Description
<i>Identifier</i>	Obligatoire	IdentifierType	Définit un type d'engagement

8.8.1 Structure IdentifierType

Paramètre	Présence	Type	Description
<i>anyURI</i>	Optionnel	String	Définit l'URI du type d'engagement

<i>Qualifier</i>	Optionnel	QualifierType	Définit le type de qualifier
------------------	-----------	-------------------------------	------------------------------

8.8.2 Structure QualifierType

Paramètre	Présence	Type	Description
<i>value</i>	Optionnel	String	Type de qualifier : <ul style="list-style-type: none">• OIDASURI• OIDASURN

8.9 Structure SignatureReport

Cette structure permet de définir le rapport de vérification renvoyé par le serveur de signature.

Paramètre	Présence	Type	Description
<i>signatureNumber</i>	Optionnel	String	Indique l'indice de la signature qui a été vérifiée. (dans le cas où le document dispose de plusieurs signatures).
<i>report</i>	Obligatoire	RapportType	Rapport de vérification

8.10 Structure SignVerifReport

Cette structure permet de définir le résultat de la vérification (sur toutes les signatures du document) renvoyé par le serveur de signature.

Paramètre	Présence	Type	Description
<i>signatureReport</i>	Obligatoire	Array of SignatureReport	Résultat de la vérification des signatures du document

<i>status</i>	Optionnel	Boolean	Statut du résultat
---------------	-----------	---------	--------------------

8.11 Structure InfoFile

Cette structure permet de définir des informations sur un fichier.

Paramètre	Présence	Type	Description
<i>descriptions</i>	Obligatoire	Liste de Description	Description de l'élément
<i>keyWords</i>	Obligatoire	Liste de Description	Mots-clés pour identifier l'élément

Exemple :

```
"infoFile" :  
  {  
    "descriptions" :  
    {  
      "lang" : "fr",  
      "description" : "Sign Server Test policy 2019"  
    },  
    "keyWords" :  
    {  
      "lang" : "fr",  
      "description" : "TEST"  
    }  
  }
```

8.11.1 Structure Description

Paramètre	Présence	Type	Description
<i>lang</i>	Obligatoire	String	Langue utilisée (ex : FR ou EN)
<i>description</i>	Obligatoire	String	Description de l'élément

8.12 Structure SignaturePolicy

Cette structure permet de définir une politique de signature.

Paramètre	Présence	Type	Description
<i>SignPolicyDigestAlg</i>	Obligatoire	Algorithm	Algorithme utilisé pour le calcul du hash de la politique
<i>Transforms</i>	Optionnel	Transform	Description des transformations à appliquer sur la politique avant calcul du hash
<i>SignPolicyInfo</i>	Obligatoire	SignPolicyInfo	Informations sur la politique de signature
<i>SignPolicyDigest</i>	Obligatoire	String	Hash de la politique encodé en base64

8.12.1 Structure Algorithm

Paramètre	Présence	Type	Description
-----------	----------	------	-------------

<i>content</i>	Optionnelle	Liste d'objets	Liste d'objets décrivant l'algorithme
<i>Algorithm</i>	Obligatoire	URI	URI décrivant l'algorithme (ex : http://www.w3.org/2001/04/xmlenc#sha256)

Exemple :

"SignPolicyDigestAlg":

```
{
  "Algorithm": "http://www.w3.org/2001/04/xmlenc#sha256"
}
```

8.12.2 Structure Transform

Paramètre	Présence	Type	Description
<i>Transform</i>	Obligatoire	Liste de TransformType	Liste d'objets décrivant des transformations

Exemple :

"Transforms":

```
{
  "Transform":
  [
    {
      "Algorithm": "http://www.w3.org/TR/2001/REC-xml-c14n-20010315"
    }
  ]
}
```

8.12.2.1 Structure TransformType

Paramètre	Présence	Type	Description
<i>content</i>	Optionnelle	Liste d'objets	Liste d'objets décrivant la transformation
<i>Algorithm</i>	Obligatoire	URI	URI décrivant l'algorithme de transformation (ex : http://www.w3.org/2001/04/xmlenc#sha256)

