PUMAPAY MERCHANT BACKEND API GUIDE

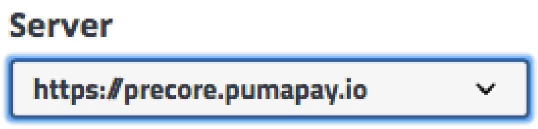
DOCUMENT V.1.2

PUMAPAY MERCHANT BACKEND API GUIDE

https://prembackend.pumapay.io/api/v2/doc/api/#

Important note #1: If you aren't familiar with Swagger functionality, please check our brief instructions here, it's easy to understand.

Important note #2: choose precore.pumapay.io in the list of servers.



Server

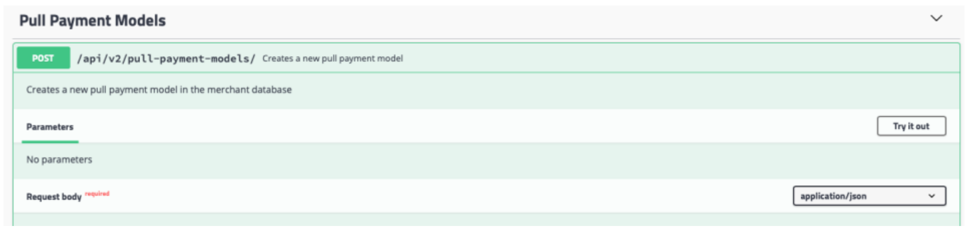
https://precore.pumapay.io

Billing models

Models define how your customers will be billed for using your products. It's like a billing/service plan. You can create one or multiple billing models.

To create a Billing model

Hit POST, then "Try it out" to start.



Pull Payment Models

POST /api/v2/pull-payment-models/ Creates a new pull payment model

Creates a new pull payment model in the merchant database

Parameters

No parameters

Request body required

application/json

Then edit the code in the Example Value field. Change values (text in quotes to the right) to set your Billing model preferences:

- Paste merchants id generated at the registration process.

```
"merchantID": "4a17335e-bf18-11e8-a355-000000fb1459",
```

- Write a title – that's how your Billing model will be called and that's the name what your customer sees.

```
"title": "National Cryptographic Gold Mebmership",
```

- Write a description of the model, telling the customer what he or she will get and how the payment for that is arranged.

```
"description": "Access to all gold articles",
```

- Specify the full sum to be paid by the customer.

```
"amount": 1099,
```

- Specify the first payment to be pulled from the customer's wallet immediately.

```
"initialPaymentAmount": 199,
```

- Specify the trial period in seconds. That's the time between the moment customer hits "Submit" and the moment the first recurring payment is pulled from his or her wallet. If there's no trial period, insert "0".

```
"trialPeriod": 86400,
```

- Specify the currency in which your product's price is evaluated: USD, EUR, GBP or JPY. Note that your customers are not obliged to pay in this currency. Customers will pay in PMA tokens.

```
"currency": "USD",
```

- Set a number of payments for a recurring Billing model. The frequency of recurring payments will be set later.

```
"numberOfPayments": 12,
```

- Choose a Billing model. For push payment: 1, for single pull payment: 2, for recurring pull payment: 3, fur recurring pull payment with initial payment: 4, for recurring pull payment with a trial period: 5.

```
"typeID": 2,
```

- Now set the frequency of recurring payments in seconds. For example, 604800 is a month.

```
"frequency": 604800,
```

- Set the network ID: 1 for the mainnet (to get real payments from customers) and 3 for the testnet (to experiment with your payment models).

```
"networkID": 3,
```

- Set "true" for automatedCashOut to transfer customers' payments to your Treasury automatically. Note that every transaction will cost you Ethereum gas. If you wish to transfer payments to Treasury manually, then set "false".

```
"automatedCashOut": true,
```

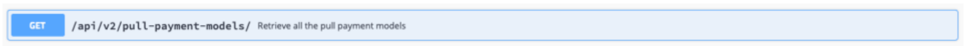
- If you had chosen "True" in the previous line, then set a frequency for transactions to your Treasury. If you set 1, then money will be transferred after every single payment. Set 2 or 3 or <...> to transfer money to the Treasury after every second, third, <...> payment.

```
"cashOutFrequency": 1
```

