



API Testing Documentation

Wayl makes it easy to get paid by connecting your business

Wayl is a payment facilitator that makes it easy for your customers to pay you for goods and services through many payment gateways.

Wayl API

Wayl has developed its own RESTful API to make it easy for merchants who build their own websites to integrate with the Wayl payment page.

المبلغ المطلوب

1,000

محتويات الطلب

1,000

Basket Value



خروج

التوصيل

m m

بغداد

العراق

Baghdad

طريقة الدفع









معلومات البطاقة

نحن نطبق اعلى معايير الامان لضمان سرية معلومات بطاقتك

رقم البطاقة

MM/\

CVC

ادفع

→ Start using Wayl API

Start Using Wayl API

Register and get your Merchant Token to start using Wayl API for seamless payments.

To use Wayl API for your app or a website and benefit from its various payment options, you first have to get a Merchant Token. If you don't have one already, please head to <https://wayl.io>, register your account, and contact us to get your Merchant Token.

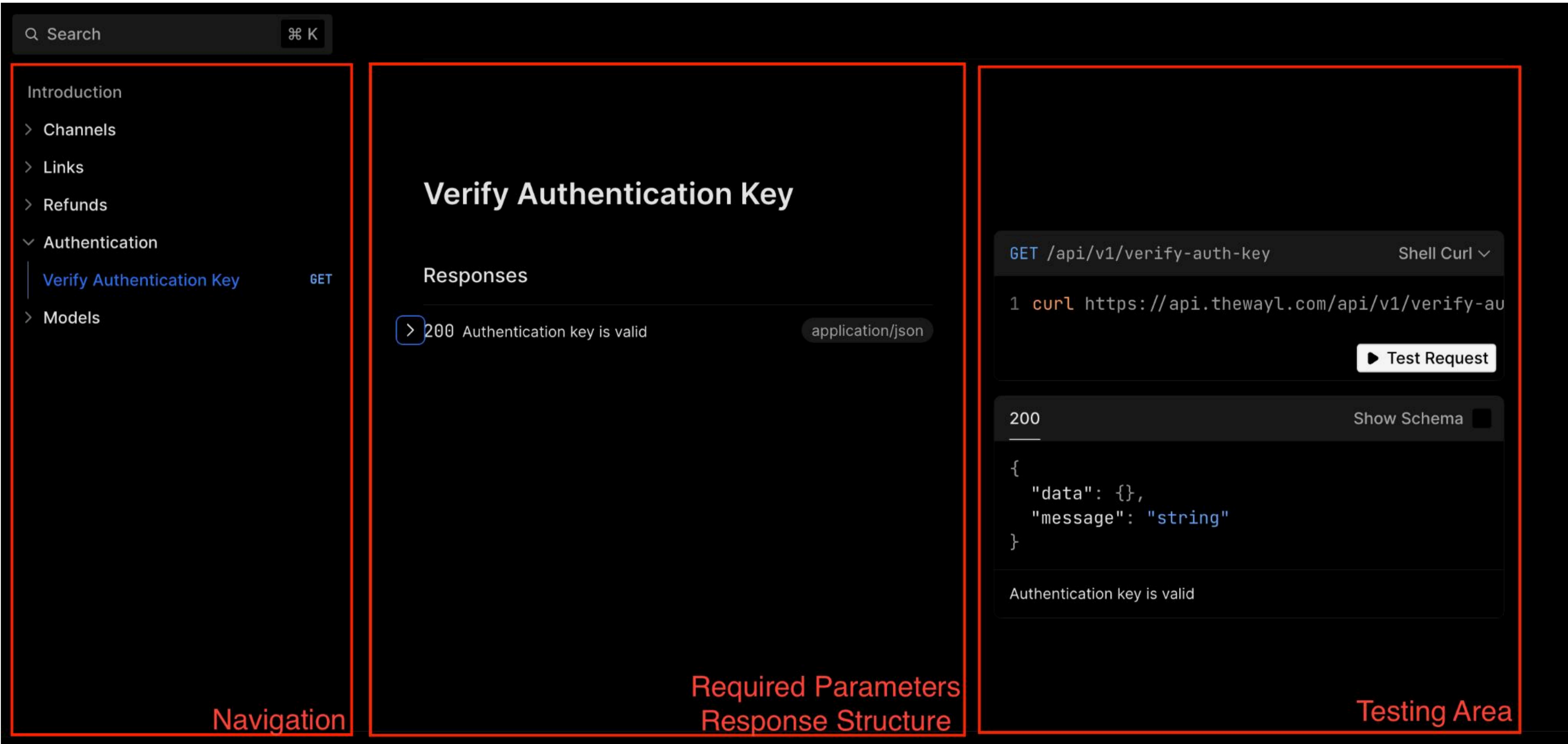
After you get your **Merchant Token**, you can either use it to directly interact with the Wayl API from your server, or you can use our API platform to test the API. For Demo purposes, we will be using the API Platform <https://api.thewayl.com>

