

Specification SI Crous Encaissement Restauration - Logiciel Externe

Sommaire

- 1 Objectifs
- 2 Ressources disponibles
 - 2.1 Portail des API Crous
 - 2.2 IzlyPay
 - 2.3 Lecture de la carte Izly
 - 2.4 Utilisation du QRCode
 - 2.5 Numérotation des points de ventes
 - 2.6 Règles métiers dans l'utilisation des API Crous
- 3 Quelques données générales métiers 'Crous'
 - 3.1 Sociétés et tarifs

Objectifs

Les objectifs de cette documentation sont de permettre et de faciliter

- l'intégration du paiement Izly dans des applications informatiques, à partir
 - de la lecture d'une carte Izly et via un lecteur PC/SC
 - de la lecture d'un QRCode de paiement Izly, généré à partir de l'application mobile Izly
 - de la saisie du login/mot de passe des ayants droits Izly sur un site de commerce en ligne
- l'interfaçage, via des API, entre le SI des Crous et un SI externe utilisant le paiement Izly

Ressources disponibles

Portail des API Crous

Le site <https://dev.lescrous.fr/> est le portail des API Crous. Il décrit les ressources pour

- Récupérer les données du SI des Crous

- Renvoyer vers les SI des Crous l'intégralité de l'équivalent des tickets de caisses des ventes
- Alimenter l'application Whiz
 - Whiz, est une application dédiée à l'affichage dynamique et à la gestion des files d'attente. Whiz est utilisée dans les cuisines des restaurants universitaires pour aider à l'ordonnancement des commandes à préparer.
- Utiliser l'application Izly de paiement
 - accessible aussi par <https://periph-pp.izly.fr/rest/swagger/ui/index>
- ...

Pour faire une demande d'accès au portail API des Crous

- Si vous ne possédez pas de compte schibboleth (en général le cas des sociétés privées) , veuillez utiliser le bouton 'créez un compte' en haut à droite de l'écran
- Si vous possédez un compte schibboleth, merci de l'utiliser

Dans les deux cas de figure, après votre demande de création de compte, l'équipe VEM devra créer votre compte et vous attribuer vos droits d'utilisation

IzlyPay

Les ressources suivantes décrivent l'utilisation du moyen de paiement Izly, à partir d'un site web via le login/mot de passe de l'ayant droit Izly :

- Izly_-_Interface_-_IzlyPay
- [1] avec la possibilité de demander au consortium ESUP-SGC un exemple de code mettant en œuvre le paiement Izly

Lecture de la carte Izly

La lecture des cartes Izly est nécessaire pour déterminer le profil tarifaire des ayants droits. Il existe trois solutions pour lire une carte Izly.

A) Utiliser une interface déjà existante.

Se procurer la liste des sociétés ayant réalisées l'interface Izly

B) Développer l'interface sur un lecteur Tetra Ingenico

Le Cnous a intégré dans un lecteur Ingenico Tetra (Desk 5000 ou move 5000), une application qui permet la lecture des cartes Izly. Cette application permet uniquement de lire le mapping de la carte, principalement le numéro de carte izly. L'api Izly, appelée à partir du logiciel de caisse, permettra ensuite de connaître le profil tarifaire de l'ayant droit et d'effectuer le paiement. Les différentes documentations sont disponibles

- Doc configuration liaison PC/Tetra, 'InstallationTetra.pdf'
- Doc description interface PCL & messages application izly 'Interface Tetra.pdf'
 - Seule la requête 'DemandeZDC' est à utiliser. Cette requête permet de lire le mapping des cartes Izly et vous permet de récupérer les paramètres, SAM, Application Number et idSupport. Ces données étant indispensables pour l'opération de paiement Izly décrite dans le chapitre ci-dessous
- Installer le Bridge PCL sur un tetra 'copierBridgePCLsurTetra.pdf'
- Mise à jour Tetra par clé USB 'miseAJourApplicationViaUSB.pdf'
- PROJET - Application IZLY - TPE Ingenico TETRA autonomes.pdf

C) Développer l'interface sur un lecteur PC/SC

Dans le cas de l'utilisation d'un lecteur PC/SC, Prox'n'roll Identive Omnikey, il est nécessaire de développer des programmes de lecture de la carte Izly pour y trouver la zone mémoire dédiée au fonctionnement d'izly (ZDC : Zone de données Crous) Différents documents sont à demander au cnous pour permettre cette lecture

- CNOUS_CARTE-SPEC_DLL-GA.pdf, pour la lecture des cartes NXP
 - Dans cette documentation les fonctions à utiliser, sont les suivantes
 - activate
 - initialise
 - liste_lecteurs
 - session
 - liberer_liste_lecteurs
 -
 - ainsi que la lecture attentive du chapitre 6 pour la détail du calcul du cryptogramme d'authentification
- Spécification Lecture Carte Calypso sans SAM V1.pdf, pour la lecture des cartes Calypso

Un SAM, Secure Access Module, est indispensable pour effectuer une lecture d'une carte Izly, ainsi que les dll Cnous_carte_lecture.dll , pcsc_desfire.dll

Mise à disposition des spécifications techniques

Sur demande le CNOUS fournira :

- Un SAM Desfire AV2 permettant la lecture de la ZDC
- Les derniers mises à jour des documents nécessaires à la lecture de la ZDC

La ZDC (Zone de Données Cnous)

Il s'agit d'une zone mémoire présente sur les différentes technologies de cartes multiservices utilisées par les étudiants. La ZDC contient un identifiant unique, spécifique au fonctionnement des œuvres Universitaires, qui est le point d'entrée pour les appels des webservices de lecture et de paiement. La lecture de la ZDC impose l'utilisation d'un SAM de lecture. La méthode de lecture de la ZDC varie suivant les différentes technologies de cartes utilisées

SAM de lecture

Izly fait appel à des cartes à puces de sécurité « SAM » pour sécuriser la carte multiservice Tout équipement (caisse, automate, ...) devant lire une carte multiservice devra disposer d'un SAM de lecture (ou pouvoir y accéder par un canal de communication fiable).