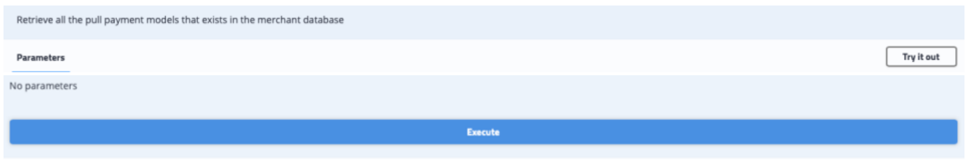
After you've created a Billing model, it goes to the database.

To view the models you’ve created

Hit GET /api/v2/pull-payment-models/


GET /api/v2/pull-payment-models/ Retrieve all the pull payment models

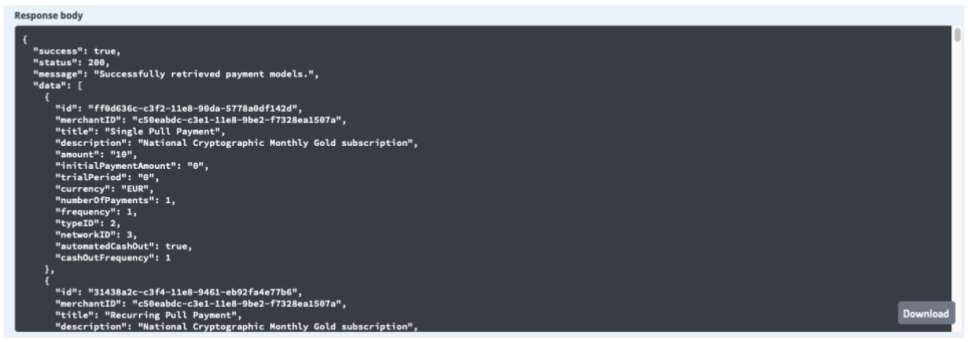
Then hit “Try it out” and “Execute”.


Retrieve all the pull payment models that exists in the merchant database

Parameters

No parameters

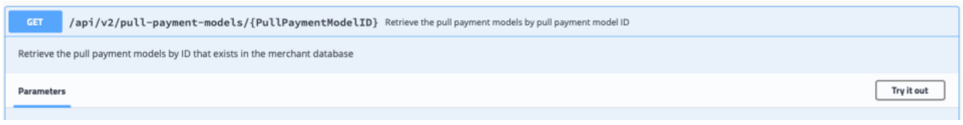
You’ll see a list of the models in the Response body


Response body

```
{
  "success": true,
  "status": 200,
  "message": "Successfully retrieved payment models.",
  "data": [
    {
      "id": "ff0d636c-c3f2-11e8-90da-5778a0df142d",
      "merchantID": "c50eabdc-c3e1-11e8-9be2-f7328ea1507a",
      "title": "Single Pull Payment",
      "description": "National Cryptographic Monthly Gold subscription",
      "amount": "10",
      "initialPaymentAmount": "0",
      "trialPeriod": "0",
      "currency": "EUR",
      "numberOfPayments": 1,
      "frequency": 1,
      "typeID": 2,
      "networkID": 3,
      "automatedCashOut": true,
      "cashOutFrequency": 1
    },
    {
      "id": "31438a2c-c3f4-11e8-9461-eb92fa4e77b6",
      "merchantID": "c50eabdc-c3e1-11e8-9be2-f7328ea1507a",
      "title": "Recurring Pull Payment",
      "description": "National Cryptographic Monthly Gold subscription",
```

