Service Integration Guide

v1.4.0

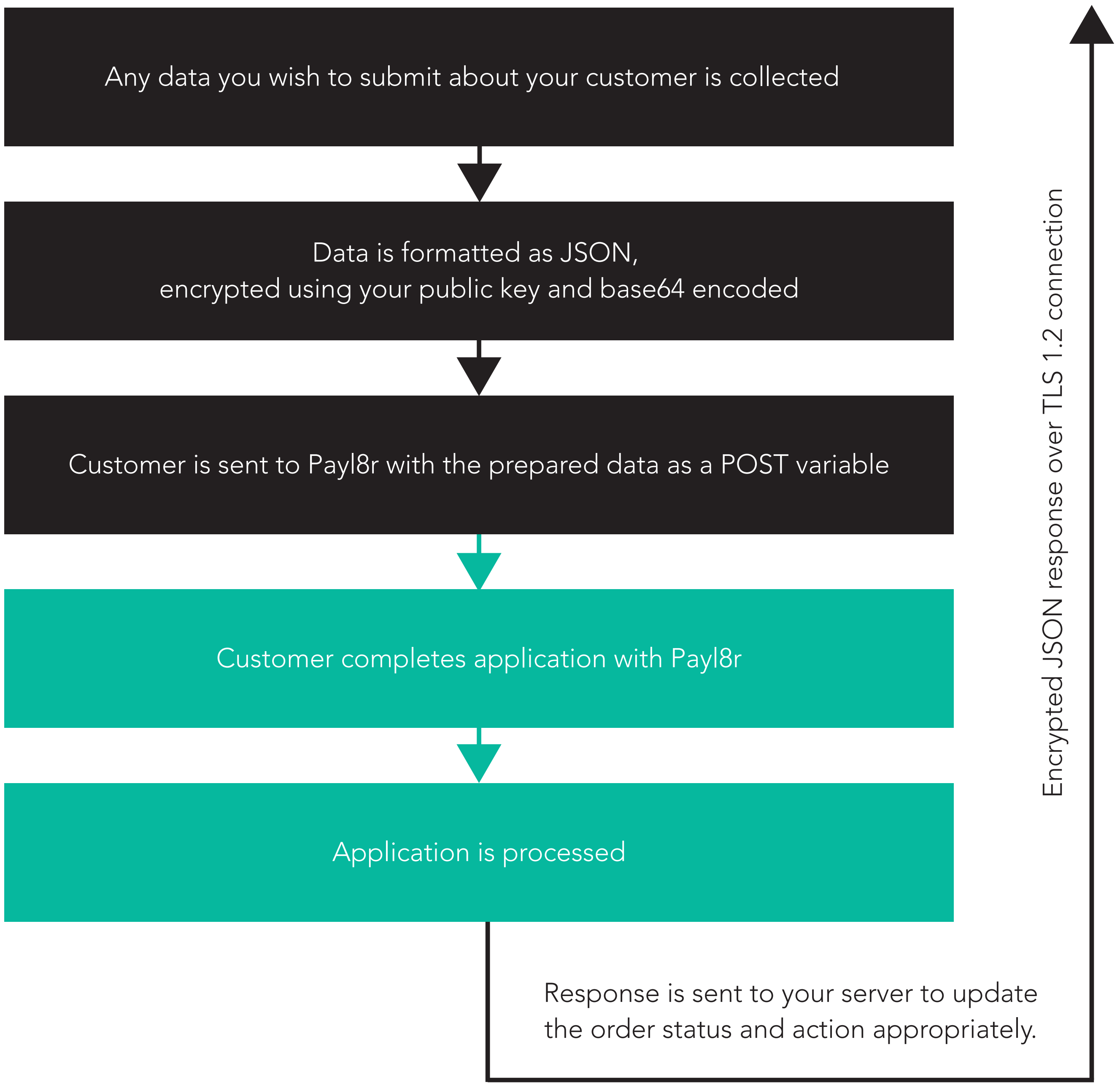
Contents

Before you begin

You should have received your username and public key along with this document. If you do not have these available, you will need to request them from admin@payl8r.com.

1. Summary

Payl8r provides services that allow a retailer to offer finance to their customers. Integration with Payl8r is kept as simple as possible without compromising on security. Below is a diagram to illustrate how a standard finance request works:


Any data you wish to submit about your customer is collected
Data is formatted as JSON, encrypted using your public key and base64 encoded
Customer is sent to Payl8r with the prepared data as a POST variable
Customer completes application with Payl8r
Application is processed
Response is sent to your server to update the order status and action appropriately.
Encrypted JSON response over TLS 1.2 connection

🟩 Processing on Payl8r's servers

⬛ Processing on client servers

2. Services

Standard Finance Request
https://payl8r.com/process
This is the primary URL used to process finance requests. A client requesting finance should be sent to this URL with appropriate POST data as noted under section 4.

Standard Finance Request: Response Delivery
When a standard finance request has been completed by the client, responses are sent to a URL you specify in the request. This should be used to update your order status and any related actions within your system. The URL must be secured using TLS 1.2. Full details on the responses to expect and format can be found in section 5b.

