



# oodrive \_sign

## DESCRIPTION DES CALLBACKS STANDARDS

Révision 2.0





# PRÉAMBULE

**oodrive\_sign** Solution de signature électronique, vous propose un callback pour permettre la connexion de **oodrive\_sign** avec votre Système d'Information.

Cette documentation vous présentera les différentes possibilités d'intégration de callbacks et les éléments à mettre en place de votre côté.



# SOMMAIRE

**01.** DECLENCHEMENT DE CALLBACKS

**02.** CALLBACK FTP/SFTP

**03.** CALLBACK GET

**04.** CALLBACK POST JSON

SUIVI DES SIGNATURES

SUIVI DU CONTRAT

**05.** CALLBACK POST MULTIPART

SUIVI DES SIGNATURES

SUIVI DU CONTRAT

**06.** GESTION DES ERREURS

SUPPORT D'AIDE

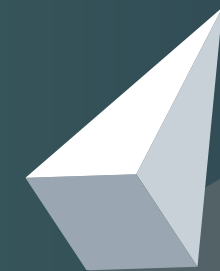
GARDONS LE LIEN

CONTACT



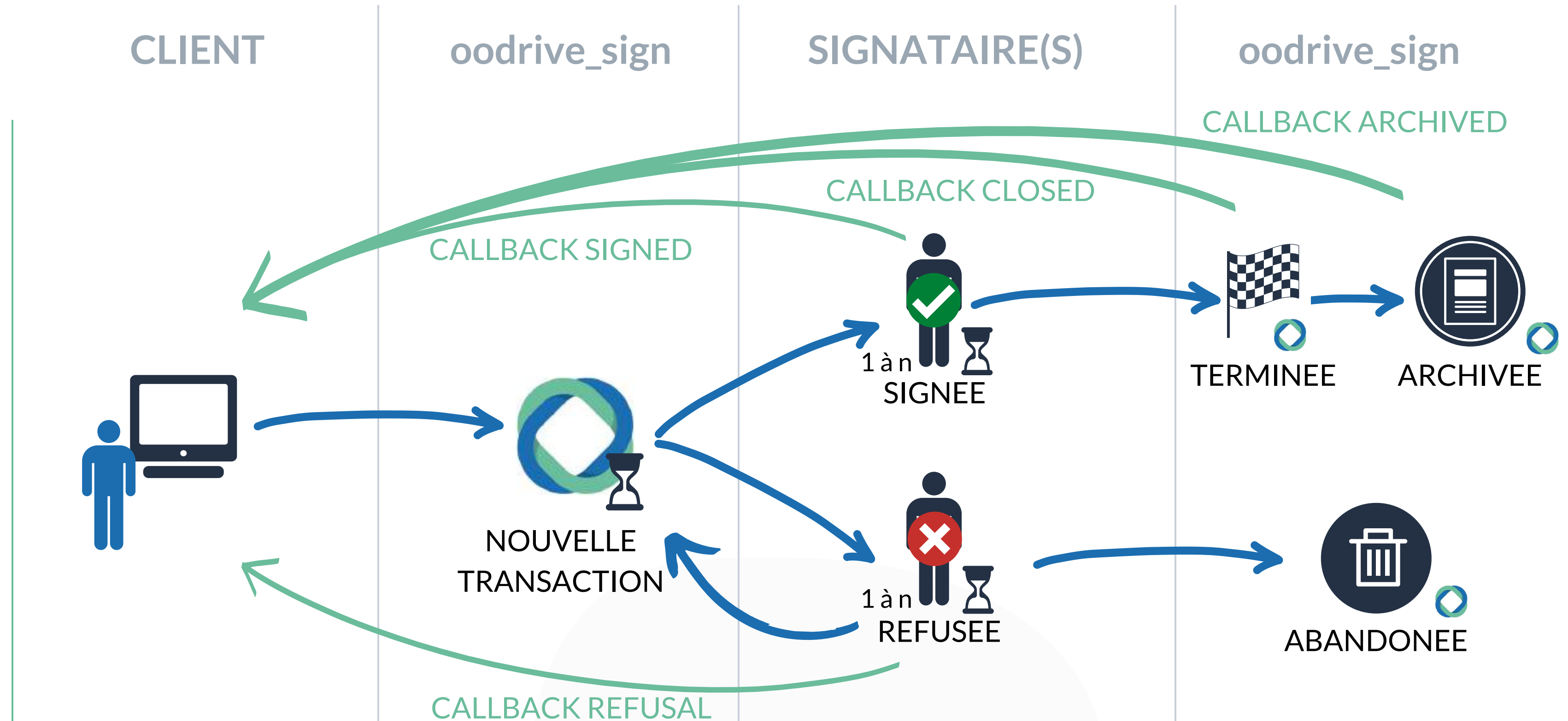
# DECLenchement DE CALLBACKS

01



# DECLenchement DE CALLBACKS

Un callback est déclenché pour chaque état de la transaction : refusé, signé, fermé, archivé.



# CALLBACK FTP/SFTP



# CALLBACK FTP/SFTP

Pour mettre cela en place, vous devez mettre à disposition un serveur ftp ou sftp sur lequel nous pourrions déposer les contrats au format pdf.

