

DRAFT Specification for APIs for PFF new build

V 0.5 beta

Table of Contents

Standard Finance Request.....	3
Add clinic	4
Edit clinic.....	5
User login.....	6
Edit user details	7
When a user updates details on Payl8r	8
User order summary.....	9
Refund API	10
Clinic confirm booking.....	11
Order approve.....	12
Responses to unsuccessful requests.....	14

Standard Finance Request

request_type
N/A*

The response to the Standard Finance Request will be modified to include 'user_id' for the new user.

Example response

```
{
  "success": true,
  "return_data": {
    "order_id": "000012432",
    "application_id": "8293471",
    "status": "ACCEPTED",
    "customer_data": {
      "user_id": 27,
      "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"
    }
  }
}
```

Note: Users (customers) are not created for declined and abandoned finance requests, so the "user_id" element will be omitted from responses where "status" is "DECLINED" or "ABANDONED".

DECLINED requests will include a "reason" element in the return data*.

*Please refer to Payl8r Integration Guide version **1.4.0** for more information on how this request works

Note: The 'user_id' element in the response's return data is the Payl8r ID for the customer, and therefore needs to be stored in your customer's records as it is required in other messages relating to the customer, namely ['edit_user'](#), ['get_user_one_time_login_token'](#), ['get_user_order_summary'](#) and is also contained in the data provided [when a user updates details on Payl8r](#)

Add clinic

request_type:
add_retailer_client

request_data requirements:

1. client_name // string, the clinic's business name, up to 255 chars
2. email // string, up to 255 chars
3. sort_code // string of up to 10 chars, numbers only e.g. no spaces, no dashes
4. account_number // string of up to 20 chars, numbers only e.g. no spaces, no dashes
5. phone // up to 20 chars
6. address // up to 255 chars, line breaks indicated by '
'

Example request:

```
{
  "username": "example",
  "key": "dfga87ygd8oa32oh9a8whg",
  "request_data": {
    "request_type": "add_retailer_client",
    "client_name": "ACME, Inc.",
    "email": "jorn@acme.co.uk",
    "sort_code": "112233",
    "account_number": "12345678",
    "phone": "07123456789",
    "address": "1 Any Street<br>Anytown<br>Anyshire<br>AB1 2CD"
  }
}
```

The response

The 'retailer_client_id' element in the response's return data is the Payl8r ID for the clinic, and therefore needs to be stored in your clinic records as it is required for ['edit_retailer_client'](#) requests

Example response:

```
{
  "success": true,
  "return_data": {
    "retailer_client_id": 27
  }
}
```

Edit clinic

request_type
edit_retailer_client

request_data requirements:

1. retailer_client_id // integer
2. One or more of the following:
 - a. client_name // string, the clinic name
 - b. email // string, up to 255 chars
 - c. sort_code* // string, up to 10 chars, numbers only, no spaces or dashes
 - d. account_number* // string, up to 20 chars, numbers only, no spaces or dashes
 - e. phone // up to 20 chars
 - f. address // up to 255 chars, line breaks indicated by '
'

* Bank Details: If a sort code or account number is in the data, a sort code .

Example request:

```
{
  "username": "example",
  "key": "dfga87ygd8oa32oh9a8whg",
  "request_data": {
    "request_type": "edit_retailer_client",
    "retailer_client_id": 27,
    "client name": "ACME, Inc.",
    "email": "jorn@acme.co.uk",
    "sort_code": "112233",
    "account_number": "12345678",
    "phone": "07123456789",
    "address": "1 Any Street<br>Anytown<br>Anyshire<br>AB1 2CD",
    "error": ""
  }
}
```

Example response:

```
{
  "success": true,
}
```

User login

request_type
get_user_one_time_login_token

request_data requirements:

user_id // integer

Example request:

```
{
  "username": "example",
  "key": "dfga87ygd8oa32oh9a8whg",
  "request_data": {
    "request_type": "get_user_one_time_login_token",
    "user_id": 27
  }
}
```

Example response:

```
{
  "success": true,
  "return_data": {
    "one_time_login_token": "ONE_TIME_LOGIN_TOKEN"
  }
}
```

When appended to "<https://payl8r.com/>" the one_time_login_token forms a url which will log the user it was issued to into Payl8r.com without requiring their username or password. It works once only and expires after 24 hours.

Note: The one_time_login_token will usually be 30 characters, though reserve 128 characters for future use. It will only be valid for a maximum of 24 hours, and will only work once.

