



Implementation by:

LigdiCash SARL



PAYMENT SOLUTIONS AGGREGATOR

Contents

1. General Presentation of Ligdicash.....	2
1.1. Description	2
1.2. PAYIN and PAYOUT API integrations	4
1.2.1. How to get Tokens and Apikey	5
1.2.2. PAYINS integrations through plugins modules	7
1.2.3. Integration PAYIN LigdiCash API with redirection.....	8
1.2.4. Integration PAYIN LigdiCash API without redirection.....	10
1.2.5. Checking the status of the PAYIN request (With or Without redirection)	12
1.2.6. PAYOUT CUSTOMER API integration: CUSTOMER refund	13
1.2.7. PAYOUT MERCHANT API integration: MERCHANT refund.....	15
1.2.8. Checking the status of a PAYOUT (CUSTOMER OR MERCHANT withdrawal)	17
1.3. Creation of payment links: PAYLINK.....	18
1.4. Interoperability	20
2. Constraints	20
2.1. The volume of information and security	20
2.2. Payment systems	21
2.3. LigdiCash architecture.....	22
2.4. Les langues	22
2.5. Currencies.....	22
2.6. Accommodation and Security	23
3. Useful links (Platforms in Production)	24

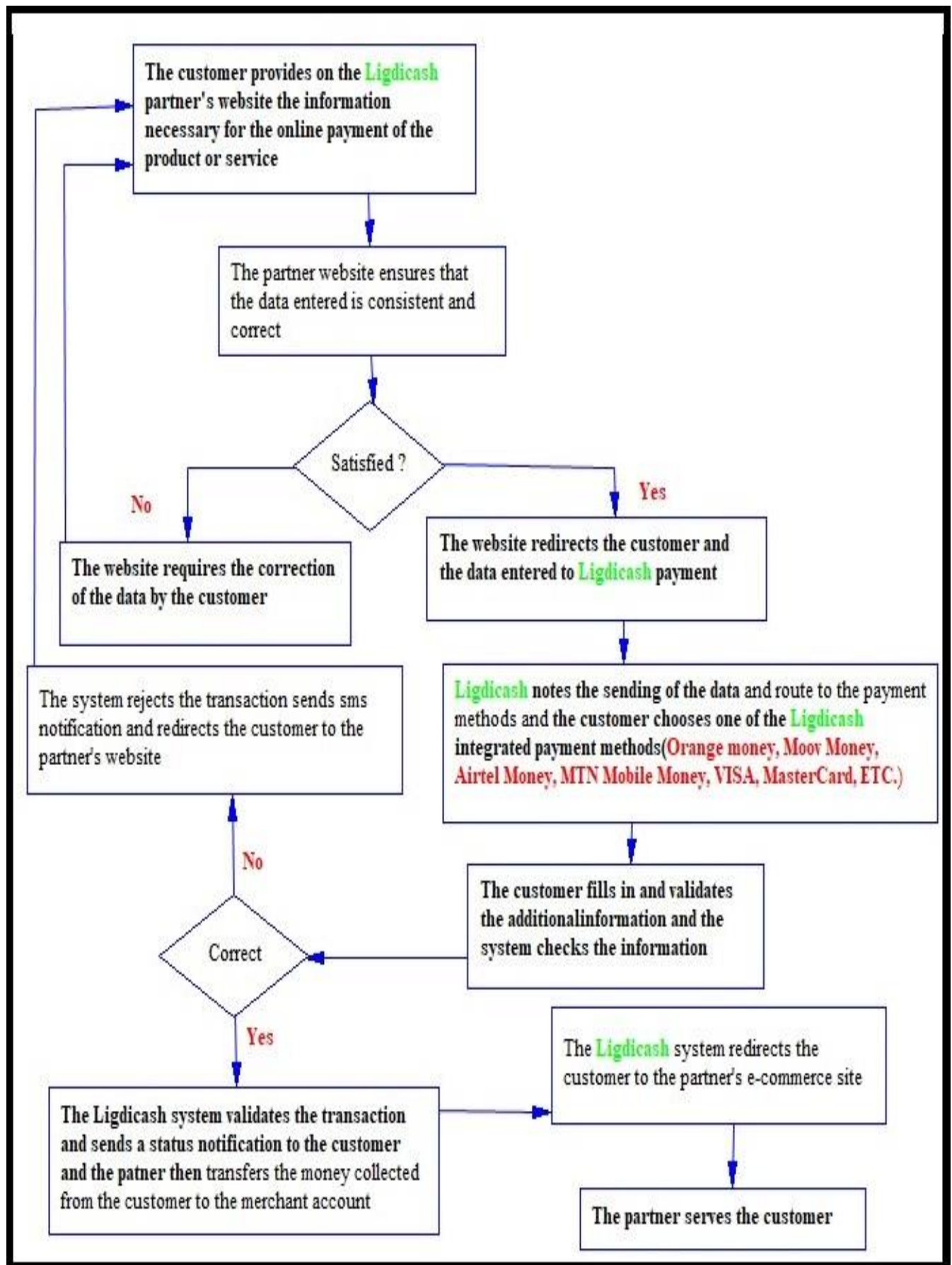
1. General Presentation of Ligdicash

1.1. Description

« **LigdiCash** » is a payment solution, a payment gateway, which aims to secure and optimize online trade between "merchants" and their "customers". It works as follows:

- The user (*merchant's customer*) must have one of the accounts (Mobile Money, electronic wallet, VISA Card, MasterCard, etc.) integrated into the Ligdicash API, but he is not obliged to have or create an "LigdiCash" account;
- An "e-merchant (www.logoba.net)" integrates **LigdiCash** on his site or in his application, a user of this site, to pay what he wants, can decide to use "LigdiCash"
- The user (customer) can decide to pay without creating a LigdiCash account but go through one of the payment methods (Orange Money, MOBICASH, Africard VISA, etc.) integrated with LigdiCash according to his choice to make his payment.
- Once connected to "LigdiCash", the user validates his payment by clicking on PAY after having filled in the additional information.
- Once the payment has been validated, "LigdiCash" credits the LigdiCash merchant account of the "e-merchant" and refers the user to the merchant site, for example <https://www.equiterroir.com/> or <https://www.logoba.net/> or <https://www.vishop-bf.com/>).

See the diagram below of the payment process for a Pay customer:



1.2. PAYIN and PAYOUT API integrations