To view a certain Billing model

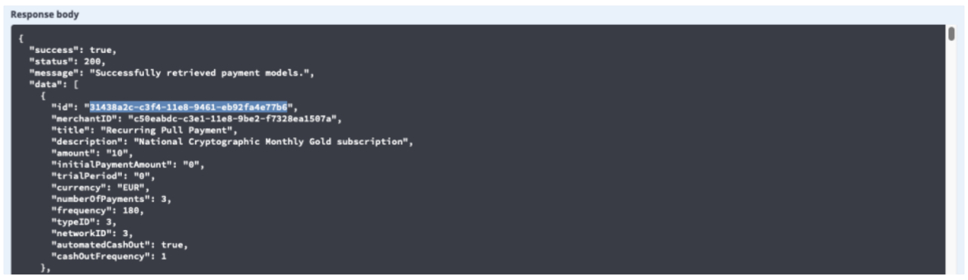
Hit GET /api/v2/pull-payment-models/{PullPaymentModelID} and “Try it out”:


GET /api/v2/pull-payment-models/{PullPaymentModelID} Retrieve the pull payment models by pull payment model ID

Retrieve the pull payment models by ID that exists in the merchant database

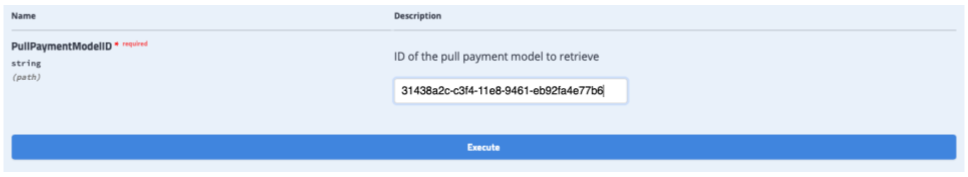
Parameters

Use GET /api/v2/pull-payment-models/ described above and copy ID of a Billing model you wish to view:


Response body

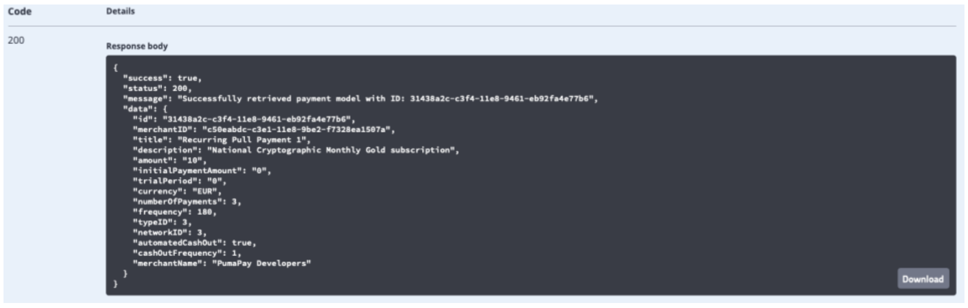
```
{
  "success": true,
  "status": 200,
  "message": "Successfully retrieved payment models.",
  "data": [
    {
      "id": "31438a2c-c3f4-11e8-9461-eb92fa4e77b6",
      "merchantID": "c50eabdc-c3e1-11e8-9be2-f7328ea1507a",
      "title": "Recurring Pull Payment",
      "description": "National Cryptographic Monthly Gold subscription",
      "amount": "10",
      "initialPaymentAmount": "0",
      "trialPeriod": "0",
      "currency": "EUR",
      "numberOfPayments": 3,
      "frequency": 180,
      "typeID": 3,
      "networkID": 3,
      "automatedCashOut": true,
      "cashOutFrequency": 1
    },
```

Paste it to the Billing model ID field and hit “Execute”:



Name	Description
PullPaymentModelID * required string *(path)*	ID of the pull payment model to retrieve 31438a2c-c3f4-11e8-9461-eb92fa4e77b6

View the details of the requested Billing model in the Response body field:

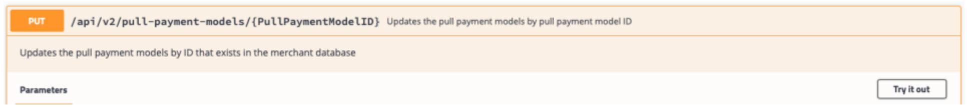


Code	Details
200	Response body

```
{
  "success": true,
  "status": 200,
  "message": "Successfully retrieved payment model with ID: 31438a2c-c3f4-11e8-9461-eb92fa4e77b6",
  "data": {
    "id": "31438a2c-c3f4-11e8-9461-eb92fa4e77b6",
    "merchantID": "c50eabdc-c3e1-11e8-9be2-f7328ea1507a",
    "title": "Recurring Pull Payment 1",
    "description": "National Cryptographic Monthly Gold subscription",
    "amount": "10",
    "initialPaymentAmount": "0",
    "trialPeriod": "0",
    "currency": "EUR",
    "numberOfPayments": 3,
    "frequency": 180,
    "typeID": 3,
    "networkID": 3,
    "automatedCashOut": true,
    "cashOutFrequency": 1,
    "merchantName": "PumaPay Developers"
  }
}
```