8.12.3 Structure SignPolicyInfo

Paramètre	Présence	Type	Description
<i>SignPolicyIdentifier</i>	Obligatoire	ObjectIdentifierType	OID de la politique
<i>FieldOfApplication</i>	Obligatoire	String	Champ d'application de la politique
<i>DateOfIssue</i>	Optionnelle	Date	Date de création de la politique
<i>PolicyIssuerName</i>	Optionnelle	String	Nom du créateur de la politique
<i>SignatureValidationPolicy</i>	Obligatoire	SignatureValidationPolicy	Description pour la validation

8.12.3.1 Structure SignatureValidationPolicy

Paramètre	Présence	Type	Description
-----------	----------	------	-------------

<i>SigningPeriod</i>	Obligatoire	TimePeriod	Description des dates de validité de la politique
<i>CommonRules</i>	Conditionnel	CommonRulesType	Règles applicables à tous les types d'engagements
<i>CommitmentRules</i>	Conditionnel	CommitmentRulesListType	Règles applicables à des types d'engagements spécifiques

Le traitement de *CommonRules* et *CommitmentRules* est décrit dans le document MSIGN-PS-02 §7.2.

8.12.3.2 Structure TimePeriod

Paramètre	Présence	Type	Description
<i>NotBefore</i>	Obligatoire	Date	Date à partir de laquelle la politique peut être utilisée
<i>NotAfter</i>	Optionnelle	Date	Date à partir de laquelle la politique ne peut plus être utilisée pour la génération de signatures

Exemple :

"SigningPeriod":

```
{
  "NotBefore": "2014-04-09T10:00:00.000+02:00"
  "NotAfter": "2054-04-09T10:00:00.000+02:00"
}
```

8.12.3.3 Structure CommonRulesType

Paramètre	Présence	Type	Description
-----------	----------	------	-------------

<i>SignerAndVerifierRules</i>	Optionnelle	SignerAndVerifierRulesType	Définition des règles applicables au signataire et au vérificateur
<i>SigningCertTrustCondition</i>	Optionnelle	SigningCertTrustConditionType	Définition des règles applicables au certificat du signataire
<i>TimeStampTrustCondition</i>	Optionnelle	TimeStampTrustConditionType	Définition des règles applicables aux certificats des unités d'horodatage
<i>AlgorithmConstraintSet</i>	Optionnelle	AlgorithmConstraintSetType	Définition des contraintes sur les algorithmes
<i>SignPolExtensions</i>	Optionnelle	SignPolExtensionsListType	Définition d'un ensemble d'extensions propriétaires

8.12.3.4 Structure SignerAndVerifierRulesType

Paramètre	Présence	Type	Description
<i>SignerRules</i>	Optionnelle	SignerRulesType	Règles applicables à la génération de la signature
<i>VerifierRules</i>	Optionnelle	VerifierRulesType	Règles applicables à la vérification de la signature

8.12.3.5 Structure SignerRulesType

Paramètre	Présence	Type	Description
<i>MandatedSignedQProperties</i>	Obligatoire	QPropertiesListType	Extensions signées qui doivent être présentes dans la signature générée

Exemple :

```
"SignerRules":  
  
{  
  
  "MandatedSignedQProperties":  
  
  {  
  
    "QPropertyID":["urn:oid:1.2.840.113549.1.9.16.2.47"]  
  
  }  
  
}
```

8.12.3.6 Structure QPropertiesListType

Paramètre	Présence	Type	Description
<i>QPropertyID</i>	Obligatoire	Liste d'URI	Liste des extensions (voir MSIGN-PS-02 §4.4.2.1 & §4.4.2.2)

8.12.3.7 Structure VerifierRulesType

Paramètre	Présence	Type	Description
<i>MandatedQUnsignedProperties</i>	Obligatoire	QPropertiesListType	Extensions non signées qui doivent être présentes dans la signature à vérifier