1. Using Wayl API

Verifying your Merchant Token (see if it works)

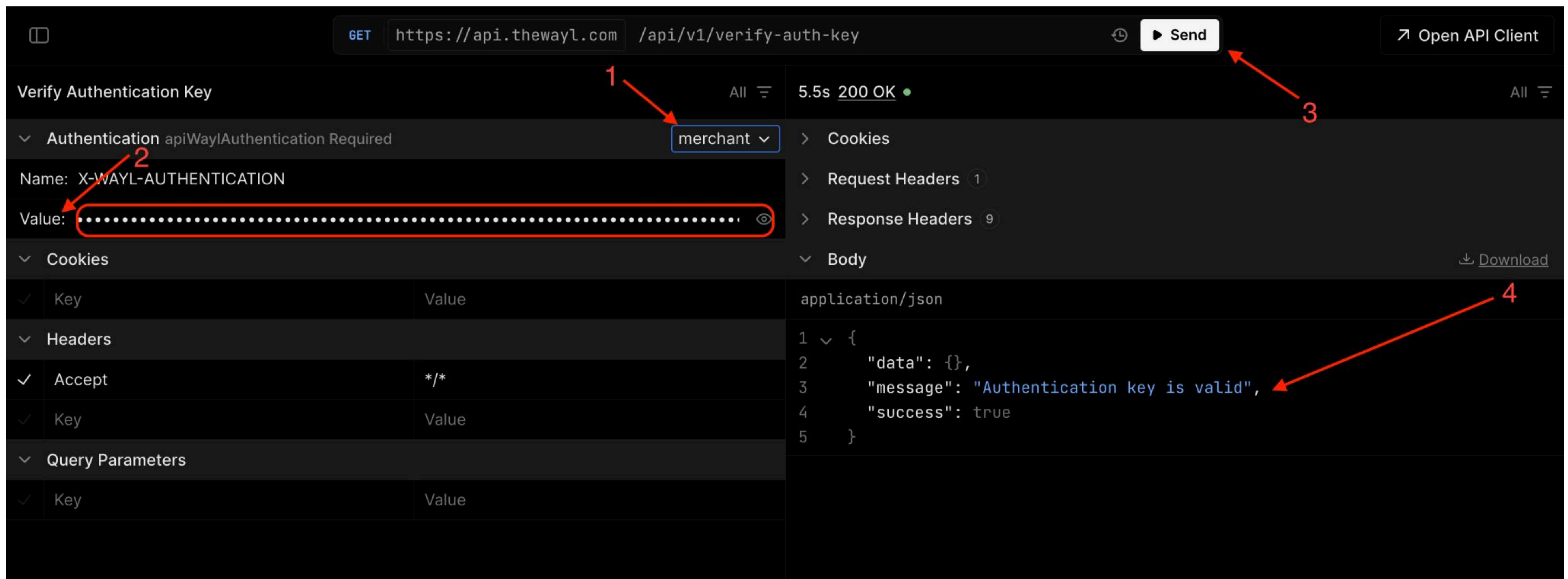
```
GET https://api.thewayl.com/api/v1/verify-auth-key
```