Edit user details

request_type

edit_user

request_data requirements

1. user_id // integer
2. One or more of the following:
 - a. firstname // string, 255 chars max
 - b. surname // string, 255 chars max
 - c. email // string, up to 255 chars
 - d. phone // string up to 20 chars
 - e. address // string, up to 255 chars, line breaks indicated by '
'
 - f. postcode // string, up to 24 chars
 - g. town // string, up to 255 chars

Example request:

```
{
  "username": "example",
  "key": "dfga87ygd8oa32oh9a8whg",
  "request_data": {
    "request_type": "edit_user",
    "user_id": 27
    "firstname": "John",
    "surname": "Smith",
    "email": "jorn@acme.co.uk",
    "phone": "07123456789",
    "address": "1, Any Street<br>Anytown<br>Anyshire",
    "postcode": "AB1 2CD",
    "town": "Anytown"
  }
}
```

Example response:

```
{
  "success": true
}
```

When a user updates details on Payl8r

If we are provided with a url for us to deliver user detail updates to your system, we will provide them as per the format below whenever a change to a user is made at Payl8r (excluding Bank Account details. Requests will be sent to the provided url immediately.

Example request:

```
{
  "key": "dfga87ygd8oa32oh9a8whg",
  "return_data": {
    "request_type": "edit_user",
    "user_id": 27
    "firstname": "John",
    "surname": "Smith",
    "email": "jorn@acme.co.uk",
    "phone": "07123456789",
    "address": "1, Any Street<br>Anytown<br>Anyshire",
    "postcode": "AB1 2CD",
    "town": "Anytown"
  }
}
```

Response

We do require a response, but nit a specific response. If a response other than the suggested response below is received we will assume your system has updated. If no response is received or the suggested response on unsuccessful update is received, we will attempt to resend the request within 24 hours.

Suggested response on successful update:

```
{
  "success": true
}
```

Suggested response on unsuccessful update:

```
{
  "success": false
}
```


User order summary

request_type
get_user_order_summary

request_data requirements:

1. user_id // integer
2. order_id // integer

Both these fields are provided in the initial "Standard Finance Request" response body (see above)

Example request:

```
{
  "username": "example",
  "key": "dfga87ygd8oa32oh9a8whg",
  "request_data": {
    "request_type": "get_user_order_summary",
    "order_id": 123456,
    "user_id": 27
  }
}
```

Example response:

```
{
  "success": true,
  "return_data": {
    "order_id": 123456,
    "user_id": 27,
    "status": "ACCEPTED",
    "total_paid": 1000.00,
    "total_payable": 1800.00,
    "next_payment_date": "DD\\MM\\YYYY",
    "next_payment_amount": 150.00,
    "final_payment_date": "DD\\MM\\YYYY"
  }
}
```

Refund API *CURRENTLY DISABLED*

request_type
refund_request

request_data requirements:

1. order_id // integer
2. refund_amount // float, 2 decimal places
3. currency // string, currently only "GBP" is supported

Example request:

```
{
  "username": "example",
  "key": "dfga87ygd8oa32oh9a8whg",
  "request_data": {
    "request_type": "refund request",
    "order_details": {
      "order_id": 000012432,
      "currency": "GBP",
      "refund_amount": 142.99
    }
  }
}
```

Example response:

```
{
  "success": true
}
```

Note: This request is also to be used for cancelling orders.

Clinic confirm booking

request_type
retailer_confirm_order

request_data requirements:

1. order_id // integer

Example request:

```
{
  "username": "example",
  "key": "dfga87ygd8oa32oh9a8whg",
  "request_data": {
    "request_type": "retailer_confirm_order",
    "order_id": 123456
  }
}
```

Example response:

```
{
  "success": true
}
```

Order approve

request_type
retailer_approve_order

request_data requirements:

1. order_id // integer
2. confirm_code // integer

Example request:

```
{
  "username": "example",
  "key": "dfga87ygd8oa32oh9a8whg",
  "request_data": {
    "request_type": "retailer_approve_order",
    "order_id": 123456,
    "confirm_code": 2755
  }
}
```

Example response:

```
{
  "success": true
}
```

Order update order total

request_type

retailer_set_order_amount

request_data requirements:

1. order_id // integer
2. confirm_code // float or string

Example request:

```
{
  "username": "example",
  "key": "dfga87ygd8oa32oh9a8whg",
  "request_data": {
    "request_type": "retailer_set_order_amount",
    "order_id": 123456,
    "new_total": 355.22
  }
}
```

Example response:

```
{
  "success": true
}
```

NOTE:

Only works for customers that were approved for additional amount in underwriting.

Responses to unsuccessful requests

Unless stated otherwise, unsuccessful messages will receive a response as follows:

```
{
  "success": false,
  "return_data": {
    "errors": {
      "Description of 1st error",
      "Description of 2nd error",
      "Description of 3rd error" ... etc etc
    }
  }
}
```