8.12.3.8 Structure SigningCertTrustConditionType

Paramètre	Présence	Type	Description
<i>SignerTrustTrees</i>	Obligatoire	CertificateTrustTreesType	Liste de certificats d'AC autorisés et règles applicable à l'ensemble du chemin de certification
<i>SignerRevReq</i>	Optionnelle	CertificateRevReqType	Conditions de vérification de la non-révocation

8.12.3.9 Structure CertificateTrustTreesType

Paramètre	Présence	Type	Description
<i>CertificateTrustPoint</i>	Obligatoire	Liste de CertificateTrustPointType	Règles applicables à un chemin de certification

8.12.3.10 Structure CertificateTrustPointType

Paramètre	Présence	Type	Description
<i>TrustPoint</i>	Obligatoire	String	Certificat de l'AC racine du chemin de certification encodé en base64
<i>PathLenConstraint</i>	Optionnelle	Integer	Longueur maximale de la chaîne de certification
<i>AcceptablePolicySet</i>	Optionnelle	AcceptablePolicyListType	Identifiants des politiques de certification autorisés
<i>NameConstraints</i>	Optionnelle	NameConstraintsType	Liste des noms d'autorités intermédiaires autorisés

8.12.3.11 Structure AcceptablePolicyListType

Paramètre	Présence	Type	Description
<i>AcceptablePolicy</i>	Obligatoire	Liste de ObjectIdentifierType	Liste d'OID décrivant des politiques

8.12.3.12 Structure NameConstraintsType

Paramètre	Présence	Type	Description
-----------	----------	------	-------------

<i>PermittedSubtrees</i>	Optionnelle	GeneralSubTreesListType	Liste des noms d'autorités intermédiaires autorisés
<i>ExcludedSubtrees</i>	Optionnelle	GeneralSubTreesListType	Liste des noms d'autorités intermédiaires interdits

8.12.3.13 Structure GeneralSubTreesListType

Paramètre	Présence	Type	Description
<i>GeneralSubTree</i>	Obligatoire	GeneralSubTreeType	Description d'un nom

8.12.3.14 Structure GeneralSubTreeType

Paramètre	Présence	Type	Description
<i>Base</i>	Obligatoire	String	Sous-ensemble du nom
<i>Minimum</i>	Obligatoire	Integer	Longueur minimale du nom
<i>Maximum</i>	Obligatoire	Integer	Longueur maximale du nom

8.12.3.15 Structure CertificateRevReqType

Paramètre	Présence	Type	Description
<i>EndRevReq</i>	Obligatoire	RevocationReqType	Définition de la méthode de vérification de la non-révocation à utiliser pour les certificats finaux

CACerts	Obligatoire	RevocationReqType	Définition de la méthode de vérification de la non-révocation à utiliser pour les certificats d'autorité racine
---------	-------------	-----------------------------------	-----------------------------------------------------------------------------------------------------------------

8.12.3.16 Structure RevocationReqType

Paramètre	Présence	Type	Description
<i>EnuRevReq</i>	Obligatoire	String	Définition de la méthode de vérification de la non-révocation : <ul style="list-style-type: none"> • <i>clrcheck</i> (CRL) • <i>ocspcheck</i> (OCSP) • <i>bothcheck</i> (CRL et OCSP) • <i>eithercheck</i> (CRL ou OCSP) • <i>nocheck</i> (pas de vérification) • <i>other</i> (autre)

Exemple :

```
{
  "EnuRevReq": "clrcheck"
}
```

8.12.3.17 Structure TimeStampTrustConditionType

Paramètre	Présence	Type	Description
<i>TtsCertificateTrustTrees</i>	Optionnelle	CertificateTrustTreesType	Liste de certificats d'autorité racine autorisés
<i>TtsRevReq</i>	Optionnelle	CertificateRevReqType	Conditions de vérification de la non-révocation