To verify that your merchant token is working properly, go to <https://api.thewayl.com>, then from the navigation menu press Authentication – Verify Authentication Key. This page will appear:



→ Using Wayl API

Now in **Testing Area**, Press **Test Request**, and the API testing screen will appear:



1. Select **Authentication** as **Merchant**
2. Enter your **Merchant Token** in **Value** of as **X-WAYL-AUTHENTICATION**
3. Press **Send**
4. You will get a response. If your merchant token is **valid**, the message will be: **"Authentication key is valid"**

Note: In each session, your token will be saved in the testing platform, so you can perform other requests without the need to reenter the token

Getting The Available Payment Methods

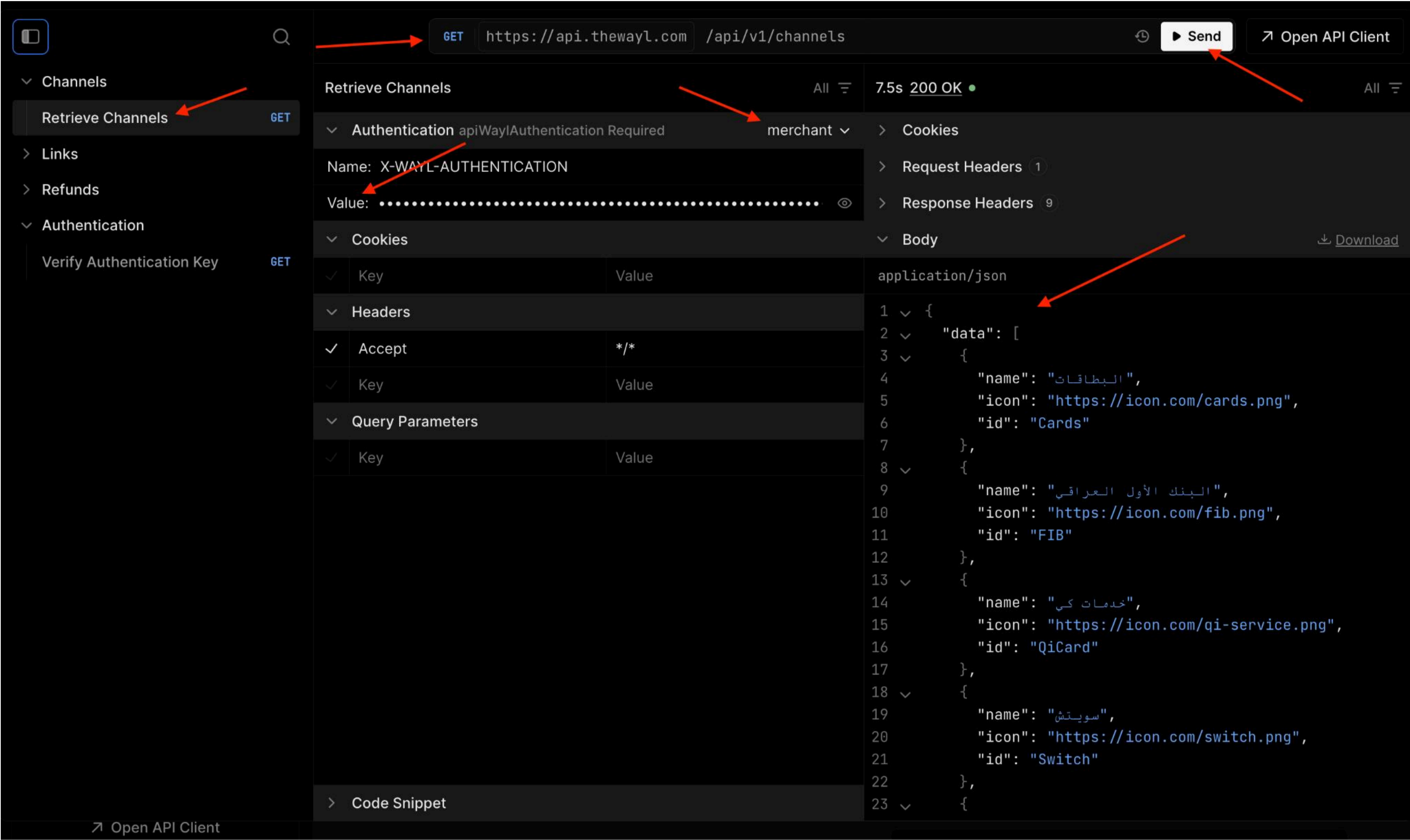
GET https://api.thewayl.com/api/v1/channels