Les types de cartes à prendre en compte sont les suivants

- Mifare Desfire EV1 et EV2
- Mifare Ultralight C
- Mifare Classic
 - Avec des spécifications différentes pour les cartes 1 K et 4 K
 - Calypso

Utilisation du QRCode

Le paiement Izly peut se déclencher à partir de la lecture d'un QRCode

Génération de QRCode de test Il faut demander au Cnous un compte Izly de test. A partir de là deux solutions existent :

- Se connecter à son espace personnel Izly de test <https://mon-espace-pp.izly.fr> et demander la génération de QRCode de paiement. Ces QRCode peuvent être imprimés ou scannés à l'écran.
- Télécharger l'application mobile Izly de Pré prod et générer via le smartphone des QRcode de paiement (actuellement cette solution n'est plus fonctionnelle)

Voir dans l'application developers.lescrous les ressources suivantes de l'API paiements Izly

- Récupération d'information sur l'ayant droit Get `/api/rightolders/searchQRCode`
- Effectuer un débit Post `/api/payments/debit/{type}/{code} ...` avec type = qrcode

Le QR Code doit être encrypté en Base 64 dans les API

Numérotation des points de ventes

La numérotation des points de ventes doit suivre la codification utilisée dans les Crous, de type idCrous, idUg, idRu, idUd, idCaisse

- idCrous : numéro du Crous
- idUg : numéro de l'unité de gestion
- idRu : numéro du restaurant Universitaire
- idUd : numéro de l'unité de distribution
- idCaisse : numéro de caisse

Règles métiers dans l'utilisation des API Crous

Ce chapitre concerne uniquement le cas d'un logiciel, caisse ou commandes en lignes, **intégré au SI des Crous**, c'est à dire utilisé par les Crous. Dans ce cas le logiciel doit utiliser les données métiers des Crous (code et libelle article, prix) et retourner vers le SI des Crous, l'équivalent d'un ticket de caisse.

La connexion au portail API des Crous, cf ci-dessus, décrit l'ensemble des ressources permettant les échanges avec le SI des Crous

Les ressources à utiliser de façon obligatoires sont

- authenticate
 - Pour récupérer le token à partir des paramètres d'authentification générés dans VEM par le Crous
 - Voir 'générer une nouvelle clé' dans la fiche caisse de VEM
 - A chaque point de vente, crous ug ru ud caisse, doit correspondre une valeur de clé d'installation différente. Cette clé est générée dans VEM.
Un contrôle d'utilisation de l'API refusera les envois avec une clé d'identification ne correspondant pas à l'identification crous-ug-ru-ud-caisse
- caisse
 - Pour récupérer une clé d'authentification Izly
- tva
 - Pour connaître la table de correspondance entre code et taux de tva
- articles
 - Préciser la valeur versBornesPrepaiement = 1 pour récupérer uniquement les articles dédiés aux fonctionnement des bornes
- tickets
 - Pour renvoyer vers VEM le ticket de caisse

- **champs obligatoires pour la ressource tickets**

- idTicketEntete
 - Un numéro unique et séquentiel par point de vente
 - numeroTicket (imprimé sur le ticket de caisse)
 - le numéro de ticket doit être unique par point de vente. Un exemple de numérotation peut être **20031810** pour la 10ème commande du 18/03/2020 sur ce point de vente
 - versionlogiciel , pour aide au débogage
 - date
 - codeEvenement = 'VT'
-
- listeLignes (au moins un élément)
 - numeroLigne (autoincrément par exemple, en partant de 1 à chaque ticket de caisse . Chaque numéro de ligne est unique pour un ticket)
 - codeArticle
 - quantite
 - codeTva
 - prixUnitaireHt
 - prixUnitaireTtc
 - totalHt
 - totalTtc
-
- listeReglements (un seul élément)
 - codeReglement (4:CB, 9:Izly, 12:fidélité)
 - montant (ttc)
 - idSupport ou qrCode si codeReglement = 9
 - idTransactionIzly si codeReglement = 9
 - numeroApplicatif si idSupport renseigné
-
- complementaire (un seul élément)
 - codeClient si client izly
 - peut ne pas être obligatoire si inconnu (paiement par carte izly, dans ce cas là idSupport est obligatoire)
 - soldeClient si client izly (solde après transaction en € et centimes d'€ : ex 15.25 €)
 - cumulPointsFidelite si client izly (non obligatoire, sauf si utilisé)
 - soldePointsFidelite si client izly (non obligatoire, sauf si utilisé)
 - nomClient si client izly (non obligatoire)

- dateFinValiditeClient": "2018-07-06T09:30:41+02:00", si client izly (non obligatoire)
 - codeSocieteClient code société utilisé pour déterminer les prix de vente
 - tarifClient code tarif utilisé pour déterminer les prix de vente
 - codeCrousClient si client izly, codeCrous du client Izly (non obligatoire)
 - rneEtablissement si client izly, non obligatoire
 - affectation si client izly, non obligatoire
 - adresseMail si client izly , non obligatoire
 - idSupport si client izly
 - obligatoire si paiement à partir de la carte Izly
 - idTypeDistribution: 1 : sur place, 2 : à emporter, 3 : livré (non mis en place : à ne pas renseigner)
 - idLogiciel
- cohérences à effectuer
 - pour chaque ligne code article soit exister dans la ressource article
 - pour chaque ligne : $\text{total} = \text{PU} * \text{quantité}$
 - pour chaque ligne : calcul de la TVA
 - pour le ticket : $\text{total listeLignes} = \text{total listeReglements}$
 - pour le ticket : $\text{total des gainfidleite} = \text{cumulpointsfidelite}$

Dans le cas de l'interface d'un logiciel de caisse, les ressources à utiliser suivant les demandes du Crous sont

- societes
- societes-tarifs
 - Afin de récupérer le taux de tva des clients de ce societe/tarif, et l'appliquer dans le cas du paiement par CB
- periodes
- societes-tarifs-periode
- articles-societes-tarifs-periodes
 - Pour récupérer les prix des articles des sociétés 10 et 8000. 10 pour les étudiants et 8000 pour les non étudiants avec l'application de la tva pour ces derniers en fonction de leur societe/tarif
 - Préciser la valeur versBornesPrepaiement = 1 pour récupérer uniquement les articles dédiés aux fonctionnement des bornes
 - Préciser la valeur 0 dans le paramètre periodesFutures pour ne récupérer que les prix actuels
- droits-entree
- subventions
- droits-entree-types

« Suivant demande du Crous » : Les prix associés aux articles peuvent venir du SI Crous ou être saisi dans le logiciel fourni. Si les prix viennent du SI Crous les mentions « suivant demande du Crous » deviennent obligatoires.