<i>CautionPeriod</i>	Optionnelle	DeltaTimeType	Période de grâce (intervalle de temps autorisé entre la date de génération de la signature et la date d'augmentation de la signature)
<i>SignatureTimeStampDelay</i>	Optionnelle	DeltaTimeType	délai maximum entre la date de génération de la signature et l'apposition du jeton d'horodatage

8.12.3.18 Structure DeltaTimeType

Paramètre	Présence	Type	Description
<i>DeltaSeconds</i>	Obligatoire	Integer	Delta en secondes
<i>deltaMinutes</i>	Obligatoire	Integer	Delta en minutes
<i>DeltaHours</i>	Obligatoire	Integer	Delta en heures
<i>DeltaDays</i>	Obligatoire	Integer	Delta en jours

8.12.3.19 Structure AlgorithmConstraintSetType

Paramètre	Présence	Type	Description
<i>SignerAlgConstraints</i>	Optionnelle	AlgConstraintsListType	contraintes sur les algorithmes utilisés pour la génération de la signature
<i>EeCertAlgConstraints</i>	Optionnelle	AlgConstraintsListType	contraintes sur les algorithmes utilisés pour les certificats des signataires

<i>CACertAlgConstraints</i>	Optionnelle	AlgConstraintsListType	contraintes sur les algorithmes utilisés pour les certificats d'Autorités de Confiance
<i>TSACertAlgConstraints</i>	Optionnelle	AlgConstraintsListType	contraintes sur les algorithmes utilisés pour les certificats des unités d'horodatage

8.12.3.20 Structure AlgConstraintsListType

Paramètre	Présence	Type	Description
<i>AlgAndLength</i>	Obligatoire	Liste de AlgAndLength	Description des contraintes

8.12.3.21 Structure AlgAndLength

Paramètre	Présence	Type	Description
<i>AlgId</i>	Obligatoire	URI	algorithme de signature (cf MSIGN-PS-02 §4.4.5)
<i>MinKeyLength</i>	Optionnelle	Integer	taille minimum de clé pour les algorithmes de signature

8.12.3.22 Structure SignPolExtensionListType

Paramètre	Présence	Type	Description
<i>SignPolExtension</i>	Obligatoire	Liste de AnyType	Définition de balises propriétaires

Exemple :

"SignPolExtension":

[

```
{
  "MetaSignPolExtension":
  {
    "PolicyValidation":
    {
      "SigningCertTrustConditionOid": "1.0.1",
      "TimeStampTrustConditionOid": "1.0.2"
    },
    "SemanticInvariance": {"isOptional": "true"}
  }
}
```

8.12.3.23 Structure CommitmentRulesListType

Paramètre	Présence	Type	Description
<i>CommitmentRule</i>	Obligatoire	CommitmentRuleType	Règles sur les types d'engagements

8.12.3.24 Structure CommitmentRuleType

Paramètre	Présence	Type	Description
<i>SelCommitmentTypes</i>	Obligatoire	SelectedCommitmentTypeList	définit la liste des types d'engagement autorisés
<i>SignerAndVerifierRules</i>	Optionnelle	SignerAndVerifierRulesType	définit les règles applicables au signataire et au vérificateur
<i>SigningCertTrustCondition</i>	Optionnelle	SigningCertTrustConditionType	définit les règles applicables au certificat du signataire

<i>TimeStampTrustCondition</i>	Optionnelle	TimeStampTrustConditionType	définit les règles applicables aux certificats des unités d'horodatage utilisées pour générer les jetons d'horodatage
<i>AlgorithmConstraintSet</i>	Optionnelle	AlgorithmConstraintSetType	définit les contraintes sur les algorithmes
<i>SignPolExtensions</i>	Optionnelle	SignPolExtensionsListType	définit un ensemble d'extensions propriétaires

8.12.3.25 Structure SelectedCommitmentTypeList

Paramètre	Présence	Type	Description
<i>SelCommitmentType</i>	Obligatoire	Liste de SelectedCommitmentType	Engagements autorisés