📄

Sometimes, you want to display for your customer the available payment options on your app or a website.

Press Channels – Retrieve Channels – Test Request

Then send a GET request, you will receive a response as following:



What You Will Need The Most

Links are the order you will create on the Wayl platform and send it link to the customer to complete the payment

3.1. Create a Link

POST https://api.thewayl.com/api/v1/links



Creating A link (in addition to **Merchant Token**) requires some **req.body** parameters that you have to send to our server. The required parameter is explained in the image below:

Introduction

Channels

Retrieve ChannelsGET

Links

Create a LinkPOST

Retrieve LinksGET

Retrieve a specific LinkGET

Invalidate a specific LinkPOST

Refunds

Create a RefundPOST

Retrieve RefundsGET

Retrieve a specific RefundGET

Cancel a specific RefundDEL

Authentication

Verify Authentication KeyGET

Models

Create a Link

Body requiredapplication/json

referenceIdstringExample

a unique identifier for the link on your system.

totalnumberExample

The total amount to be paid, the total amount should equal the sum of all the line items.

currencystringExample

The currency of the link.

lineItemarray object[]

The line items to be paid for, you can add multiple line items to the link.

× Hide Child Attributes

labelstringExample

amountnumberExample

typestring · enum

increase or decrease the total amount to be paid.

increase

decrease

webhookUrlstringExample

The webhook url to be called when the link is paid.

webhookSecretstringExample

The webhook secret to be used to verify the webhook request.

redirectionUrlstringExample

The url to redirect the user to after the link is paid.

POST /api/v1/linksShell Curl ▾

1 curl https://api.thewayl.com/api/v1/links \

2 --request POST \

3 --header 'Content-Type: application/json'

4 --data '{

5 "referenceId": "dummy-reference-id",

6 "total": 10000,

7 "currency": "IQD",

8 "lineItem": [

9 {

10 "label": "Basket Value",

11 "amount": 10000,

12 "type": "increase"

13 }

14],

15 "webhookUrl": "https://webhook.site/#!/vie

16 "webhookSecret": "1234567890",

17 "redirectionUrl": "https://www.google.com"