Quelques données générales métiers 'Crous'

Sociétés et tarifs

Les articles sont vendus à des prix et tva différents suivants le profil tarifaire des clients
Les profils tarifaires des clients sont répartis en société et tarif , une société pouvant contenir plusieurs tarifs

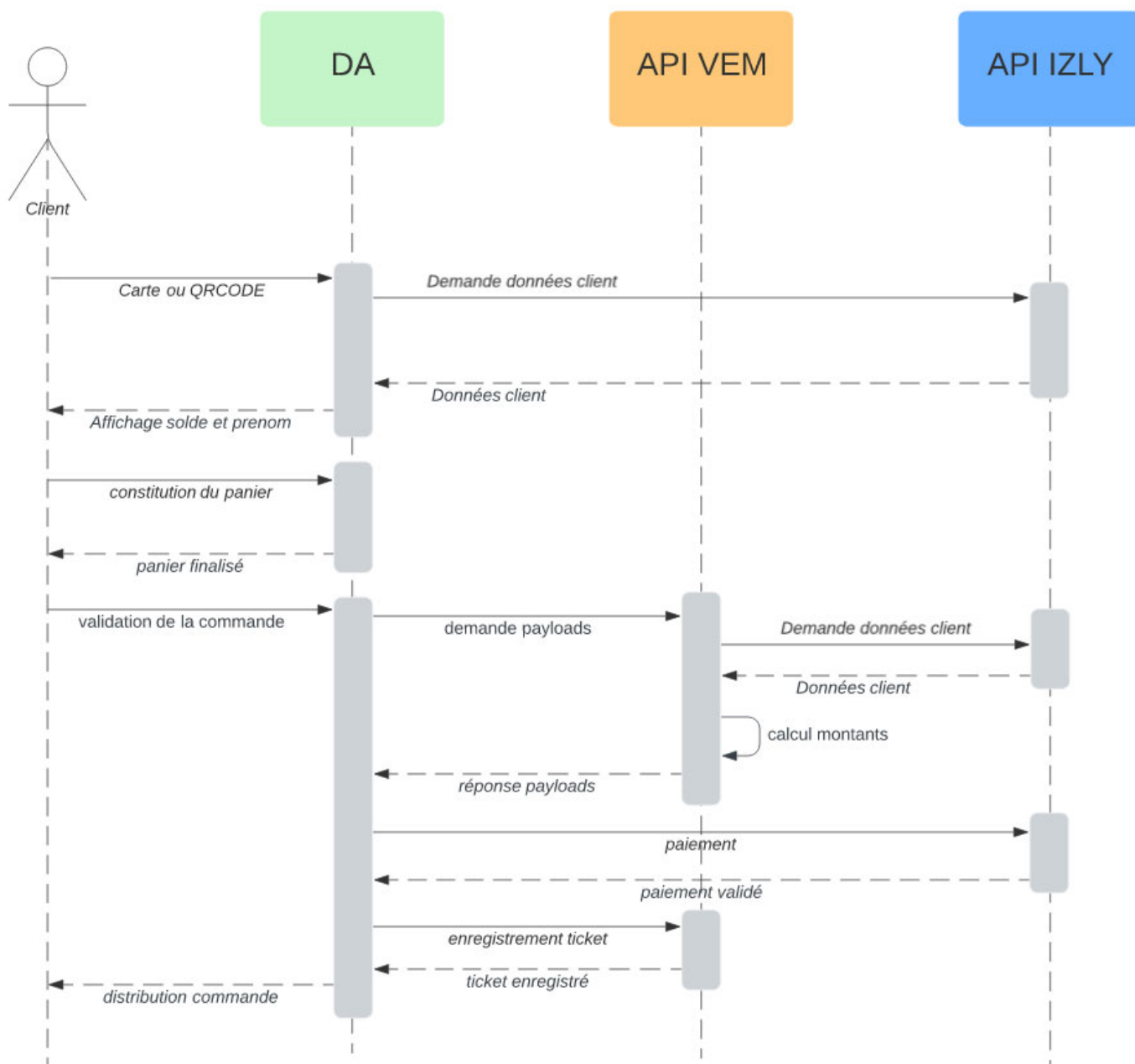
Quelque soit le Crous

- société 10 tarif 1 représente le profil tarifaire associé aux étudiants
- société 8000 tarif 1 représente le profil tarifaire associé aux hôtes de passages
- société 8024 tarif 1 représente, si il est utilisé dans le Crous, le profil tarifaire associé aux étudiants pour l'application Crous de commande en ligne
- société 8024 tarif 2 représente, si il est utilisé dans le Crous, le profil tarifaire associé aux non étudiants pour l'application Crous de commande en ligne

Récupérée de « https://wiki.doc.lescrous.fr/index.php?title=Specification_SI_Crous_Encaissement_Restauration_-_Logiciel_Externe&oldid=5836 »

Specification SI Crous Encaissement Restauration - NDV

Ce cas d'utilisation nécessite la mise en place d'une ressource de l'API VEM qui se charge de calculer le montant de l'article en fonction des règles métiers et des données instantanées des CROUS et qui retourne les payloads à envoyer dans les requêtes aux APIs izly et recap.



Sommaire

- 1 Lecture de la carte ou du Qrcode de l'ayant droit
- 2 Requête vers l'API Izly pour récupérer les données de l'Ayant droit
- 3 Requête vers l'API VEM
- 4 Requête vers l'API izly pour procéder au paiement
- 5 Requête vers recap pour enregistrer le ticket

Lecture de la carte ou du Qrcode de l'ayant droit

Requête vers l'API Izly pour récupérer les données de l'Ayant droit

Requête sur `/api/cards/{idSupport}` (si carte) ou `/api/rightolders/searchQRCode` (qrcode)

Exemple <https://periph-pp.izly.fr/rest/api/cards/36000134>

réponse :

```
{
  "dateOpposition": "",
  "typeOpposition": "",
  "pixSS": "03",
  "pixNN": "0000",
  "appl": "D1",
  "numeroProtocolaire": "041516928E1194",
  "numeroApplicatif": "041516928E1194",
  "identifiantEmetteur": 80,
  "versionMapping": "1",
  "dateCreation": "2023-06-21T16:17:43+02:00",
  "Compteur": [
    {
      "idCompteur": 1,
      "valeur": 0,
      "Date": null
    },
    {
      "idCompteur": 2,
      "valeur": 0,
      "Date": null
    },
    {
      "idCompteur": 3,
      "valeur": 0,
      "Date": null
    },
    {
      "idCompteur": 4,
      "valeur": 0,
      "Date": null
    }
  ],
  "codeClient": "test.pcad4@yopmail.com",
  "numeroAffectation": 3,
  "solde": 0,
  "codeSociete": 10,
  "numTarif": 90,
  "nom": "VIBRAC",
  "prenom": "Victor",
  "dateDernierServiceMidi": "",
  "dateDernierServiceSoir": "",
  "dateFinValidite": "2019-11-30T00:00:00+01:00",
  "cycle": 1,
  "courriel": "test.pcad4@yopmail.com",
  "rneEtablissement": "0330151K",
  "rneComposante": "0330151K",
  "codeEtudiant": "ee",
  "dateNaissance": "1999-09-09T00:00:00+02:00",
  "ine": "050QQQAAZZZ",
  "telPortable": "33000008464",
  "typeClient": "PME",
  "idIzly": 6000016714,
  "identifiantCrous": 5,
  "statutCompteClient": 2,
  "statutBlocage": 1
}
```

Requête vers l'API VEM

Il faut envoyer le contenu le codeClient récupéré dans la réponse précédente ainsi que le qrcode ou le numero de zdc de la carte + le contenu du panier articles avec le tarif "standard" de chaque article.