« **LigdiCash** » is a simple and secure online payment platform which therefore revolves around web services available in the form of APIs that "merchants" will integrate into their projects. Integration will be done through **POST/GET** methods with examples of **requests/responses** in **JSON** format or via plugins.

NB .: For non-IT professionals, who do not have enough computer knowledge, LigdiCash has another means of integration which is "**PAYLINKS**". All ligdiCash merchants need to create in their merchant account, payment links for each online product or service and then copy each payment link to paste in the articles of its website, its publication pages of its social networks (**Facebook,WhatsApp,twiter,etc.**)


The platform is intended to be a universal payment solution because by integrating it into a website or application project, the "merchant" will be able to accept payments made by mobile money, VISA and others and without having to modify their computer code.

Create a ligdicash account and application in order to have your LigdiCash API parameters

To facilitate integration, you should create an account by registering at the platform : <https://client.ligdicash.com/registration> in order to have credit.

An application at **LigdiCash** level corresponds to your **website or web or mobile applications and others**. You have the possibility to create several **APPLICATIONS** in the same **LigdiCash account** corresponding to your different projects.

Once done, follow the next steps:

- Sign in your account
- Click on the menu **Mes projets=>Mes applications**
- Create your applications by clicking on the button 
- Fill out the form indicating the name, description and password of your application
- Click on the **Validate button (or modify if already created)**
- Click the button **YES**

Statuts des applications :





Validated application



Application awaiting validation



Suspended application

NB. : When your application is not activated, then you must contact technical support by mail info@ligdicash.com to help you chat on <https://Ligdicash.com>

If your application is valid, you can now perform your integration tests and check your application balances in the menu **PROJETS=>MES COMPTES ;**

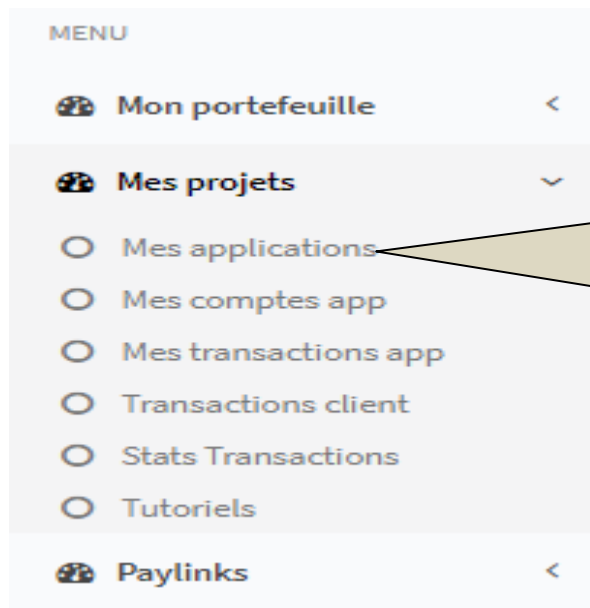
Application	Description	Statut	
LafiDocTest	Rendez-vous medical en ligne		IP
LogobaTest	Marketplace: une plateforme de vente en ligne		IP
DOUNASSO	réservation des chambres des hôtels et auberges en ligne		IP

If all the tests are good you can now go to production.