To edit a Billing model

Hit PUT /api/v2/pull-payment-models/{PullPaymentModelID} and “Try it out”:



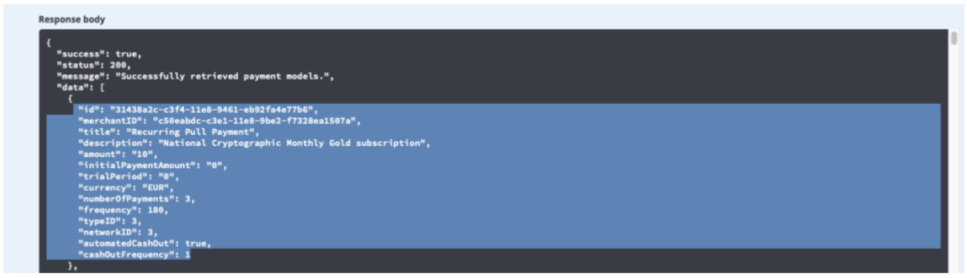
PUT /api/v2/pull-payment-models/{PullPaymentModelID} Updates the pull payment models by pull payment model ID

Updates the pull payment models by ID that exists in the merchant database

Parameters

Try it out

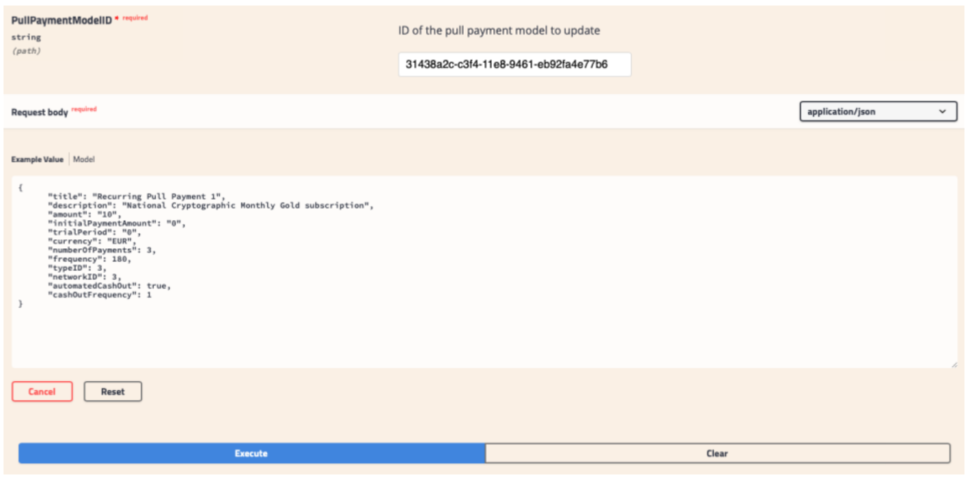
Then copy all the Billing model details you want to edit from the list using GET /api/v2/pull-payment-models/:



Response body

```
{
  "success": true,
  "status": 200,
  "message": "Successfully retrieved payment models.",
  "data": [
    {
      "id": "31438a2c-c3f4-11e8-9461-eb92fa4e77b6",
      "merchantID": "c50eabdc-c3e1-11e8-9be2-f7328ea1507a",
      "title": "Recurring Pull Payment",
      "description": "National Cryptographic Monthly Gold subscription",
      "amount": "10",
      "initialPaymentAmount": "0",
      "trialPeriod": "0",
      "currency": "EUR",
      "numberOfPayments": 3,
      "frequency": 180,
      "typeID": 3,
      "networkID": 3,
      "automatedCashOut": true,
      "cashOutFrequency": 1
    },
```

Paste it to the Example Value field, delete the Merchant’s ID and change something. For example, change the title adding “1”. Hit “Execute” to implement changes.