Dans l'exemple ci dessous, l'article avec le code 1 est une entrée et le code 2 est un plat. Cette ensemble est une "formule" éligible au repas à 1€ ou 3,3€ selon le statut de l'étudiant.

url : /v1/{numeroCrous}/montant-ticket

```
{
  "versionLogiciel": "1.2.3",
  "panier": [{
    "codeArticle": 1,
    "quantite": 1,
  }, {
    "codeArticle": 2,
    "quantite": 1
  }],
  "client": {
    "codeClient": "test.pcad4@yopmail.com",
    "code": "56356"
  }
}
```

L'API VEM va réaliser une requête vers izly afin d'avoir une version récente des données et notamment des compteurs puis calculer le montant en fonction des règles métier du crous.

Dans l'exemple ci dessous, le ticket recapï contient 3 articles. Les 2 articles du panier ont le montant a 0 car c'est l'article "menu" (1500) qui contient le prix.

réponse :

```
{
  "izly":{
    "url":"https://periph.izly.fr/api/payments/debit/qrcode",
    "headers": {
      "Application-Number": "04281ADA404D80",
      "Software": "50",
      "SAM": "{SAM}",
      "TransactionDate": "{DATE_HEURE_VENTE}"
    },
    "payload": {
      "lCompteur": [
        {
          "idCompteur": 4,
          "valeur": "2023-06-21T00:00:00"
        }
      ],
      "numeroTicketCaisse": {NUMERO_TICKET},
      "montantADebiter": 100,
      "code": "12345647",
      "dateServiceMidi": "2023-06-21T00:00:00",
      "dateServiceSoir": null
    },
  },
  "recapi":{
    "url":"https://api.lescrous.fr/recapi/v1/tickets",
    "payload": {
      "idTicketEntete": "{NUMERO_TICKET}",
      "numeroTicket": "{NUMERO_TICKET}",
      "versionLogiciel": "1.2.3",
      "date": "{DATE_HEURE_VENTE}",
      "codeEvenement": "VT",
      "listeLignes": [{
        "numeroLigne": 1,
        "codeArticle": 1,
        "quantite": 1,
        "codeTva": 0,
        "prixUnitaireHt": 0,
        "prixUnitaireTtc": 0,
        "prixUnitaireTtcOrigine": 2
        "totalHt": 0,
        "totalTtc": 0,
        "codeArticleMenu": 1500
      },{
        "numeroLigne": 2,
        "codeArticle": 2,
        "quantite": 1,
        "codeTva": 0,
        "prixUnitaireHt": 0,
        "prixUnitaireTtc": 0,
        "prixUnitaireTtcOrigine": 5
        "totalHt": 0,
        "totalTtc": 0,
        "codeArticleMenu": 1500
      },{
        "numeroLigne": 3,
        "codeArticle": 1500,
        "quantite": 1,
        "codeTva": 0,
        "prixUnitaireHt": 1,
        "prixUnitaireTtc": 1,
        "totalHt": 1,
        "totalTtc": 1,
        "articleTypeRepas": true,
        "codeArticleMenu": 1500
      }],
      "listeReglements": [{
        "codeReglement": 9,
        "montant": 1,
        "montantOrigine" : 7,
        "idTransactionIzly": {ID_TRANSACTION_IZLY},
        "idSupport":"36000134",
        "numeroApplicatif":"041516928E1194",
        "qrCode":null,
        "codeClient":0
        "affectation":0
      }],
      "complementaire": {
        "idTypeDistribution": 2,
        "idLogiciel": 1
      }
    }
  }
}
```

Requête vers l'API izly pour procéder au paiement

La requête doit contenir le payload indiqué dans l'element **izly** de la réponse précédente

Certaines données sont à personnaliser, elle sont retournées sous de libellés en majuscules entourés d'accolades (ex {NUMERO_TICKET})

Requête vers recap pour enregistrer le ticket

Seulement si le paiement izly a bien eu lieu.

La requête doit contenir le payload indiqué dans l'élément **recapi** de la réponse précédente

Certaines données sont à personnaliser, elle sont retournées sous de libellés en majuscules entourés d'accolades (ex {NUMERO_TICKET})

Récupérée de « https://wiki.doc.lescrous.fr/index.php?title=Specification_SI_Crous_Encaissement_Restauration_-_NDV&oldid=4771 »

Documentation IzlyPartner

I. Les étapes à suivre pour utiliser IzlyConnect

1- Enregistrement du site web partenaire sur Izly

a. Création de point d'encaissement de type « Izly Partner »

Pour utiliser le service Izly Partner, le site web partenaire doit être enregistré sur IZLY, en tant que point d'encaissement de type « Izly Partner ».

Pour ce faire, un administrateur national ou régional d'IZLY doit créer le point d'encaissement dans le Back-Office IZLY, en utilisant le type « Izly Partner » comme type de point d'encaissement.

En choisissant ce type, six champs obligatoires vont apparaître comme indiqué ci-dessous :

Type de point d'encaissement *	Izly Partner
Controle SAM :	<input type="checkbox"/>
1 Izly Pay :	<input type="checkbox"/>
2 Nom du site en ligne *	<input type="text"/>
3 URL de callback *	<input type="text"/>
Lien de redirection de l'utilisateur après connexion*	<input type="text"/>
4	
Lien de redirection de l'utilisateur après paiement*	<input type="text"/>
5	
Lien vers la page d'accueil du site partenaire	<input type="text"/>
6 *	

1 Autoriser le paiement sur le point d'encaissement via le service proposé par IZLY.

2 Le nom du site en ligne, visible par l'utilisateur du site, au moment de l'autorisation du paiement sur Izly.

3 URL de callback, notification serveur à serveur du succès de la transaction.

4 URL vers laquelle Izly va rediriger l'utilisateur après connexion.

5

URL vers laquelle Izly va rediriger l'utilisateur après le paiement.

6

URL vers laquelle Izly va rediriger l'utilisateur en cas d'annulation de connexion ou de paiement.

b. [Autorisation de flux sortant de la part de l'hébergeur du site Izly](#)

Par mesure de sécurité, les flux sortants du serveur Izly sont contrôlés par l'hébergeur.

De ce fait, le callback depuis le **serveur Izly** vers le **serveur du site web partenaire** ne pourra s'effectuer que lorsque l'URL du callback sera autorisée par l'hébergeur.