1.2.1. How to get Tokens and Apikey

Now that you have created your **PRODUCTION** here:


<https://client.ligdicash.com/registration> and your app is activated then you can now recover your **Apikey** and your **Tocken** for the integrations in the codes of your payment platforms. You still have to **download** the plugin that suits the technologies you use for your developments or use the examples of **API requests/responses** in **JSON** format in your programming languages.



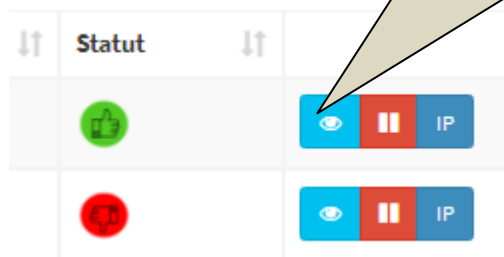
Dans votre menu

Mes projets=>Mes applications

And you will have the list of your

created applications or the Button  to add an application


Click on the icon above to view and collect your **Apikey** and **Authorization**









When your tests are successful, follow the next steps for your **Apikey** and the **Token**:

Log in to your production account <https://client.ligdicash.com> or create a production account <https://client.ligdicash.com/registration>

Once done, follow the next steps:

- Sign in your account
- Click on the menu **Mes projets=>Mes applications**
- Create your applications by clicking on the button 
- Fill out the form indicating the name, description and password of your application
- Click on the **Validate button (or modify if already created)**
- Click the button **YES**

And retrieve your access parameters as shown in the figure above.

Statut	
	 
	  IP

Click on the icon above to view and collect your **Apikey** and **Authorization**

1.2.2. **PAYINS** integrations through plugins modules

In its current version, the Ligdicash platform is already available with **downloadable** plugins modules: <https://ligdicash.com> in the **API MENU**.

- Plugin **PHP**
- Plugin **Joomla**
- Plugin **WordPress**
- **Woocommerce**
- Plugin **Android**
- Plugin **Prestashop**
- Plugin **WinDev**
- Plugin **Joomla**
- Etc.

1.2.3. Integration **PAYIN LigdiCash API** with redirection

This **PAYIN LigdiCash API** is used to request payment from a consumer (Payer). The Payer is redirected to a set of payment methods (*Mobile money, VISA Cards, MarsterCard, GIM ECOWAS, etc.*), the Payer chooses the payment method and follows the procedure of the method and authorizes the payment. After authorization, the transaction will be executed. Is used to request payment from a consumer (Payer). The consumer (payer) will be asked to authorize the payment. After authorization, the transaction will be executed.

This procedure is more interesting in that your developer no longer needs to touch the code after this integration, because each time Ligdicash integrates a new payment method in a country. We just need to activate the method on your application, it appears at your Payers customers directly without your developer making any code changes.

Sending the request

Method : **POST**

<https://app.ligdicash.com/pay/v01/redirect/checkout-invoice/create>

headers :

Apikey : contains the **master key** generated during the creation of your application

Authorization : contains the word **Bearer** followed by a **space** then the **token** generated during the creation of your application

payload (format json)

```
{
  "commande": {
    "invoice": {
      "items": [
        {
          "name": " Product or Service Name ",
          "description": "Product Description or Service ",
          "quantity": 1,
          "unit_price": 100,
          "total_price": 100
        }
      ],
      "total_amount": 100,
      "devise": "XOF",
      "description": "Description of the invoice (Purchase of fruit juice)",
      "customer": "Leave this field blank",
      "customer_firstname": "The customer firstname ",
      "customer_lastname": "The customer_lastname ",
      "customer_email": "email of customer for sample tester@gligdicash.com"
    },
    "store": {
      "name": "Name of your site or shop",
      "website_url": "url of your site or shop"
    },
    "actions": {
      "cancel_url": "request cancellation url",
      "return_url": "return url after validation of the request",
      "callback_url": "Processing URL after customer payment validation "
    },
    "custom_data": {
      "transaction_id": "2021000000001"
    }
  }
}
```