PullPaymentModelID * required
string
(path)

ID of the pull payment model to update

31438a2c-c3f4-11e8-9461-eb92fa4e77b6

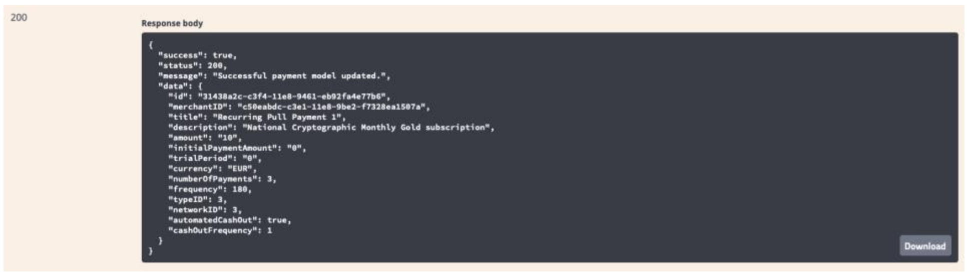
Request body required

application/json

Example Value | Model

```
{
        "title": "Recurring Pull Payment 1",
        "description": "National Cryptographic Monthly Gold subscription",
        "amount": "10",
        "initialPaymentAmount": "0",
        "trialPeriod": "0",
        "currency": "EUR",
        "numberOfPayments": 3,
        "frequency": 180,
        "typeID": 3,
        "networkID": 3,
        "automatedCashOut": true,
        "cashOutFrequency": 1
}
```

Check the Response body. You’ll see “Successful payment model updated” message and a new title.



200

Response body

```
{
  "success": true,
  "status": 200,
  "message": "Successful payment model updated.",
  "data": {
    "id": "31438a2c-c3f4-11e8-9461-eb92fa4e77b6",
    "merchantID": "c50eabdc-c3e1-11e8-9be2-f7328ea1507a",
    "title": "Recurring Pull Payment 1",
    "description": "National Cryptographic Monthly Gold subscription",
    "amount": "10",
    "initialPaymentAmount": "0",
    "trialPeriod": "0",
    "currency": "EUR",
    "numberOfPayments": 3,
    "frequency": 180,
    "typeID": 3,
    "networkID": 3,
    "automatedCashOut": true,
    "cashOutFrequency": 1
  }
}
```

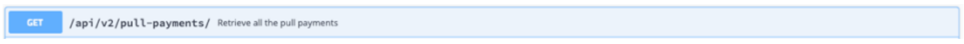
PullPayments

PullPayments are single transactions which are executed in Ehtereum blockchain, they belong to certain Billing models and made according to the rules you've set creating the latter. "Pull" stands for pulling funds from a customer's wallet after he or she agrees with the conditions. Each PullPayment is created automatically after the customer accepts a payment, has an Ethereum address and can be seen in the database.

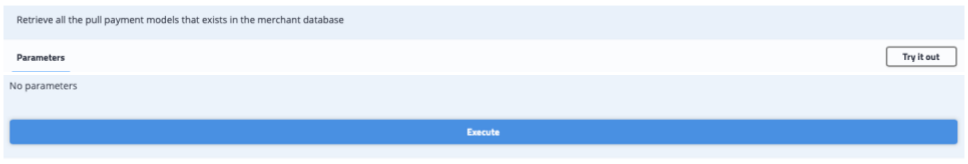
In Swagger you can view all PullPayments that were made within your Billing models.

