GameCredits API Documentation

Quick start

This is a quick overview of how the GameCredits developer API works. If wish to get a quick feel for how to implement the API in your next game, read on!

Before you start

First of all, you will need to register for a developer account (which you probably already did since you are reading this). Once you have successfully logged in, go ahead and create a new App in the app tab inside the Developer Center.

On your game server, create a receiving URL which will handle the invoice notifications from the GameCredits API whenever a user pays for an invoice.

You are now ready to begin using the GameCredits API!

Quick overview

1.Create an invoice for the user by sending an invoice creation request to the GameCredits API server

2.Send the user to the invoice's payment_url so they can pay for the invoice

3.Receive a notification from the GameCredits API server once the user has paid for the invoice

That's it! After the user has paid for the invoice, the funds will appear shortly in your GameCredits wallet.

An example using cURL

cURL is a handy tool (on UNIX based systems) that is used to send HTTP requests from the command line. Let's test the API and make an invoice using cURL by typings the following command in your command line:

```
curl -H "Authorization: d5c55d0b-8012-4f94-bb8b-0361f720324e" \
--data "app=22&item_name=Sword&item_description=Very Cool Sword&\
item_price=0.99&item_currency=USD&is_test=true" \
https://wallet.gamecredits.org/api/app/22/invoices/
```

Now let's analyze this command:

- `curl` - executes the cURL command line tool
- `-H "Authorization: d5c55d0b-8012-4f94-bb8b-0361f720324e"` - sets the HTTP request header to your API token. You can find your API tokens in theApplications tabinside the Developer Center.
- `--data` - this argument specifies that we are sending a POST request and it expects a string of POST parameters separated by &.
- `https://wallet.gamecredits.org/api/app/22/invoices/`- the URL of the API's invoice system.

Now let's analyze the POST parameters sent:

- `app=22`- this is your application's ID, you can see and manage all of your applications in the Developer Center.
- `item_name=Sword`- name of your item that will be displayed to the buyer.
- `item_description=Very Cool Sword`- description of the item.
- `item_price=0.99&item_currency=USD`- item price amount and currency, supported currencies are **USD** and **GAME**.

After sending this request if the invoice creation was successful (if you entered the correct information it should be) you should receive a response from the server that looks like this:

```
        {
                "id":  51,
                "app":  22,
                "item_name":  "Sword",
                "item_description":  "Cool sword",
                "item_price_game":  17.9664,
                "item_price":  0.99,
                "item_currency":  "USD",
                "data":  "",
                "redirect_url":  "null",
                "notify_url":  "null",
                "payment_url":  "https://wallet.gamecredits.org/api/pay/51",
                "status":  "new",
                "transaction_id":  ""
        }
```

Architecture

This is an overview of the architecture of the GameCredits API. It's highly recommended that you read this, but if you really don't have the time you can skip to the Unity section.

Payment Flow

Here is a brief step by step overview how to setup a payment for a single item in your app. Read it to get a feel of how to API works. The payment flow consists of **3 main steps**, each of the steps will be described in detail later in the documentation:

1. You **create an invoice** by sending a HTTP POST request to our API that contains:

   - All of the **relevant information about your item.** Like it's name, description and price.
   - Any **pass-through data** that you will use later to identify the buyer when we send you, or anything you want.
   - **Redirect URL:**a URL the buyer will be redirected to if the payment is successful. That is usually a some kind of a Thank you page.
   - **Notify URL:**a URL of your server endpoint where we will send you the notifications when the invoice status changes.

   If the invoice was created successfully you will receive the payment URL as the part of the response from the API, and then you can redirect the buyer to that URL where he can confirm the payment.

2. The **buyer confirms the purchase** and the system sends you a notification that the invoice status has changed.

3. You handle the notification anyway you want, usually that means unlocking the item for the buyer. This where you read the pass-through data you provided during invoice creation from the notification, identify the buyer and you take the necessary action.

Invoice creation using HTTP

The API is HTTP based and RESTful, which means you can interface with the API by sending HTTP requests and exchanging information encoded in the JSON format using **any programming language or tool**.

In order to create an invoice you should send a HTTP POST request to the following URL:

```
https://wallet.gamecredits.org/api/app/<your_app_id>/invoices/
```

You can see all of your app id's in the developer center.

In order to authenticate with the API you must set the Authorization HTTP header in your request to:

```
Authorize: <your_app_token>
```

The request must contain data about the invoice you want to create. Read about it in the next section.

Invoices

Invoices are a core element of the Payment API. You use them to request and receive payments.

Invoice required parameters:

Parameter name	Description
`app`	Your application's ID
`item_name`	Name of the item, for example "*Sword of Orc slaying"*
`item_description`	Short description of the item like "*It slays Orcs"*
`item_price`	Price of the item, the currency is determined by the next parameter
`item_currency`	USD or GAME

Optional parameters:

Parameter name	Description
`data`	Any pass-through data you provided
`redirect_url`	URL the user will be redirected to if the payment is successful
`notify_url`	URL notifications will be sent to when the invoice status changes
`is_test`	True or False. Test invoices can be paid without any funds - used for testing the API

Server response

If the invoice creation was successful the server will return your created invoice with all of the data above and a few additional parameters, like:

```
payment_url = <some_url>
```

That's the URL you should redirect the buyer to, so he can confirm the purchase. Other additional parameters are:

Parameter name	Description
`id`	ID of your created invoice
`status`	Status of your invoice, which will be "*new"* on creation.
`transaction_id`	Blank field, it will be filled with the transaction ID when the payment is confirmed by the blockchain

Unity (C#)

We have provided a simple Unity script which you can use freely to see how to send requests to the GameCredits API.

Example script

Using the script is demonstrated in the **TestManager.cs** file:

- `CreateTestInvoice` is used to send an invoice creation request to the server. If the request was a success, the invoice will be printed out to the console.
- `GetInvoice` is used to fetch details for an invoice. If the request was a success, the invoice's details will be printed out to the console.

The **GameCreditsAPI.cs** file contains the wrapper code used to send requests to the GameCredits API. You may modify the code at will to suit your needs. However, this is not required - you can rely on your own methods, or use the ones in the TestManager file.

Invoice notifications

When the status of your invoice changes the system will send you a notification by issuing a HTTP POST request to your `notify_url`, if you provided the URL during invoice creation. Then you can read the status of the invoice, and if necessary take whatever action is needed to deliver the item to the user.

Invoice states

The notification will contain information about the status of your invoice. If the invoice is in a desired state (usually paid or confirmed) you can take whatever action is needed to deliver the purchased item to the user.

Parameter name	Description
`new`	Default status upon creation, invoice has not been paid yet.
`paid`	Invoice is paid but not confirmed
`confirmed`	Invoice payment has been confirmed by the blockchain
`failed`	Payment failed for some reason, usually because the user had insufficient funds to pay the Invoice.
`expired`	New invoices expire if they are not paid within 15 minutes