Réponse (format json)

```
{
  "response_code": "contains 00 if successful. if different from 00 then failure ",
  "token": " transaction token. you must keep it on your platform",
  "response_text": " contains the url of the payment validation page (page to which you must redirect the customer for him to proceed to payment)",
  "description": " text message describing the result of the request",
  "custom_data": {
    "transaction_id": "2021000000001 ID if your transaction",
    "logfile": "202110210048426170b8ea884a9"
  },
  "wiki": ["Description of the error code obtained response text and response code "]
}
```

1.2.4. Integration **PAYIN LigdiCash API** without redirection

This **PAYIN LigdiCash API** is used to request payment from a consumer (Payer). The Payer is not redirected during payment, because the integration is done directly method by method by the developer. The Payer follows the procedure of the method and authorizes the payment directly without redirection. After authorization, the transaction will be executed. The consumer (payer) will be asked to authorize the payment. After authorization, the transaction will be executed.

Warning !!! This feature is not available on the test platform

Sending the request

Method : **POST**

url : <https://app.ligdicash.com/pay/v01/straight/checkout-invoice/create>

headers :

Apikey : contains the main key generated during the creation of your application

Authorization : contains the word Bearer followed by a space then the token generated during the creation of your application

Accept : application/json

Content-Type : application/json

payload (format json)

```
{
  "commande": {
    "invoice": {
      "items": [
        {
          "name": " Product or Service Name ",
          "description": "Product Description or Servedce ",
          "quantity": 1,
          "unit_price": 100,
          "total_price": 100
        }
      ],
      "total_amount": 100,
      "devise": "XOF",
      "description": "Description of the invoice (Purchase of fruit juice)",
      "customer": " the phone number of the customer to pay (format: 22676275726 or 22997761182)",
      "customer_firstname": "The customer firstname ",
      "customer_lastname": " The customer_lastname ",
      "customer_email": "email of customer for sample tester@gligdicash.com",
      "external_id": "",
      "otp": "OTP code of the transaction (only for orange money, otherwise leave blank"
    },
    "store": {
      "name": "Name of your site or shop",
      "website_url": "url of your site or shop"
    },
    "actions": {
      "cancel_url": "request cancellation url",
      "return_url": "return url after validation of the request",
      "callback_url": "Processing URL after customer payment validation "
    },
    "custom_data": {
      "transaction_id": "2021000000001"
    }
  }
}
```

Réponse (format json)

```
{
  "response_code": "contains 00 if successful. if different from 00 then failure ",
  "token": " transaction token. you must keep it on your platform",
  "response_text": " contains the url of the payment validation page (page to which you must redirect the customer for him to proceed to payment)",
  "description": " text message describing the result of the request",
  "custom_data": {
    "transaction_id": "2021000000001 ID of your transaction",
    "logfile": "202110210048426170b8ea884a9"
  },
  "wiki": ["Description of the error code obtained response text and response code "]
}
```

1.2.5. Checking the **status** of the **PAYIN** request (With or Without redirection)

Checking the **status** of the payment request

Method : GET

url : <https://app.ligdicash.com/pay/v01/redirect/checkout-invoice/confirm//?invoiceToken=>

headers

Apikey : contains the master key generated during the creation of your application

Authorization : contains the word bearer followed by a space then the token generated during the creation of your application

URL parameters :

invoiceToken : contains the token you received after sending the request

Response (format json)

```
{
  "response_code": "contains 00 if successful. if different from 00 then failure",
  "token": "invoice id. you must keep it on your platform",
  "response_text": "text message describing the result of the request",
  "status": "The status of the transaction: pending(pending) completed(validated),
             notcompleted(canceled)",
  "custom_data": [
    {
      "id_customdata": "contains the customer's custom_data ID",
      "id_invoice": "contains the customer's order",
      "keyof_customdata": "idinvoice", // constant fields
      "valueof_customdata": "contains the ID or invoice of the Payer customer"
    }
  ],
  "operator_id": "contains the operator ID of the chosen method",
  "operator_name": "contains the name (Example:OrangeMoney) of the chosen operator",
  "external_id": "contains the ID of the chosen non-mobile money operator",
  "customer": "contains the number of the example customer 22676275726"
}
```



1.2.6. PAYOUT CUSTOMER API integration: CUSTOMER refund

Warning!!! This feature is not available on the test platform

This **PAYOUT CLIENT API** is used to issue **refunds** to your customers. Your customers must have a wallet account at LigdiCash level. We recommend this PAYOUT CUSTOMER API because it allows you to avoid errors in the destination of **refunds** to your customers.