To view the full list of PullPayments

Hit GET /api/v2/pull-payments/

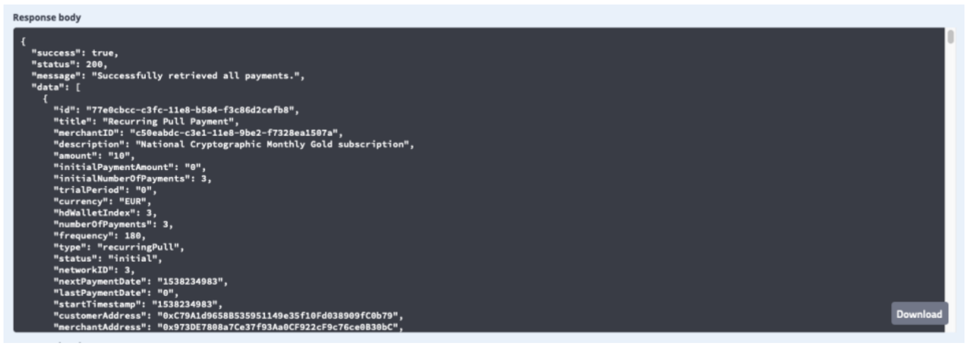


Then hit "Try it out" and "Execute".


Retrieve all the pull payment models that exists in the merchant database

No parameters

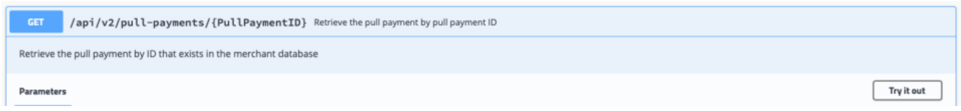
You'll see a list of the PullPayments in the Response body



```
{
  "success": true,
  "status": 200,
  "message": "Successfully retrieved all payments.",
  "data": [
    {
      "id": "77e0cbcc-c3fc-11e8-b584-f3c86d2cefb8",
      "title": "Recurring Pull Payment",
      "merchantID": "c50eabdc-c3e1-11e8-9be2-f7328ea1507a",
      "description": "National Cryptographic Monthly Gold subscription",
      "amount": "10",
      "initialPaymentAmount": "0",
      "initialNumberOfPayments": 3,
      "trialPeriod": "0",
      "currency": "EUR",
      "hdWalletIndex": 3,
      "numberOfPayments": 3,
      "frequency": 180,
      "type": "recurringPull",
      "status": "initial",
      "networkID": 3,
      "nextPaymentDate": "1538234983",
      "lastPaymentDate": "0",
      "startTimestamp": "1538234983",
      "customerAddress": "0xC79A1d9658B535951149e35f10Fd038909fC0b79",
      "merchantAddress": "0x973DE7808a7Ce37f93Aa0CF922cF9c76ce0B30bC",
```

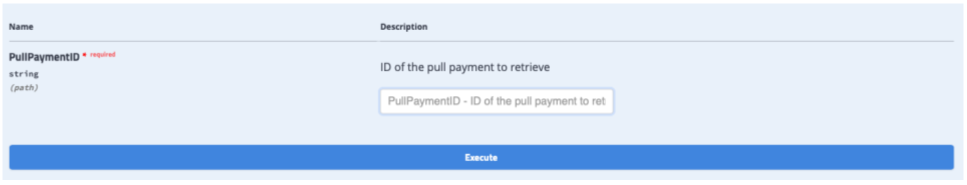
To view a certain PullPayment

Hit GET /api/v2/pull-payments/{PullPaymentID}. Then hit “Try it out”:


GET
/api/v2/pull-payments/{PullPaymentID}
Retrieve the pull payment by pull payment ID
Parameters
Try it out

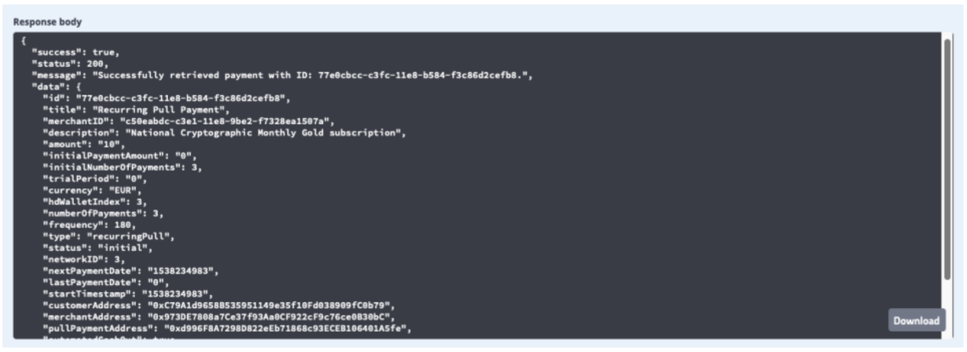
Retrieve the pull payment by ID that exists in the merchant database