Ce traitement nécessite des actions manuelles de la part de Xpollens pour autoriser l'accès au site définit lors de la création du point d'encaissement.

c. [Agrément du point d'encaissement](#)

Comme tous les autres types de point d'encaissement, le point d'encaissement de type Izly Partner doit être agréé avant d'être utilisable.

d. [Activation IzlyPay du point d'encaissement Izly Partner](#)

Afin de permettre au point d'encaissement de type Izly Partner de réaliser des paiements, la coche IzlyPay doit être cochée.

Si celle-ci n'est pas cochée, seule la fonctionnalité de connexion est utilisable.

e. [Récupération des paramètres d'authentification du site web partenaire](#)

Pour utiliser les services Izly Partner, des paramètres d'authentification sont requis pour appeler les web services.

Ces paramètres sont récupérables depuis la page d'édition du point d'encaissement, comme indiqué ci-dessous :

1

Identifiant de l'application client:

Site-en-ligne-16-16-16-13

2

Clé d'API :

U2l0ZS1lbi1saWduZS0xNi0xNi0xNi0xMz

1

Identifiant du site web partenaire sur Izly.

2

Clé d'API du site web partenaire.

Pour utiliser le service IzlyConnect, on récupère les informations d'un ayant droit en se basant sur son authentification et sur le point d'encaissement utilisé de type izlyPartner enregistré sur IZLY.

2- Authentification de l'Ad et récupération du code généré sur Izly

Syntaxe de l'Url utilisé pour s'authentifier :


https://rest-pp.izly.fr/oauth/Home/Authorize?response_type=code&client_id={app_identifiant}&scope=1

Remplacer l'app_identifiant sur le site par l'Identifiant de l'application client du point d'encaissement qui se trouve dans le BO.

Exemple :

https://rest-pp.izly.fr/oauth/Home/Authorize?response_type=code&client_id=Aix%20Marseille-777-777-777-777&scope=16385

Ci-dessous l'interface d'authentification :



Connectez-vous à votre compte Izly

Pour pouvoir régler votre achat avec votre compte Izly, merci d'indiquer votre identifiant et votre mot de passe Izly.

1 Identifiant :

2 Mot de passe :

[Mot de passe oublié ?](#)

1 L'adresse email ou le numéro de téléphone de l'Ayant droit.

2 Mot de passe.

Après la validation du login et du mot de passe de l'Ayant droit, l'utilisateur est redirigé automatiquement vers le site de connexion avec un code généré.

Exemple :

<https://mon-espace-pp.izly.fr/IzlyConnect/loginCallback?code=MTs1NjUeTI5R3lrNjRPdkFtRzJvNEZLQzIBbGdxYkhiUjQ%3d>

3- Génération du token :

Syntaxe CURL utilisé sur POSTMAN :

```
curl -X POST \ 'https://rest-pp.izly.fr/oauth/Home/Token?grant_type=authorization_code&code={code}' \
-H 'authorization: Basic {authorization}'
```

Ci-dessous un exemple de génération de token réalisé sur POSTMAN :

1

POST

http://rest-pp.izly.fr/oauth/Home/Token?grant_type=authorization_code&code=MTsyODs0eE5yZ3E...

2

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

	KEY	VALUE	DESCRIPTION
3	<input checked="" type="checkbox"/> grant_type	authorization_code	
	<input checked="" type="checkbox"/> code	MTsyODs0eE5yZ3BYMjBtVI9PckpoNWJoYl9YfjB...	
	Key	Value	Description

POST

http://rest-pp.izly.fr/oauth/Home/Token?grant_type=authorization_code&code=MTsyODs0eE5yZ3E...

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Headers

7 hidden

	KEY	VALUE	DESCRIPTION
4	<input checked="" type="checkbox"/> authorization	Basic QWl4IE1hcnNlaWxsZS03NzctNzc3LTc3Ny...	
	Key	Value	Description

Pretty

Raw

Preview

Visualize

JSON

5

```

1  {
2    "access_token": "MTsyODtQUURPVk9kQnFaRmhGZGZST212cTNBNdhYaHlaNi4=",
3    "token_type": "Bearer",
4    "expires_in": 1797,
5    "refresh_token": "MTsyODtuOV8yWEY4f1Y3RTR5SVdlUmlwZlVjc05IT1lIMzE=",
6    "scope": "users_details payments izly_pay"
7  }

```

- 1 Le verbe POST
- 2 L'Url
- 3 Paramètres (grant_type est toujours authorization_code alors le code on l'a récupéré du site d'authentification)
- 4 Entêtes (Basic + la clé API récupéré depuis le BO)
- 5 La réponse (Le token généré)

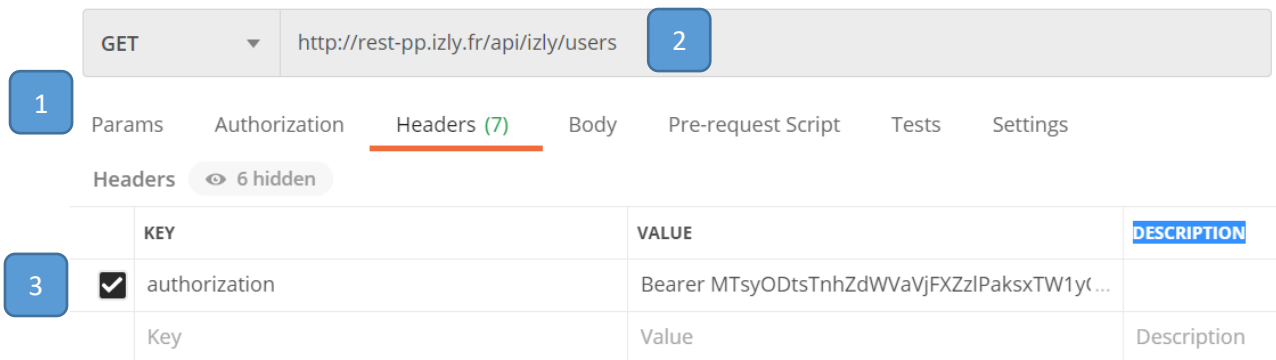
4- Utiliser le access_token pour récupérer le détail de l'AD sur POSTMAN :

Syntaxe CURL utilisé en PreProd sur POSTMAN :

```
curl -X GET \
```

```
https://rest-pp.izly.fr/api/izly/users \
-H 'authorization: Bearer MTs1NjtjR2I0anpsNncxRnNmewFXcnUwXy5BOH5KMXhWT0g= '
```

Ci-dessous un exemple de récupération des informations d'un AD en utilisant le token récupérer par avant sur POSTMAN :