You can download a **demo** written in **PHP** for PHP developers here:

https://ligdicash.com/docs/ligdicash-php-gateway_withdrawal.zip

You can also use requests and responses in **JSON** formats:

Envoi de la requête

Methode : **POST**

url : <https://app.ligdicash.com/pay/v01/withdrawal/create>

headers :

Apikey : contient the main code generated during the creation of your application

Authorization: contient le mot Bearer suivi d'un space alors token generated during the creation of your application

Accept: application/json

Content-Type: application/json

payload (format json)

```
{
  "commande": {
    "amount": 500,
    "description": "Description of the content of the WITHDRAWAL",
    "customer": "contains the recipient's phone number (format:22676275726 or 22997761182)",
    "custom_data": {
      "transaction_id": "Contains your withdrawal transaction id"
    },
    "callback_url": " can contain your successful withdrawal URL, if processed at
your level, this is optional",
    "top_up_wallet":1
  }
}
```

Réponse (format json)

```
{
  "response_code": " contains 00 is successful. If different from 00 then failure ",
  "token": " transaction token. You must keep it on your platform ",
  "response_text": "Description",
  "description": " Message describing the result of the request ",
  "customdata": [],
  "wiki": ["Description of the error code obtained response_text et response_code "]
}
```

1.2.7. PAYOUT MERCHANT API integration: MERCHANT refund

This **PAYOUT MERCHANT API** is used to make your merchant refunds or refunds to your customers who do not have a ligdicash account. In this case, the recipient is not required to have a ligdicash wallet account. With this PAYOUT MERCHANT API, the merchant is required to ensure at the level of his merchant application that the destination of the refund to the **mobile money or wallet account** is correct.

This API does the same as CUSTOMER PAYOUT except that there is no verification of the destination of your refund since the customer does not necessarily have a ligdicash account.

Take notice : This **PAYOUT MERCHANT API** is mainly used to make **CASHOUT MARCAHND** when the amount of your **CASHOUT** refund does not exceed **1,000,000 CFA** francs; Otherwise it must be done by a **bank transfer**

Sending the request

Methode : **POST**

url : <https://app.ligdicash.com/pay/v01/straight/payout>

headers :

Apikey : contient le **main code generated** during the **creation of your application**

Authorization: contient le mot **Bearer** suivi d'un **space** alors **token generated** during the **creation of your application**

Accept: **application/json**

Content-Type: **application/json**



payload (format json)

```
{
  "commande": {
    "amount": 500,
    "description": "Description of the content of the WITHDRAWAL",
    "customer": "contains the recipient's phone number (format:22676275726 or 22997761182)",
    "custom_data": {
      "transaction_id": "Contains your withdrawal transaction id"
    },
    "callback_url": " can contain your successful withdrawal URL, if processed at
your level, this is optional"
  }
}
```

Réponse (format json)

```
{
  "response_code": " contains 00 is successful. If different from 00 then failure ",
  "token": " transaction token. You must keep it on your platform ",
  "response_text": "Description",
  "description": " Message describing the result of the request ",
  "customdata": [],
  "wiki": ["Description of the error code obtained response_text et response_code "]
}
```


1.2.8. Checking the status of a PAYOUT (CUSTOMER OR MERCHANT withdrawal)

Vérification du **statut** d'un PAYOUT client OU marchand: **callBack**

Method : GET

Production url: <https://app.ligdicash.com/pay/v01/withdrawal/confirm/?withdrawalToken=>

headers :

Apikey : contains the master key generated when creating your application

Authorization : contains the word Bearer followed by a space then the token generated when creating your application

URL parameters:

withdrawalToken: contains the withdrawal token that you received after sending the payout request

Réponse (format json)


```
{
  "response_code": " contains 00 is successful. If different from 00 then failure",
  "token": " withdrawal transaction token. You must keep it on your platform ",
  "montant": " The amount of the customer withdrawal transaction",
  "response_text": " contains the text message of the withdrawal transaction ",
  "status ": " The status of the transaction: pending or completed(when validated)"
  "custom_data": " contains customer custom_data",
  "operator_id": " contains the ID of the mobile money operator of the chosen method ",
  "operator_name": " contains the name (Example:OrangeMoney) of the chosen operator",
  "external_id": " contains the ID of the external operator(VISA,MASTERCARD,GIM CEDEAO, ETC)",
  "customer": " contains the customer number Pay example 22676275726"
}
```