Enter the ID of the PullPayment you wish to view and hit “Execute”:



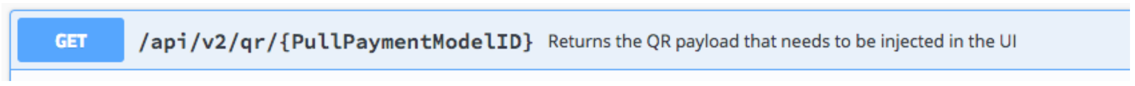
Name	Description
PullPaymentID * required string (path)	ID of the pull payment to retrieve PullPaymentID - ID of the pull payment to ret

You’ll see details of a certain PullPayment in the Response body:



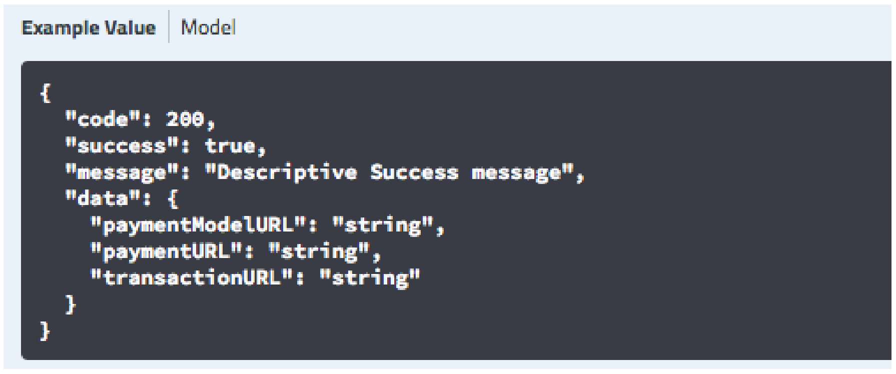
```
{
  "success": true,
  "status": 200,
  "message": "Successfully retrieved payment with ID: 77e0cbcc-c3fc-11e8-b584-f3c86d2cefb8.",
  "data": {
    "id": "77e0cbcc-c3fc-11e8-b584-f3c86d2cefb8",
    "title": "Recurring Pull Payment",
    "merchantID": "c50eabdc-c3e1-11e8-9be2-f7328ea1507a",
    "description": "National Cryptographic Monthly Gold subscription",
    "amount": "10",
    "initialPaymentAmount": "0",
    "initialNumberOfPayments": 3,
    "trialPeriod": "0",
    "currency": "EUR",
    "hdWalletIndex": 3,
    "numberOfPayments": 3,
    "frequency": 180,
    "type": "recurringPull",
    "status": "initial",
    "networkID": 3,
    "nextPaymentDate": "1538234983",
    "lastPaymentDate": "0",
    "startTimestamp": "1538234983",
    "customerAddress": "0xC79A1d9658B535951149e35f10Fd038909fC0b79",
    "merchantAddress": "0x973DE7808a7Ce37f93Aa0CF922cF9c76ce0B30bC",
    "pullPaymentAddress": "0xd996F8A7298D822eEb71868c93ECEB106401A5fe",
```

QR Payload


GET `/api/v2/qr/{PullPaymentModelID}` Returns the QR payload that needs to be injected in the UI

QR Payload provides merchants with necessary data bases on their billing model, which allows the merchants to easily generate the QR code for the clients, so they can make a single payment, or subscribe to the recurring payment.

Method: **GET**
Required Parameters: **PullPaymentModelID** -> the model created by the merchant and stored in the merchants database.



```
{
  "code": 200,
  "success": true,
  "message": "Descriptive Success message",
  "data": {
    "paymentModelURL": "string",
    "paymentURL": "string",
    "transactionURL": "string"
  }
}
```

Example of data that can be used by the merchants to generate the QR code (copied and pasted into the QR code generator, or used with the tools that merchant builds on their site)