8.12.3.26 Structure SelectedCommitmentType

Paramètre	Présence	Type	Description
<i>Empty</i>	Optionnelle	Object	signifie que les règles s'appliquent aux signatures électroniques qui ne possèdent pas de type d'engagement
<i>RecognizedCommitmentType</i>	Optionnelle	CommitmentType	permet de définir la liste des types d'engagement pour lesquels les règles de la politique s'appliquent

8.13 Structure ConstraintChain

Paramètre	Présence	Type	Description
<i>maxResults</i>	Optionnelle	Integer	Nombre maximal de résultats retournés
<i>type</i>	Obligatoire	String	Type de contrainte : <ul style="list-style-type: none"> • StringConstraint • BooleanConstraint • ListIdConstraint • RangeIdConstraint
<i>constrainedElement</i>	Obligatoire	String	Element sur lequel porte la contrainte
<i>isNegated</i>	Optionnel	Boolean	Si positionné à true, la contrainte est négative
<i>typeStringConstraint</i>	Conditionnel	String	Type de contrainte (uniquement pour StringConstraint) : <ul style="list-style-type: none"> • EQUALS • CONTAINS • BEGINS_WITH • ENDS_WITH
<i>value</i>	Conditionnel	String si StringConstraint Boolean si BooleanConstraint	Valeur à tester. Uniquement pour StringConstraint et BooleanConstraint
<i>ids</i>	Conditionnel	Liste de String	Liste des identifiants (uniquement pour ListIdConstraint)
<i>from</i>	Conditionnel	String	(uniquement RangeIdConstraint) pour
<i>to</i>	Conditionnel	String	(uniquement RangeIdConstraint) pour

8.14 Structure SigPolInfo

Paramètre	Présence	Type	Description
<i>policyOid</i>	Obligatoire	String	OID de la politique
<i>policyName</i>	Optionnelle	String	Nom de la politique
<i>isLocked</i>	Obligatoire	boolean	Booléen à true si la politique ne peut plus être modifiée

Exemple :

```
{  
  
  "policyOid": "1.0.9.4.2015",  
  
  "policyName": "Test policy oid urn:oid:1.0.9.4.2015",  
  
  "isLocked": false  
}
```

8.15 Structure SkGenerationProfileSpec

Paramètre	Présence	Type	Description
<i>kpAlgo</i>	Obligatoire	KPType	Description du type de clé
<i>algoParameters</i>	Optionnelle	AlgoParameters	Définition des algorithmes utilisés pour la création de la clé
<i>enforcedAuthentication</i>	Optionnelle	EnforcedAuthenticationMethod	Paramètres d'authentification
<i>keyUsage</i>	Optionnelle	KeyUsageArray	Usages de clé
<i>name</i>	Obligatoire	String	Nom du profil de clé

8.15.1 Structure KPType

Paramètre	Présence	Type	Description
<i>oid</i>	Optionnelle	OID	Type de la clé
<i>uri</i>	Optionnelle	URI	Type de la clé
<i>name</i>	Optionnelle	String	Type de la clé

8.15.2 Structure AlgoParameters

Paramètre	Présence	Type	Description
<i>parameter</i>	Optionnelle	Liste de Parameter	Element contenant la paire de clé correspondant à un paramètre

Exemple :

```
"parameter": {  
    "key": "KEY_LEN",  
    "value": "2048"  
}
```

8.15.2.1 Structure Parameter

Paramètre	Présence	Type	Description
<i>key</i>	Obligatoire	String	Clé identifiant le paramètre
<i>value</i>	Optionnelle	String	Valeur du paramètre