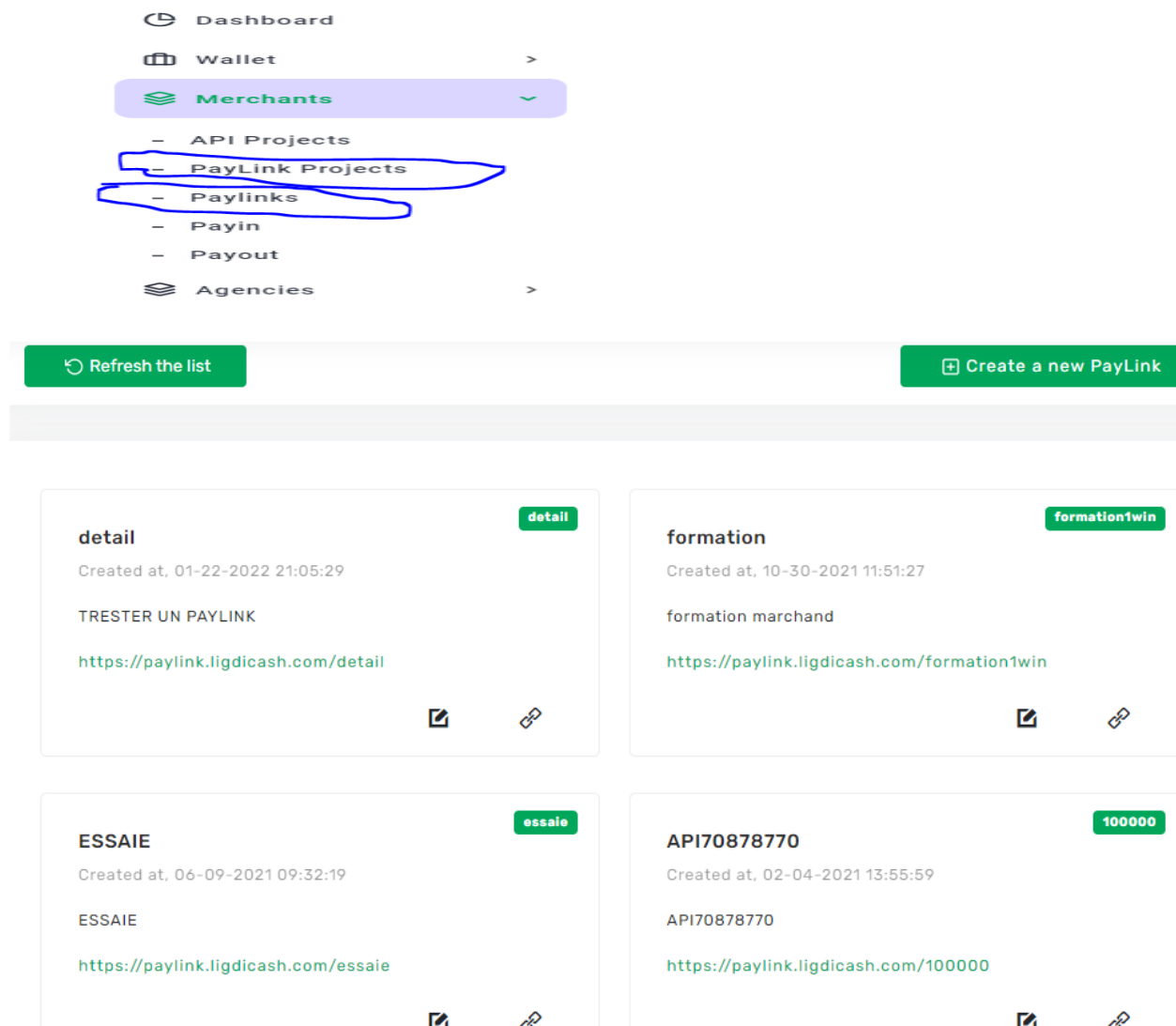
1.3. Creation of payment links: PAYLINK

To create a PayLink payment link, you must have an activated LigdiCash merchant account in order to have access to the PayLinks MENU to create your payment links for your fundraising, Social Shop, Webpay, MobilePay for the sales of your products or services online. This module is very interesting for **E-merchants, E-merchants, Marketplates, E-services, Microfinances, SMEs, NGOs ...**) who do not have computer programming skills to be able to create payment links for their products or service that it sends to their customers in order to receive payments.

Example of Paylink payment link : <https://paylink.ligdicash.com/logoba>

You can create as many payment links for your products or services.

