

Service E-parapheur

EparapheurCallService

Permet d'exécuter l'appel de fonctionnalités relatives aux services web des e-parapheur suivants :

- I-parapheur
- Ixbus
- Contrat Privé
- Xparaph

Fonctionnalités actuellement disponibles :

- Dépôt d'un document dans le parapheur
- Récupération de l'état d'un document envoyé au parapheur
- Récupération de la liste des parapheurs Multigest disponibles

```
string EparapheurCallService (  
    string user,  
    string method,  
    String account,  
    String params  
);
```

Paramètres

user : Login de l'utilisateur Multigest

method : Nom de la fonctionnalité appelée.

Méthodes disponibles :

- *sendDocument* => Dépôt d'un document dans le parapheur
- *getDocumentState* => Récupération de l'état d'un document envoyé au parapheur
- *getParapheurAccounts* => Liste des parapheurs Multigest disponibles

account : Nom du compte parapheur Multigest utilisé

params : Paramètres transmis lors de l'appel de la méthode. Chaîne au format JSON

Codes retour

Chaîne au format JSON fonction de la méthode appelée.

Descriptif des paramètres à transmettre (@params) et retour (@return) associé :

Liste des méthodes :

– **sendDocument** :

@params:

```
{
  doc = Document à déposer
  customDatas= Données personnalisées à transmettre au parapheur
  annexes= Liste des documents à transmettre en annexes
  engine= Outil parapheur à utiliser si paramètre accountvide
}
```



Information

@params obligatoire : doc

@params facultatif : customDatas, annexes , engine

@return: Chaîne JSON

```
{
  id => identifiant du dossier créé dans le parapheur ou vide si échec
  message => message d'erreur retourné si échec
}
```

– **getDocumentState** :

@params :

```
{
  doc = Document à interroger
  engine = Outil parapheur à utiliser si paramètre account vide
}
```



Information

@params obligatoire : doc

@params facultatif : engine

@return: Chaîne JSON

```
{
  state => 0 = en cours, 1 = terminé
  statusText => Texte de l'étape de validation du parapheur
  validation => Validation effectué dans le parapheur. 1 = signé\validé, 0 =
rejeté
}
```

```
    xml => Archive zip du XML de retour du parapheur stocké en GED encodé en  
base64  
}
```

– ***getParapheurAccounts*** :

@params : vide

@return : Chaîne JSON

Tableau d'objet JSON

```
[  
  {  
    id => identifiant du compte  
    libelle => nom du compte  
    engine => outil parapheur associé au compte  
  }  
]
```

Descriptif des propriétés de @params :

- doc => JSON contenant des informations sur le doc (cf. @doc ci-dessous)
- customDatas => Tableau associatif
- annexes => Tableau d'objet @doc
- engine => string nom de l'outil parapheur à utiliser si *account* est vide

@doc:

```
{  
  uid => Identifiant unique du document  
}
```

Ou

@doc:

```
{  
  iddoc => Identifiant du document dans l'armoire  
  repository => Nom physique ou identifiant de l'armoire  
}
```