Please see our API Specification for more APIs

3. Data formatting and encryption

All data submitted to payl8r must be sent using POST to the service integration URLs listed in section 2. Data should be formatted into JSON and encrypted using your public key. Your public key works as both your password and to encrypt your data from being read or changed by the end user making the request. The main advantage of this is ease of use, as it allows all data to be securely submitted to us from a standard HTML form.

Example of JSON formatting:

```
{
     "username": "example",
     "request_data": {
          "return_urls": {
               "abort": "https://example.com/checkout/abort",
               "fail": "https://example.com/checkout/rejected",
               "success": "https://example.com/checkout/success",
               "return_data": "https://example.com/c832jdafs94kb"
          },
          "request_type": "standard_finance_request",
          "test_mode": 1,
          "order_details": {
               "order_id": 000012432,
               "description": "Basket with example.com",
               "currency": "GBP",
               "total", 142.99
          }
          // ANY ADDITIONAL DATA, SEE SECTION 4 FOR FULL LIST
     }
}
```

JSON should be first encrypted with your public key and then encoded with MIME base64 before transfer. Example of encryption using public key and data preperation for transfer in PHP:

4. Standard Finance Request

The data required to make a finance request is kept to a minimum. Additional data can be submitted, such as card data, customer name, and address. The more data you submit to us, the less we need to request from the customer to complete the process.

Field	Description	Data type	Required
username	Your payl8r username	String	Yes
request_data: return_urls: abort	Abort URL, see section 5	String	Yes
request_data: return_urls: fail	Fail URL, see section 5	String	Yes
request_data: return_urls: success	Success URL, see section 5	String	Yes
request_data: request_type	Set as "standard_finance_request"	String	Yes
request_data: test_mode	0 or 1, see section 7	Integer	Yes
request_data: order_details: order_id	Your order ID, must be unique	String	Yes
request_data: order_details: description	A description of the order	String	Yes
request_data: order_details: currency	Set as "GBP" only	String	Yes
request_data: order_details: total	The grand total for the order	Float	Yes
request_data: customer_details: firstnames	Client's first and middle names	String	No
request_data: customer_details: surname	Client's surname	String	No
request_data: customer_details: email	Client's email	String	No
request_data: customer_details: phone	Client's mobile phone number	String	No
request_data: customer_details: address	First line of address for customer	String	No
request_data: customer_details: city	Client's city e.g. Manchester	String	No
request_data: customer_details: country	Set as "UK" only	String	No
request_data: customer_details: postcode	Client's postcode without spaces	String	No
request_data: customer_details: dob	Client's dob formatted dd/mm/yyyy	String	No
request_data: customer_details: delivery_address	Delivery address for the order	String	Yes/No*
request_data: customer_details: delivery_city	City/Town being delivered to	String	Yes/No*
request_data: customer_details: delivery_postcode	Delivery postcode	String	Yes/No*
request_data: customer_details: delivery_country	Set as "UK" only	String	Yes/No*

* If your order has a different delivery address to the customer's address, this data is required.

```
{
      "username": "example",
      "request_data": {
            "return_urls": {
                  "abort": "https://example.com/checkout/abort",
                  "fail": "https://example.com/checkout/rejected",
                  "success": "https://example.com/checkout/success",
                  "return_data": "https://example.com/c832jdafs94kb"
            },
            "request_type": "standard_finance_request",
            "test_mode": 1,
            "order_details": {
                  "order_id": "000012432",
                  "description": "Basket with example.com",
                  "currency": "GBP",
                  "total", 142.99
            },
            // ADDITIONAL DATA
            "customer_details": {
                  "firstnames": "Larry John",
                  "surname": "Ridley",
                  "email": "ljr@example.com",
                  "phone": "07769123567",
                  "address": "Flat 140",
                  "city": "Manchester",
                  "country": "UK",
                  "postcode": "M51LN",
                  "dob": "29/05/1985",
                  "delivery_address": "15 Scale St",
                  "delivery_city": "Manchester",
                  "delivery_postcode": "M24LS",
                  "delivery_country": "UK"
            }
      }
}
```

An example of a HTML form to send formatted data to Payl8r is shown below. The HTML response is displayed in an iframe. This example uses PHP:

```
<?php
$key = 'file://keys/payl8r-crt.pem';
openssl_public_encrypt($json, $encrypted_json, $key);
// NOTE: $json contains the JSON from p8.
$base64_encrypted_json = base64_encode($encrypted_json);
?>
<form target="payl8r" action="https://payl8r.com/process" method=”post”>
     <input type="hidden" name="data" value="<?php echo
     $base64_encrypted_json; ?>">
     <input type="hidden" name="rid" value="example">
     <input type="submit">
</form>

<iframe src="" name="payl8r"></iframe>
```

Note that the POST data sent to https://payl8r.com/process is named “data”. This should be the case in all requests to Payl8r.