To create your paylink payment links, you must login to your **LigdiCash** account. In the **Paylink Projects=>Create a new paylink**. Then click on the button 



Dashboard Menu:

- Dashboard
- Wallet
- Merchants**
 - API Projects
 - **PayLink Projects**
 - **Paylinks**
 - Payin
 - Payout
- Agencies

Buttons:

- Refresh the list
- Create a new PayLink

Card Title	Card Type	Created at	Description	Paylink URL
detail	detail	01-22-2022 21:05:29	TRESTER UN PAYLINK	https://paylink.ligdicash.com/detail
formation	formation1win	10-30-2021 11:51:27	formation marchand	https://paylink.ligdicash.com/formation1win
ESSAIE	essaie	06-09-2021 09:32:19	ESSAIE	https://paylink.ligdicash.com/essaie
API70878770	100000	02-04-2021 13:55:59	API70878770	https://paylink.ligdicash.com/100000

For each product or service you can create a Paylink. See the figure below:

Application	
DOUNASSO	
Libelle	
Libelle	
Description	
Veuillez saisir une breve description de ce PayLink	
Tarification	
Vente à valeur fixe	
Prix de vente unitaire	
0	
Ventes	
Ne pas limiter le nombre de ventes	
<input checked="" type="checkbox"/> Ne pas demander au client de preciser la quantité lors de la vente	
Durée de vie indéterminée	
Endpoint	
Nom court (sans espace)	
Action	
Sélectionnez l'action à effectuer apres paiement	
Annuler	Valider

You can choose **pricing, the stock limit or the duration of the sale** according to your needs:

Tarification
Vente à valeur fixe
Vente à valeur fixe
Vente à valeur variable
Collecte
Vente à valeur plancher

You can follow our **Youtube** channel: <https://youtu.be/hKg82rFQ-8M>

1.4. Interoperability

LigdiCash also aims to set up a certain interoperability between the different operators. Thus users will be able to transfer money regardless of the recipient's operator (*Orange Money, Moov Money, Moov Money, Airtel money, VISA, MasterCard, etc.*)

2. Constraints

2.1. The volume of information

The volume of information stored and processed will be significant. It is therefore necessary to set up an efficient database management system sized for your server. Your server must authorize the **SSL certification** to the trusted third party at the base of the electronic certification chain. It is she who issues and manages the digital certificates used to secure dematerialized exchanges and guarantee the identity of issuers.

2.2. Payment systems

In its current version, the platform takes into account payment systems:

Orange Burkina's mobile money payment :

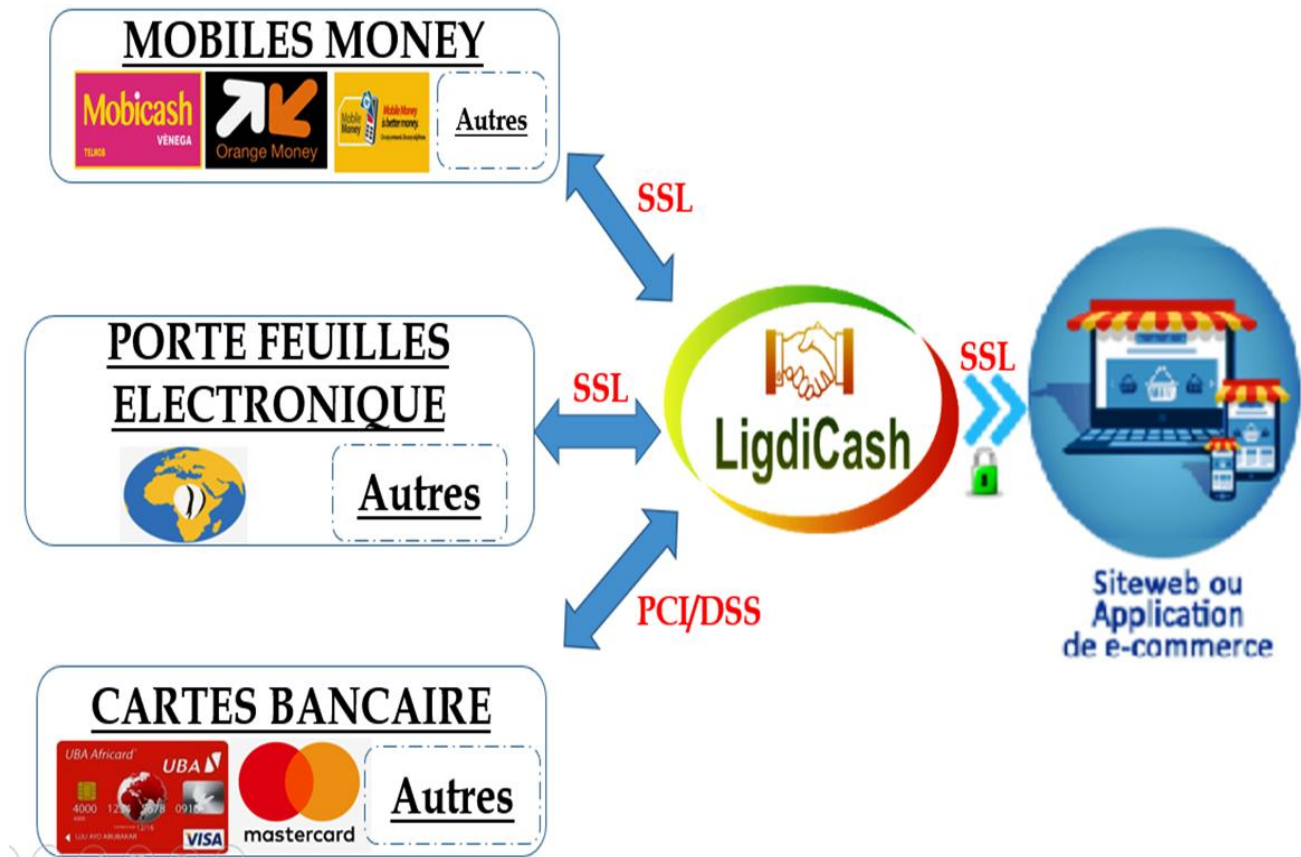
- ✓ Mobile money payment from Moov money Burkina
- ✓ Orange Mali mobile money payment
- ✓ Mtn Côte D'ivoire mobile money payment
- ✓ Moov Côte D'ivoire mobile money payment
- ✓ Mobile money payment from Mtn Benin
- ✓ Moov Benin's mobile money payment
- ✓ Moov Togo's mobile money payment
- ✓ Moov Mali's mobile money payment
- ✓ Orange Côte D'ivoire mobile money payment
- ✓ **VISA, MasterCard, GIM CEDEAO** card payments are already integrated with ligdicash including **PCI/DSS** certified **LigdiCash** servers.
- ✓ ...