8.15.3 Structure EnforcedAuthenticationMethod

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type d'authentification : <ul style="list-style-type: none">• FIDOAuthentication• DASAAuthentication
<i>counterMax</i>	Optionnelle	Integer	Nombre maximal d'essais du mot de passe dynamique avant que le recalcul d'un challenge soit nécessaire
<i>authDataLength</i>	Optionnelle	Integer	Taille des données d'authentification vérifiées. Uniquement pour DASAAuthentication
<i>authTryMax</i>	Optionnelle	Integer	Nombre maximal d'essais erronés acceptés avant blocage de la clé
<i>activationPeriod</i>	Optionnelle	Integer	Temps maximal d'activation de la clé

<i>blockingPeriod</i>	Optionnelle	Integer	Temps maximal de blocage de la clé
<i>fidoToken</i>	Conditionnelle	String	Identifiant du token FIDO. Obligatoire si <i>type=FIDOAuthentication</i> . Ne doit pas être présent sinon.

8.15.4 Structure KeyUsageArray

Cet objet représente la liste des usages de clés pour la génération des certificats

Paramètre	Présence	Type	Description
digitalSignature	Optionnelle	Boolean	Booléen à True si l'usage de clé digitalSignature doit être présent. False si il ne doit pas être présent. Null si il peut être présent ou non.
contentCommitment	Optionnelle	Boolean	Booléen à True si l'usage de clé contentCommitment doit être présent. False si il ne doit pas être présent. Null si il peut être présent ou non.
keyEncipherment	Optionnelle	Boolean	Booléen à True si l'usage de clé keyEncipherment doit être présent. False si il ne doit pas être présent. Null si il peut être présent ou non.
dataEncipherment	Optionnelle	Boolean	Booléen à True si l'usage de clé dataEncipherment doit être présent. False si il ne doit pas être présent. Null si il peut être présent ou non.
keyAgreement	Optionnelle	Boolean	Booléen à True si l'usage de clé keyAgreement doit être présent. False si il ne doit pas être présent. Null si il peut être présent ou non.
keyCertSign	Optionnelle	Boolean	Booléen à True si l'usage de clé keyCertSign doit être présent. False si il ne doit pas être présent. Null si il peut être présent ou non.

crlSign	Optionnelle	Boolean	Booléen à True si l'usage de clé crlSign doit être présent. False si il ne doit pas être présent. Null si il peut être présent ou non.
encipherOnly	Optionnelle	Boolean	Booléen à True si l'usage de clé encipherOnly doit être présent. False si il ne doit pas être présent. Null si il peut être présent ou non.
decipherOnly	Optionnelle	Boolean	Booléen à True si l'usage de clé decipherOnly doit être présent. False si il ne doit pas être présent. Null si il peut être présent ou non.

8.16 Structure ServerId2NameInfo

Paramètre	Présence	Type	Description
<i>serverId</i>	Optionnelle	String	Identifiant de l'objet
<i>name</i>	Optionnelle	String	Nom de l'objet

8.17 Structure CertificateRequestParameters

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de requête (SCEPRequestParameters)
<i>challengePassword</i>	Optionnel	String	Spécification d'un attribut « challengePassword » à envoyer
<i>transactionID</i>	Optionnel	String	Identifiant de transaction générée par le client

8.18 Structure RequestCertificateResult

Paramètre	Présence	Type	Description
-----------	----------	------	-------------

<i>status</i>	Obligatoire	String	Statut de la réponse renvoyée par le provider de certificat
<i>X509Certificate</i>	Optionnel	String	Le certificat de signature (encodé en base64) renvoyé par le provider

8.19 Structure ActivationData

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de données : <ul style="list-style-type: none"> DASActivationData FIDOActivationData
<i>userActivationSecret</i>	Obligatoire	ActivationSecret	Secret d'activation
<i>authenticationMethod</i>	Optionnel	String	Méthode d'authentification : <ul style="list-style-type: none"> BASIC_OTP (avec DASActivationData) OTP_WITH_HASH (avec DASActivationData) FIDO_U2F (seulement avec FIDOActivationData type)
<i>keyHandle</i>	Conditionnel	String	Handle de la clé (encodé en base64). Obligatoire pour le type FIDOActivationData. Ne doit pas être présent sinon