1 Le Verbe GET

2 L'url utilisé.

3 Entête (Bearer + access_token)

Résultat obtenu (Les informations retournés) :

```
"RightHolderId": 1255,
"IdIzly": 6000001255,
"Societe": 1,
"Tarif": 5,
"AnneeUniv": "",
"Employeur": "",
"RneEtablissement": "1",
"RneComposante": "1",
"CodeEtu": "lolotroll1",
"Ine": "123457",
"CodeClient": "00001035",
"TypeClient": "PME",
"NumeroAffectation": 1,
"Id": 1255,
```

5- Utiliser le refresh_token en cas d'expiration du token :

Syntaxe CURL utilisé en PreProd sur POSTMAN :

```
curl -X POST \
```

```
'https://rest-pp.izly.fr/oauth/Home/Token?grant_type=refresh_token&refresh_token=MTs1Njs5TGZ
```

```
3TlRPanJJQ1BpeEk4eEpNLnc4eEh4YkxjdXY%3D' \ -H 'authorization: Basic  
aXpseS1jb25uZWNOXRLc3QxO1JQUEVRc3FPV1JpU0drb0NnRUZP'
```

Ci-dessous un exemple de rafraichir le token en cas d'expiration sur POSTMAN :

1

POST

http://rest-pp.izly.fr/oauth/Home/Token?grant_type=refresh_token&refresh_token=MTsyODtWMm...

2

Params ● Authorization Headers (8) Body Pre-request Script Tests Settings

Headers 👁 7 hidden

	KEY	VALUE	DESCRIPTION
3	<input checked="" type="checkbox"/> authorization	Basic QWl4IE1hcnNlaWxsZS03NzctNzc3LTc3Ny03	
	Key	Value	Description

POST

http://rest-pp.izly.fr/oauth/Home/Token?grant_type=refresh_token&refresh_token=MTsyODtWMm...

Params ● Authorization Headers (8) Body Pre-request Script Tests Settings

Query Params

	KEY	VALUE	DESCRIPTION
4	<input checked="" type="checkbox"/> grant_type	refresh_token	
	<input checked="" type="checkbox"/> refresh_token	MTsyODtWMmtXT3VWYUVPTElxSmZDaGpRQ2d1Y	
	Key	Value	Description

1 Le verbe POST

2 L'url utilisé.

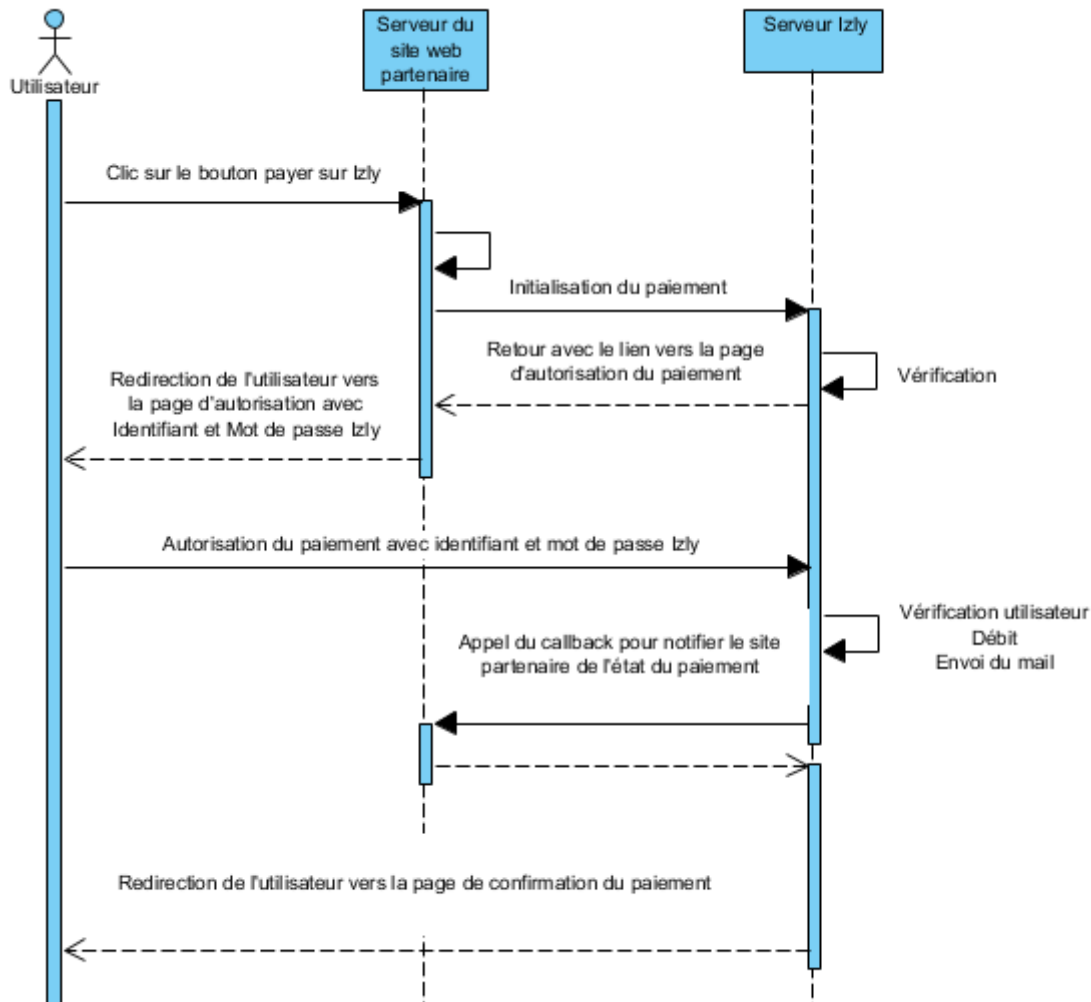
3 Entête (Basic + clé API sur la BO)

4 Le refresh_token qu'on a eu dans la première génération du token.

II. Les étapes à suivre pour utiliser IzlyPay

1- Paiement avec Izly

Le paiement avec Izly s'effectue en quelques étapes comme schématisées ci-dessous :



a. Initialisation du paiement

La première étape pour payer avec Izly est **l'initialisation du paiement** qui sera effectuée par la suite d'un appel du web service décrit ci-dessous :

Environnement	PREPROD
Resource URL	https://rest-pp.izly.fr/api/Izly/WebPayments
Method	POST

Environnement	PROD
Resource URL	https://rest.izly.fr/api/Izly/WebPayments
Method	POST

Avec les paramètres ci-dessous :

- Header

Clé	Valeur	Description
Authorization	Basic \$apiKey	\$apiKey: Clé d'API du partenaire La valeur est renseignée dans le back-office Izly, dans la page d'édition du point d'encaissement de type Izly Partner, sous le champ « Clé d'API » comme montré dans la capture d'écran I. 1- d.
Content-type	application/vnd.Xpollens.v1+json	
AppIdentifiant	\$AppIdentifiant	\$AppIdentifiant = Identifiant du site partenaire chez Izly. La valeur est renseignée dans le BO, dans la page d'édition du point d'encaissement de type Izly Partner, sous le champ « Identifiant de l'application client », comme montré dans la capture d'écran I. 1- d.