Nous attendons pour la mise en place de ce callback :

- hostname (éventuellement le port) **accessible publiquement**
- username
- password
- path

# CALLBACK GET





# CALLBACK GET

C'est le callback le plus simple. Son objectif est de vous notifier que le contrat est archivé. La réponse obtenue sera sous la forme d'une url contenant en query string :

`?contractId=[ID]`

Nous attendons, pour mettre en place ce callback, une url **fonctionnelle et accessible publiquement**.

Les méthodes d'authentification suivantes sont disponibles pour ce callback :

- Basic Auth
- Token
- OAuth

# CALLBACK POST JSON

# 04



# CALLBACK POST JSON

Ce callback vous permet de recevoir et de suivre l'avancée des contrats durant tout le processus de signature.

Nous attendons, pour mettre en place ce callback, une url **fonctionnelle et accessible publiquement**.

# SUIVI DES SIGNATURES

Le callback sera déclenché pour chaque signature, ou refus de signature, pour chaque signataire.

En cas de signature, les données transmises seront sous la forme JSON suivante :

```
{  
  type : "SIGNATURE",  
  contractId : "id",  
  contractorId : "ContractorID",  
  signatureStatus : "SIGNED",  
  date : "yyyy-MM-dd HH:mm:ss"  
}
```

En cas de refus de signature, les données transmises seront sous la forme JSON suivante :

```
{  
  type : "SIGNATURE",  
  contractId : "id",  
  contractorId : "ContractorID",  
  signatureStatus : "REFUSAL",  
  date : "yyyy-MM-dd HH:mm:ss"  
  refusJustification : "Message"  
}
```

# SUIVI DU CONTRAT

Dès lors que le contrat est validé, nous vous transmettrons les données au format JSON suivant :

```
{  
  type : "CONTRACT",  
  contractId : "id",  
  status : "ARCHIVED OR CLOSED",  
  signatureStatus : "REFUSAL",  
  date : "yyyy-MM-dd HH:mm:ss"  
  pdf : "base64 Doc"  
}
```

Les méthodes d'authentification suivantes sont disponibles pour ce callback :

- Basic Auth
- Token
- OAuth

# CALLBACK POST MULTIPART

# 05





# CALLBACK POST MULTIPART

Ce callback vous permet de recevoir et de suivre l'avancée des contrats durant tout le processus de signature.

Les données seront envoyées en utilisant le format multipart/form-data.

Nous attendons, pour mettre en place ce callback, une url **fonctionnelle et accessible publiquement**.

# SUIVI DES SIGNATURES

Le callback sera déclenché pour chaque signature, ou refus de signature, pour chaque signataire.

En cas de signature, les données transmises seront sous la forme des champs suivants :

```
{  
  type : "SIGNATURE",  
  contractId : "id",  
  contractorId : "ContractorID",  
  signatureStatus : "SIGNED",  
  date : "yyyy-MM-dd HH:mm:ss"  
}
```

En cas de refus de signature, les données transmises seront sous la forme JSON suivante :

```
{  
  type : "SIGNATURE",  
  contractId : "id",  
  contractorId : "ContractorID",  
  signatureStatus : "REFUSAL",  
  date : "yyyy-MM-dd HH:mm:ss"  
  refusJustification : "Message"  
}
```



# SUIVI DU CONTRAT

Dès lors que le contrat est validé, nous vous transmettrons un appel avec les champs suivants :

```
{  
  type : "CONTRACT",  
  contractId : "id",  
  status : "ARCHIVED OR CLOSED",  
  date : "yyyy-MM-dd HH:mm:ss"  
  pdf : file  
  option : "{ test : test }"  
  file[x] : AnnexeFile  
}
```

Les méthodes d'authentification suivantes sont disponibles pour ce callback :

- Basic Auth
- Token
- OAuth

# GESTION DES ERREURS

06



# GESTION DES ERREURS

En cas d'erreur, tous les types de callback effectueront 3 nouvelles tentatives à 5 minutes, 1h puis 24h.

Il peut arriver que l'appel arrive correctement à destination mais que le retour ne parvienne pas au service **oodrive\_sign**.

Dans ce cas, le service procédera à une nouvelle tentative à concurrence du nombre d'essais déjà réalisés. Il est donc **important de prendre en compte cette éventualité** dans vos développements.

# BESOIN D'AIDE? ON EST LÀ !

Parce qu'il n'y a pas d'heure pour apprendre, retrouvez dans l'espace **support**, toute la base de connaissances de l'application, des conseils, des tutos et toute la communauté d'utilisateurs pour partager.

Et si après cela vous avez encore une question ?

- ouvrez un ticket
- envoyez un mail à [delivery@calindasoftware.com](mailto:delivery@calindasoftware.com)
- appelez notre service Delivery au  
**09 80 090 203**





# ON S'APPELLE

## TEAM CUSTOMER SUCCESS

[customer@sellandsign.com](mailto:customer@sellandsign.com)

## TEAM SUPPORT

[support@sellandsign.com](mailto:support@sellandsign.com)

+33 (0)980 090 203

[support.sellandsign.com](https://support.sellandsign.com)