

# API CYNOD

Ce document a pour objectif de présenter les API exposées par l'application CYNOD pour la gestion des fonctionnalités internes mais aussi de celles pour la mise en place de services à valeur ajoutée de nos clients via les partenaires.

## Vue d'ensemble

<schéma de présentation des différents composants et de leurs interactions via API> **todo**

## Sécurité

La sécurité de l'interface est principalement basée sur l'authentification OAuth2 combinée au protocole https qui est le canal de communication par défaut des échanges. D'autres outils internes pourront être mis en place pour renforcer la sécurité tels que le filtrage d'adresse IP, les autorisations de ports et le VPN.

## Identification

Les informations suivantes sont mises à la disposition des partenaires autorisés à envoyer des requêtes:

1. **consumer-key**
2. **consumer-secret**

⚠ Ces informations permettront aux partenaires de pouvoir s'authentifier.

## Authentification

L'authentification est gérée via le protocole OAuth2.

Une fois connectés via l'API d'authentification, vous aurez un access\_token qui vous permettra d'accéder aux autres ressources. Les accès se feront en https.

⚠ Pour les partenaires qui le désirent, l'authentification peut être réalisée en *Basic Authentication* **que nous ne recommandons pas**.

## Principe général des échanges

Le format général des APIs respecte les principes ci-dessous :

- Les données échangées sont au format JSON.

- La gestion des erreurs se base sur les codes statuts http, et un code d'erreur interne selon les principe ci-dessous :

### **En cas de succès**

- http status : 200 - OK

- payload json de réponse

```
{
  "success": "true",
  "code": "success code",
  "message": "success message to the application user",
  "data": "response data if necessary"
}
```

### **En cas d'erreur fonctionnelle liée à la requête du client**

- http status : 400 - Bad Request (Client Error)

- payload json avec le détail de l'erreur

```
{
  "success": "false",
  "code": "error code message to the app user",
  "debugMessage": "verbose message for debug purpose",
  "moreInfo": "null"
}
```

### **En cas d'erreur d'authentification**

- http status : 401 - Unauthorized

### **En cas d'erreur système lié à un problème serveur**

- http status : 500 - Internal Server Error

- payload json avec le détail de l'erreur

```
{
  "success": "false"
  "code": "9999",
  "message": "message to the app user",
  "debugMessage": "verbose message for debug purpose"
}
```

⚠ La description des données échangées est au format *swagger*. Voir en annexe les différents codes d'erreur.

## Base Core API

### 3rd Party API

Cette section référence les API CYNOD qui sont mises à la disposition des partenaires pour construire des services digitaux au service de tous.

#### GET

##### API de consultation de solde carte

Type	GET
URI	/get-solde-carte
Description	envoi d'une requête pour la consultation du solde d'une carte

##### Paramètres :

Nom	Description
numeroCarte	Numéro de la carte du client
walletId	Numéro de téléphone du client
clientId	Identifiant du client

##### Header :

Content-type	application/json
Authorization	Bearer {{ACCESS_TOKEN}}

##### Request body :

Aucun

##### Réponses :

##### API de vérification de validité d'une carte

##### API de vérification de validité d'une carte en fonction d'un membre

##### API de récupération de la liste de carte d'un membre ou client

## POST

### API crédit carte

### API access token

From:

<https://cynod.wiki.sensoft-next.com/> - cynod

Permanent link:

<https://cynod.wiki.sensoft-next.com/doku.php?id=tecdoc:api&rev=1689348680>

Last update: **2023/07/14 15:31**