- Body

Le Body sera un objet JSON avec les propriétés suivantes

Propriété	Type	Obligatoire	Description
Amount	Int	Obligatoire	Montant du paiement TTC ⇒ Valeur en centimes
ClientOrderId	String	Obligatoire	Référence du paiement coté partenaire
CallbackParameters	String	Facultatif	Liste de paramètres avec les valeurs que Xpollens va concaténer à l'url de callback (paramétrée dans le BO) et appeler après le paiement.
RedirectUrlAfterPayment	String	Facultatif	Url que Xpollens va utiliser pour rediriger l'AD vers le site partenaire après le paiement
Message	String	Facultatif	Commentaire qui peut être indiqué de la part du partenaire
Software	Int	Facultatif	Id du logiciel du partenaire

- Response

La réponse sera un objet JSON avec les propriétés suivantes

Propriété	Type	Description
PaymentAuthorizationUrl	String	URL qui doit être utilisée par le partenaire pour rediriger l'utilisateur vers la page de paiement Izly Partner.
OperationSMoney	JSON Object	<p>Détails de l'opération Xpollens nouvellement créée.</p> <p>Cet objet est composé des propriétés suivantes :</p> <ul style="list-style-type: none"> • Id : identifiant du transfert chez Xpollens. • IdTransactionDebitIzly : identifiant de l'opération chez Xpollens • Amount : montant de l'opération. • Date : date de création de l'opération • Message : message de l'opération envoyé par le site partenaire • ClientOrderId : identifiant de l'opération chez le site web partenaire. • Status : état de l'opération.
Beneficiary	JSON Object	<p>Détails du compte monnaie électronique Xpollens du bénéficiaire du paiement.</p> <p>Cet objet est composé des propriétés suivantes :</p> <ul style="list-style-type: none"> • Id : identifiant du compte monnaie électronique. • ClientId : identifiant de l'application du site web partenaire sur Izly.

Ci-dessous un exemple d'initialisation de paiement réalisé sur POSTMAN sur la PREPROD :

1 Verbe POST

2 URL à appeler

3 Les Headers

4 Le Body

5 La réponse retournée avec le PaymentAuthorizationUrl

```

{
  "PaymentAuthorizationUrl": "https://rest-pp.izly.fr/api/Izly/WebPayments/50689/Authorize?token=1-1EJagy&software=39",
  "OperationSMoney": {
    "Id": 50689,
    "IdTransactionDebitIzly": 59317,
    "Amount": "1.00",
    "Date": "03/05/2021 15:37:30",
    "Status": "Waiting",
    "Message": "test 1",
    "ClientOrderId": "1000026",
    "CallbackParameters": null
  },
  "Beneficiary": {
    "Id": "49915",
    "AppIdentifier": "Paris-78-78-78-78"
  }
}

```

b. Redirection de l'utilisateur vers la page d'autorisation de paiement

Après avoir initialisé le paiement, le site web partenaire doit rediriger l'utilisateur vers la page d'autorisation de paiement, en utilisant le « PaymentAuthorizationUrl » dans la réponse du web service.

c. Autorisation du paiement avec identifiant et mot de passe

Une fois redirigé, l'utilisateur verra la page d'authentification suivante :

- 1 Le nom du site web partenaire.
- 2 Le montant à payer.
- 3 Champ identifiant et mot de passe avec un lien pour mot de passe oublié.
- 4 Bouton pour annuler le paiement avec Izly.
- 5 Bouton pour autoriser le paiement avec Izly.

A l'issue de ce formulaire, nous pouvons avoir trois scenarios possibles :

- L'utilisateur clique sur Annuler : le paiement sera annulé, et l'utilisateur sera redirigé vers une page lui confirmant que le paiement a bien été annulé.

Liens pour revenir vers la page d'accueil du site web partenaire.

- L'utilisateur clique sur Valider : la validation du paiement sera traitée par Xpollens en 2 étapes :
 - ⇒ Vérifier l'identifiant de l'Ad et son mot de passe, par la suite le statut du compte de l'utilisateur s'il est éligible de s'authentifier et de faire le paiement (pas d'impayés à régulariser).
 - ⇒ Vérification de solde du compte de l'Ad.

Si la transaction est validée, l'utilisateur sera redirigé vers la page lui confirmant le bon déroulement du paiement.

Païement effectué avec Izly



Votre paiement de 1,00 € à destination de Roxico a bien été effectué.

Un email de confirmation va vous être envoyé.

[Revenir sur le site marchand](#)

1

1

Lien vers la page de retour après paiement du site web partenaire.

- L'utilisateur clique sur « Mot de passe oublié ? », celui-ci entre dans un processus de récupération de mot de passe. Après récupération de son mot de passe, l'AD devra régénérer un paiement côté site partenaire.

d. Appel du callback du site web partenaire

Un appel du callback du site web partenaire sera effectué dans le traitement de la validation du paiement de l'utilisateur, pour notifier le partenaire de la réalisation ou de l'échec du paiement côté Izly.

Le callback sera un appel POST effectué entre le serveur Izly et le serveur du site web partenaire.

Ci-dessous une description de l'appel du callback :

Resource URL	\$callbackUrl	\$callbackUrl : URL du callback renseigné dans le back-office Izly, sous le champ « URL de callback » lors de la création du point d'encaissement de type Izly Partner.
Method	POST	

- Header :

Clé	Valeur	Description
-----	--------	-------------

Content-Type	application/json	
--------------	------------------	--

- Body :
Le Body sera un objet JSON avec les propriétés suivantes

Propriété	Type	Description
OperationSMoney	JSON Object	Détails de l'opération Xpollens. Cet objet est composé des propriétés suivantes : <ul style="list-style-type: none"> • Id : identifiant du transfert chez Xpollens. • IdTransactionDebitIzly : identifiant de l'opération chez Xpollens • Amount : montant de l'opération. • Date : date de création de l'opération. • Message : message de l'opération envoyé par le site partenaire. • ClientOrderId : identifiant de l'opération chez le site web partenaire. • Status : état de l'opération. • CallbackParameters : paramètres envoyés par le site partenaire au moment de l'initialisation du paiement.
Beneficiary	JSON Object	Détails du compte monnaie électronique Xpollens, du bénéficiaire du paiement. Cet objet est composé des propriétés suivantes : <ul style="list-style-type: none"> • Id : identifiant du compte monnaie électronique. • ClientId : identifiant de l'application du site web partenaire sur Izly.
User	JSON Object	Détails du compte monnaie électronique Xpollens, de l'utilisateur. Cet objet est composé des propriétés suivantes : <ul style="list-style-type: none"> • Id : identifiant du compte monnaie électronique. • Email : email de l'utilisateur • ClientCode : code client CROUS • NumeroAffectation : numéro d'affectation CROUS
Error	JSON Object	Détails des erreurs lors du traitement. Cet objet est composé des propriétés suivantes : <ul style="list-style-type: none"> • Code : le code d'erreur (0 en cas de succès). • Message : le message d'erreur

- Response :

Xpollens s'attend à recevoir une réponse avec un statut http 200.