As shown in the example above an additional parameter of “rid” can be passed containing your username. This is not required but is highly recommended as it reduces load time considerably.

5a. Return URLs

After a customer has completed an application, they are sent to one of 3 URLs specified in your request.

Abort (request_data: return_urls: abort**)**
If a user cancels during the Payl8r process, they will be sent to the abort URL.

Example message:
"You have chosen not to complete the process with Payl8r for finance on your basket. You can return to your cart by clicking the button below."

Fail (request_data: return_urls: fail**)**
If we do not think that the customer should be granted the finance due to affordability we will send them to the fail URL. A common fail URL is the stores cart page.

Example message:
"You were not eligible for finance with Payl8r. You can try one of our other checkout methods listed below."

Success (request_data: return_urls: success**)**
If a customer completes the process with Payl8r, they are sent to the success URL.

Example message:
"You have completed your finance request with Payl8r. Thank you for your purchase, an email will be sent to you to confirm your order."

There are many ways to process your return URL endpoints, and your implimentation will depend on how you want your system to work in these instances.

5b. Responses

When making a standard finance request a "return_data" URL is specified. Responses are sent to this URL as POST data once an application has finished processing. Responses come encrypted against a private key for your account and can be decrypted by your public key.

The encrypted data is also MIME base_64 encoded and will need to be decoded before you can decrypt the data. **It is best practice to only allow Payl8r's IP address to access this URL. A good method of implementation is to route all other users to your 404 page.**

An example of the decoded and decrypted JSON can be seen below:

```
{
      "return_data": {
            "order_id": "000012432",
            "application_id": 8293471,
            "status": "ACCEPTED",
            "customer_data": {
                  "user_id": 12432,
                  "firstnames": "Larry John",
                  "surname": "Ridley",
                  "email": "ljr@example.com",
                  "phone": "07769123567",
                  "address": "Flat 140, Mulan Court, Rider Street",
                  "city": "Manchester",
                  "postcode": "M51LN",
                  "delivery_name": "Sarah Hancourt",
                  "delivery_address": "15 Edward Street",
                  "delivery_city": "Manchester",
                  "delivery_postcode": "M24LS",
            }
      }
}
```

An example of the process of data verification and decoding in PHP:

```
if (isset($_POST['response'])) {
    if ($encrypted_response = base64_decode($_POST['response'])) {
        $key = 'file://keys/payl8r-crt.pem';
        if (openssl_public_decrypt($encrypted_response,$response,$key)) {
           if ($decoded_response = json_decode($response)) {
                // Verified data
                // e.g. $decoded_response['return_data']['order_id']
           }
        } else {
            // Route to 404 page
        }
    } else {
        // Route to 404 page
    }
} else {
    // Route to 404 page
}
```

Note that the POST data sent from payl8r is named “response”.

This page shows potential values for response data. The return data comes nested within a **"return_data"** object. This is shown in the JSON example on p10.

order_id
This is the ID that you sent us with the request. You should use this to match your order with the response.

application_id
This is the customer's Payl8r application ID. You can store this for later reference.

status
This states whether the application was accepted or declined.

reason
This is set if the "status" field is set to "DECLINED". It gives the reason for the refusal.

Potential "status" values:

Value	Description	Data type
ACCEPTED	The customer was accepted credit for this order. You can start processing.	String
DECLINED	The customer was refused credit. A reason will be given.	String
ABANDONED	The customer abandoned the process before finishing and 6 hours passed.	String

Potential "reason" values:

Value	Description	Data type
UNAFFORDABLE	The customer did not meet our affordability criteria.	String
POTENTIAL_FRAUD	The customer failed our fraud checks.	String

Additionally a "customer_data" object is returned within the "return_data" object. This should be used to update any values on your store that the customer may have updated during the order process with us. The example on page 11 shows all the data that can be returned in the "customer_data" object.

7. Testing

By setting test mode to 1 in a standard finance request, a different page will be returned than normal. When testing your account the normal process is disabled until testing has been completed, so you will need to start with test_mode set to 1.

The page that is returned in test mode will give you a breakdown of any issues with the request. It also gives a list of links that can be used to test responses. These buttons emulate an ACCEPTED, DECLINED, or ABANDONED response exactly as the response would come from the live version. This can be used to test that any relevant actions for these returns are working as you intend.

Once you have tested your integration and want to test the actual form, let us know and we will elevate your account to example mode. When your account is in example mode you can run through the normal process as if it were live, however no details are captured and credit checks are not ran. On completion an accepted response is always sent.

Additionally, when initialising a refund request in test mode, status will always be set to ACCEPTED as long as there are no formatting or account errors. NOT_FOUND errors are ignored and orders can not be refunded from test mode.

8. Locality

Important
At the moment, Payl8r is only available in the UK. We will not accept any applications from anywhere else. For customers outside of the UK you should not show the Payl8r checkout option or advertise that Payl8r is an option.

How you achive this will depend on your current systems. If you need advice, please don't hessitate to get in touch.

Author: Luke Roughley
Last updated: 24/10/2017