Exemple :

```
{
  "type": "DASActivationData",
  "userActivationSecret": "c2VjcmV0",
  "authenticationMethod" : "BASIC_OTP"
}
```

8.20 Structure ActivationResult

Paramètre	Présence	Type	Description
<i>dynamicChallenge</i>	Optionnelle	String	Challenge encodé en base64 (DASActivationResult)

<i>type</i>	Optionnelle	String	Type de données d'activation (DASActivationResult, FIDOActivationResult)
<i>registeredTokens</i>	Optionnelle	Liste de String	Liste de token FIDO encodés en base64 (FIDOActivationResult)

8.21 Structure SkGenerationProfileRef

Paramètre	Présence	Type	Description
<i>skgProfileId</i>	Mandatory	String	Identifiant du profil de clé de signature

8.22 Structure PasswordActivationSecret

Paramètre	Présence	Type	Description
<i>bytes</i>	Obligatoire	String	Password encodé en base64

8.23 Structure Role

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de description de rôle : <ul style="list-style-type: none"> RoleSpecification (description des caractéristiques d'un rôle) RoleReference (référence à un rôle existant)
<i>roletype</i>	Obligatoire	String	Type de rôle : <ul style="list-style-type: none"> Signer SignManager ServerManager Auditor
<i>spec</i>	Optionnel	Role De type RoleSpecification	Description des caractéristiques du rôle (uniquement si type= RoleReference)
<i>secret</i>	Conditionnel	ActivationSecret	Secret de l'utilisateur si il est de type Signer
<i>roleNS</i>	Conditionnel	String	Namespace associé au rôle. (Obligatoire uniquement si type=RoleReference) Ex : <code>http://www.bull.security.com/SignServer/extensions/v1.3.0/#SignManager</code>
<i>id</i>	Optionnel	String	Identifiant de l'utilisateur (si il existe)

Exemple :

```
{
  "type": "RoleSpecification",
    "roletype" : "Signer",
    "secret": {
      "password": "cGFzc3dvcmQ=",
      "type": "PassphraseActivationSecret"
    }
}
```

8.24 Structure Credential

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type du mode d'authentification : <ul style="list-style-type: none"> • CredentialReference • PasswordWrapper • X509Certificate
<i>credentialId</i>	Optionnel	CredentialId	Identifiant du mode d'authentification (uniquement si type=CredentialReference)
<i>spec</i>	Optionnel	Credential De type CredentialSpecification	Description des caractéristiques du mode d'authentification (uniquement si type=CredentialReference)
<i>id</i>	Optionnel	String	Identifiant de l'utilisateur du mode d'authentification
<i>hashAlgo</i>	Optionnel	String	Algorithme de hachage utilisé pour calculer le hash du mot de passe (si type=PasswordWrapper)
<i>credentialValue</i>	Optionnel	String	Mot de passe encodé en base64 si type=PasswordWrapper. Certificat encodé en base64 si type=X509Certificate

Exemples :

```
{
  "type": "PasswordWrapper",
  "credentialValue": "XohImNooBHFR00VvjcyPj3NgPQ1qq73WKhHvch0VQtg=",
  "hashAlgo": "SHA-256"
```

```

}

{
    "type": "X509Certificate",
    "credentialValue": "MIIE7..."
}

```

8.24.1 Structure CredentialId

Paramètre	Présence	Type	Description
<i>spec</i>	Optionnel	Credential De type CredentialSpecification	Description des caractéristiques du mode d'authentification
<i>credentialKey</i>	Optionnel	String	Identifiant unique du mode d'authentification associé à l'utilisateur
<i>id</i>	Optionnel	String	Identifiant de l'utilisateur

8.24.2 Structure CredentialDescription

Paramètre	Présence	Type	Description
<i>password</i>	Optionnel	String	Mot de passe pour l'authentification
<i>credentialKey</i>	Mandatory	String	Identifiant unique du mode d'authentification associé à l'utilisateur
<i>x509Certificate</i>	Optionnel	String	Certificat d'authentification encodé en base64