We plan to do integrations also at the level of English speaking countries for Mtn Nigeria, Mtn Ghana, etc.

Electronic wallets such as **PAYPLUS, WIZALL** from **Banque Atlantique** or Coris Money from **Coris Banque** are being integrated.

2.3. LigdiCash architecture

ARCHITECTURE DE LIGDICASH



2.4. Les langues

In its initial version, the platform is exclusively available in **French** and **English**, however it is envisaged that in the short term, the platform will be available in other languages also depending on the countries of integration.

2.5. Currencies

In its current version, the platform only manages the currency "**Franc CFA**", however it is envisaged that in the short term, the platform will be available with the "**Cedis**" of Ghana and the "**Naira**" of Nigeria, the "**Euro**", "**Dollars**" and *other quotes* thereafter. In this sense, the management of conversion parities from one currency to another is also essential and we are working for this.

2.6. Accommodation and Security

We opted to host our solution on a dedicated and non-shared server. This ensures almost permanent availability and accessibility of the platform and security management as we see fit. Given the sensitive side of the information that passes through our platform, we also opted to subscribe to a Transport Layer Security (TLS) or Transport Layer Security certificate, its predecessor Secure Sockets Layer (SSL) and PCI DSS (in English: Payment Card Industry Data Security Standard) which represents the data security standard for the payment card industries, in order to protect the information exchanged between the user (Client) and "LigdiCash" and Firewall software.

To support security at the level of banking transactions, we have opted for a secure server with **PCI DSS CERTIFICATE OF CONFORMITY** approved by the **card brands Visa, MasterCard, American Express, Discover and JCB**. From this point of view your server must obey the secure communication protocol **https**.

3. Useful links (Platforms in Production)

NB. : The management of each platform is not the responsibility of **LigdiCash, but of the startup that signed an integration partnership agreement.**

➤ **Some examples of platforms in production**

- <https://izichange.com/> (Plateforme de change en ligne espace UEMOA)
- <https://fasochange.com/> (Plateforme de change en ligne au Burkina, mali et cote d'ivoire)
- <https://www.lafidoc.net> (Rendez-vous médical en ligne)
- www.logoba.net (Multiplateforme MarketPlace de vente en ligne)
- <https://www.equiterroir.com> (Vente en ligne de produits agricoles et artisanaux)
- <https://www.intermoney.africa> (Recharges, transferts de compte en compte)
- <https://www.makabine.net> (Recharges, transferts, e-tickets)
- <https://www.dounasso.com> (réservation des chambres des hôtels et auberges en ligne)
- www.shopexpressburkina.net (MarketPlace de vente en ligne)
- www.fasoeshop.com (MarketPlace de vente en ligne)
- www.1xbet.com (paris sportifs en ligne).

NB. : NB. : **LigdiCash currently has 500 Integrated Website / Apps**

LIGDICASH IS DOWNLOADABLE ON PLAYSTORE:

<https://play.google.com/store/apps/details?id=com.ligdicash.ligdicash&hl=fr>