Dans le cas échéant, Xpollens s'attend à un Objet JSON « **Error** » avec les propriétés ci-dessous :

Propriété	Type	Description	Valeur si Succès	Valeur si erreur
Code	Int	Code d'erreur	0	\$codeErreur
ErrorMessage	String	Message d'erreur	vide	\$messageErreur

2- Récupération d'informations sur un paiement

Les détails d'un paiement sont accessibles à tout moment par le site web partenaire, grâce au web service décrit ci-dessous :

Clé	Valeur	Description
Resource URL	https://rest-pp.izly.fr/api/Izly/WebPayments/{OperationId}	{OperationId} : Id du transfert chez Xpollens retourné par la suite de l'initialisation du paiement, dans la propriété <code>OperationSMoney.Id</code>
Method	GET	

- Header

Clé	Valeur	Description
Authorization	Basic \$apiKey	\$apiKey: Clé d'API du partenaire La valeur est renseignée dans le back-office Izly, dans la page d'édition du point d'encaissement de type Izly Partner, sous le champ « Clé d'API » comme montré dans la capture d'écran I. 1- d.
Content-type	application/vnd.Xpollens.v1+json	
AppIdentifier	\$AppIdentifier	\$AppIdentifier = Identifiant du site partenaire chez Izly. La valeur est renseignée dans le BO, dans la page d'édition du point d'encaissement de type Izly Partner, sous le champ « Identifiant de l'application client », comme montré dans la capture d'écran I. 1- d.

- Response

La réponse sera un objet JSON avec les propriétés suivantes

Propriété	Type	Description
OperationId	long	Identifiant du transfert chez Xpollens.
IdTransactionDebitIzly	long	Identifiant de l'opération chez Xpollens
Amount	decimal	Montant de l'opération
PaymentDate	DateTime	Date de création de l'opération
Message	string	Message de l'opération envoyé par le site partenaire
OrderId	string	Identifiant de l'opération chez le site web partenaire
OperationStatus	String	État de l'opération
BeneficiarySmoneyId	string	Identifiant du compte monnaie électronique
BeneficiaryClientId	string	Identifiant de l'application du site web partenaire sur Izly
CustomerSmoneyId	string	Identifiant du compte monnaie électronique

CustomerEmail	string	Email de l'utilisateur
CustomerClientId	string	Code client CROUS

Exemple d'un objet de la réponse :

```
{
  "OperationId": 52795,
  "IdTransactionDebitIzly": 63535,
  "Amount": 1.00,
  "BeneficiarySmoneyId": "49915",
  "BeneficiaryClientId": "Paris-78-78-78-78",
  "CustomerSmoneyId": "6000050617",
  "CustomerEmail": "test@test.com",
  "CustomerClientId": "patricka23",
  "OperationStatus": "Completed",
  "Error": "",
  "ClientParameters": null,
  "Message": "test 1",
  "PaymentDate": "2021-12-09T18:53:34",
  "OrderId": "100003685"
}
```

Si l'opération n'existe pas , Xpollens retourne un Objet JSON « **Error** » avec les propriétés ci-dessous :

Propriété	Type	Description
Code	Int	Code d'erreur
ErrorMessage	String	Message d'erreur
Title	String	Titre du message d'erreur
Priority	Int	Priorité de l'erreur

Exemple d'un objet de la réponse en erreur :

```
{
  "Code": 356,
  "ErrorMessage": "Opération introuvable",
  "Title": "L'opération ne peut pas aboutir",
  "Priority": 2
}
```

3- Remboursement d'un paiement

Le remboursement d'un paiement est réalisable par le site web partenaire, en utilisant le web service décrit ci-dessous :

Clé	Valeur	Description
Ressource URL	https://rest-pp.izly.fr/api/izly/WebPayments/{OperationId}/Refund?clientOrderId={ClientOrderId}	{OperationId} : Id du transfert chez Xpollens retourné par la suite de l'initialisation du paiement, dans la propriété OperationSMoney.Id {ClientOrderId} : Identifiant de l'opération chez le site web partenaire
Method	POST	

- Header

Clé	Valeur	Description
Authorization	Basic \$apiKey	\$apiKey : Clé d'API du partenaire La valeur est renseignée dans le back-office Izly, dans la page d'édition du point d'encaissement de type Izly Partner, sous le champ « Clé d'API » comme montré dans la capture d'écran I. 1- d.
Content-type	application/vnd.Xpollens.v1+json	
AppIdentifier	\$AppIdentifier	\$AppIdentifier : Identifiant du site partenaire chez Izly. La valeur est renseignée dans le BO, dans la page d'édition du point d'encaissement de type Izly Partner, sous le champ « Identifiant de l'application client », comme montré dans la capture d'écran I. 1- d.

- Response

La réponse sera un objet JSON avec les propriétés suivantes :

Propriété	Type	Description
OperationId	long	Identifiant du transfert chez Xpollens.
IdTransactionDebitIzly	long	Identifiant de l'opération chez Xpollens
RefundDate	DateTime	Date de remboursement de l'opération
Amount	decimal	Montant de l'opération
OperationStatus	string	État de l'opération
ClientOrderId	string	Identifiant de l'opération chez le site web partenaire

Exemple d'un objet de la réponse en cas de succès :

```
{
  "OperationId": 52795,
  "IdTransactionDebitIzly": 63535,
  "RefundDate": "2021-12-09T18:53:34",
  "Amount": 1.00,
  "OperationStatus": "Refund",
  "ClientOrderId": "100003685"
}
```

Si l'opération n'existe pas , Xpollens retourne un Objet JSON « **Error** » avec les propriétés ci-dessous :

Propriété	Type	Description
Code	Int	Code d'erreur
ErrorMessage	String	Message d'erreur
Title	String	Titre du message d'erreur
Priority	Int	Priorité de l'erreur

Exemple d'un objet de la réponse en cas d'erreur :

```
{
  "Code": 356,
  "ErrorMessage": "Opération introuvable",
  "Title": "L'opération ne peut pas aboutir",
  "Priority": 2
}
```