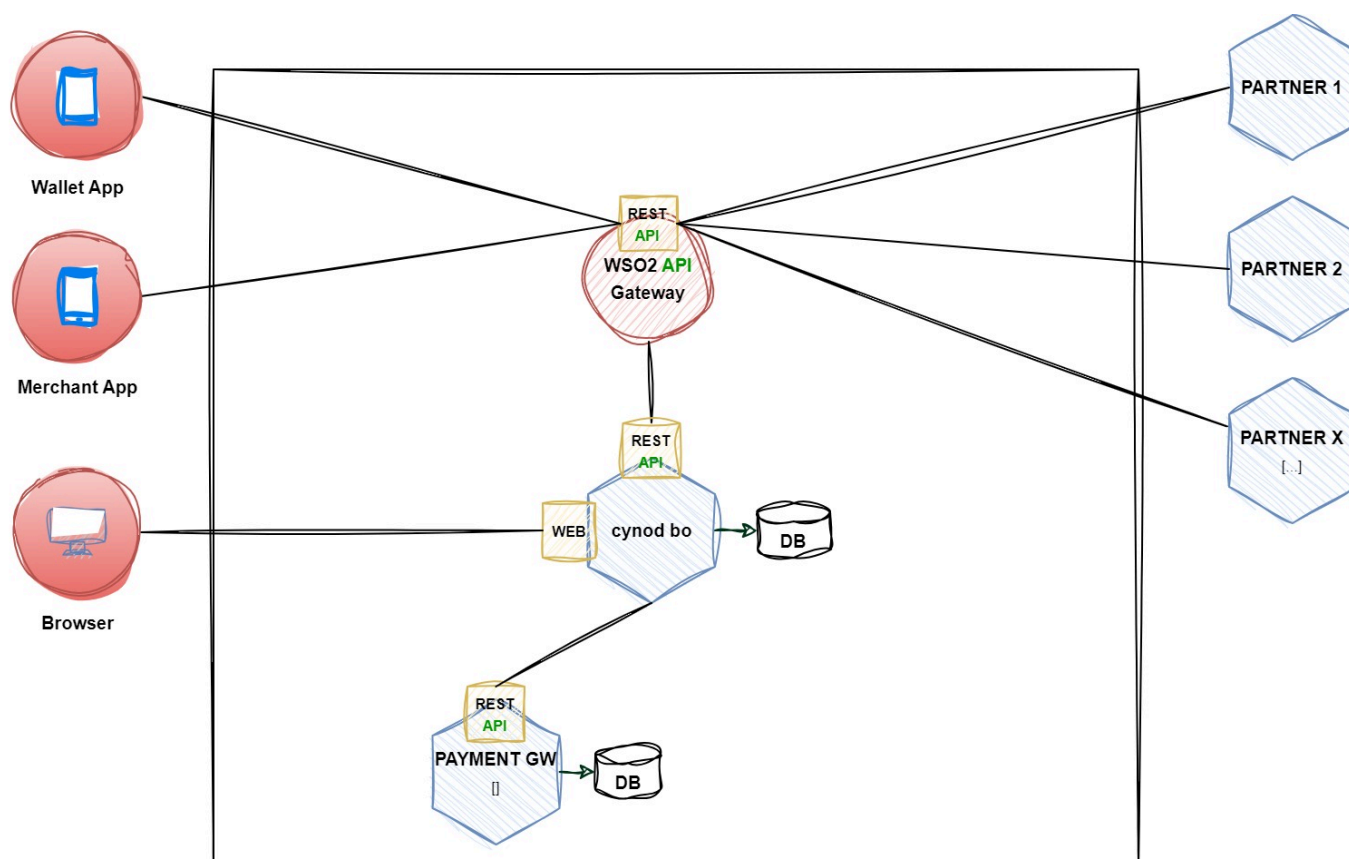


API CYNOD

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 ajouté de nos clients via les partenaires.

Vue d'ensemble



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 principes 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.

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 :

Code http	200
Description	Success

Exemple modèle payload json

```
{
  "success": true,
  "code": 200,
  "message": "Sauf omission* de notre part votre nouveau solde est de 142000 FCFA ",
  "data": [
    {
      "numeroCarte": "7019800100009569",
      "soldeOnline": 142000.0,
      "dateSolde": "23-08-2021 10:11:42"
    }
  ]
}
```

Code http	400
Description	Bad request

Exemple modèle payload json

```
{
  "success": false,
  "code": "412",
  "message": "Utilisateur inexistant ou invalide",
  "debugMessage": "Utilisateur inexistant ou invalide",
  "moreInfo": null
}
```

API de vérification de validité d'une carte

Type	GET
URI	/check-card-validity
Description	envoi d'une requête pour la validité 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 :

Code http	200
Description	Success

Exemple modèle payload json

```
{
  "success": true,
  "code": 200,
  "message": "Votre carte est valide et valable jusqu'au 18/07/2029",
  "data": [
    {
      "numeroCarte": "7019800100009569",
      "finValidite": "18/07/2029",
      "nom": "xxx",
      "prenom": "xxx",
      "email": "",
      "telephone": "",
      "numeroPiece": "",
      "dateNaissance": ""
    }
  ]
}
```

Code http	400
Description	Bad request

Exemple modèle payload json

```
{
  "success": false,
  "code": "405",
  "message": "Paramètere numeroCarte manquant ou invalide",
  "debugMessage": null,
  "moreInfo": null
}
```

```
}
```

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

API crédit carte (recharge)

API dédit carte (paiement)

API access token

From:

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

Permanent link:

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

Last update: **2023/07/26 08:39**