8.25 Structure Group

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de description du groupe : <ul style="list-style-type: none"> GroupReference GroupSpecification
<i>id</i>	Optionnel	String	Identifiant du groupe

<i>spec</i>	Optionnel	Group De type GroupSpecification	Description des caractéristiques du groupe (uniquement si type= GroupReference)
<i>linkedGroups</i>	Optionnel	Liste de Group	Ensemble des groupes liés à celui-ci
<i>linkedUsers</i>	Optionnel	Liste de User ou Application	Ensemble des utilisateurs liés à ce groupe
<i>name</i>	Optionnel	String	Nom du groupe

8.26 Structure User

Paramètre	Présence	Type	Description
<i>name</i>	Obligatoire	String	Nom de l'utilisateur
<i>type</i>	Obligatoire	String	Type de description de l'utilisateur : <ul style="list-style-type: none"> • UserSpecification • UserReference
<i>roles</i>	Optionnel	List< Role >	Liste des rôles de l'utilisateur
<i>credentials</i>	Obligatoire	List< Credential >	Liste des modes d'authentification de l'utilisateur (certificats, mots de passe)
<i>groups</i>	Optionnel	List< Group >	Liste des groupes auxquels appartient l'utilisateur
<i>applications</i>	Optionnel	List< Application >	Liste des applications reconnues par l'utilisateur

8.27 Structure Application

Paramètre	Présence	Type	Description
<i>type</i>	Obligatoire	String	Type de description du groupe : <ul style="list-style-type: none"> • ApplicationReference • ApplicationSpecification
<i>id</i>	Optionnel	String	Identifiant de l'application
<i>spec</i>	Optionnel	Application De type ApplicationSpecification	Description des caractéristiques de l'application (uniquement si type= ApplicationReference)
<i>name</i>	Obligatoire	String	Nom de l'application
<i>roles</i>	Optionnel	List< Role >	Liste des rôles de l'application
<i>credentials</i>	Obligatoire	List< Credential >	Liste des modes d'authentification de l'application

<i>groups</i>	Optionnel	List< Group >	Liste des groupes auxquels appartient l'application
<i>applications</i>	Optionnel	List< Application >	Liste des applications reconnues par l'application

8.28 Structure CertificateInfo

Paramètre	Présence	Type	Description
<i>certificate</i>	Obligatoire	String	Certificat encodé en Base64
<i>authMethod</i>	Optionnel	String	Méthode d'authentification
<i>signatureKey</i>	Optionnel	String	Identifiant de la clé de signature

8.29 Structure ServerId

Paramètre	Présence	Type	Description
<i>serverId</i>	Optionnelle	String	Identifiant de l'objet
<i>name</i>	Optionnelle	String	Nom de l'objet

8.30 Structure UsersToGroupAssociation

Paramètre	Présence	Type	Description
<i>userIds</i>	Obligatoire	Liste de String	Liste d'identifiants d'utilisateurs
<i>groupId</i>	Obligatoire	String	Identifiant du groupe

8.31 UserResponse

Paramètre	Présence	Type	Description
<i>name</i>	Obligatoire	String	Nom de l'utilisateur
<i>certificatesInfo</i>	Optionnelle	List< CertificateInfo >	Liste d'informations sur les certificats de signature de l'utilisateur
<i>roles</i>	Optionnelle	List< RoleDescription >	Liste des rôles de l'utilisateur

<i>credentials</i>	Obligatoire	List< CredentialDescription >	Liste des modes d'authentification de l'utilisateur
<i>groups</i>	Optionnelle	List< ServerId2NameInfo >	Liste de la description des groupes de l'utilisateur
<i>trustedApps</i>	Optionnelle	List< ServerId2NameInfo >	Liste des applications de confiance pour l'utilisateur

8.31.1 RoleDescription

Paramètre	Présence	Type	Description
<i>roleNS</i>	Obligatoire	String	Le rôle de l'utilisateur

Fin du document