▶ Test Request

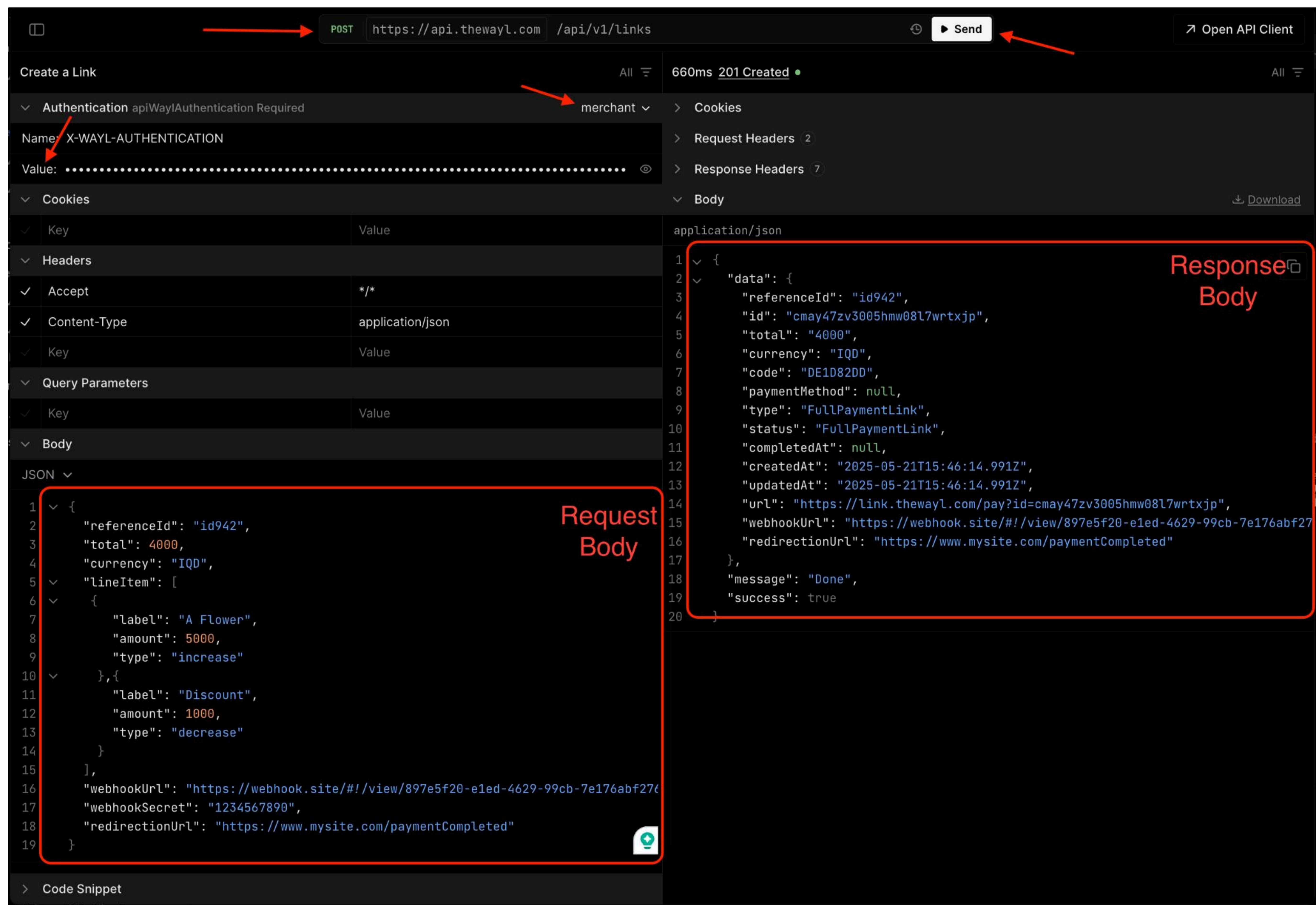
Press Test Request and send a POST request with the required Body.

Note:

- **referenceId**: should be unique in your database;
- **total**: should equal the summation of lineItem: increase/decrease;
- **webhookUrl**: a service endpoint that will receive the order and payment;
- **details**: when the payment is completed;
- **redirectionUrl**: a url that the customer will be redirected to when the payment is completed.

→ Links

After sending the request, you will get a response :



At this point, you have created a payment link that you should redirect your customer to.

"url": "https://link.thewayl.com/pay?id=cmay47zv3005hmw08l7wrtxjp",

After the customer completes the payment, he will be redirected to the earlier specified **redirectionUrl**, and the order and payment details will be sent to the earlier specified **webhookUrl**.

Note:

- **referenceId** is the Id in your database and should be unique;
- **id** is the order id in Wayl system;
- **code** is the order Code in Wayl System.

Below is what your customer will see when visiting your created link :
"url": "https://link.thewayl.com/pay?id=cmay47zv3005hmw08l7wrtxjp"

المبلغ المطلوب

4,000

محتويات الطلب

5,000	A Flower
1,000	Discount

خروج

التوصيل

لؤى محمد

العراق

بغداد

شارع الربيع

طريقة الدفع

معلومات البطاقة

نحن نطبق أعلى معايير الأمان لضمان سرية معلومات بطاقتك

رقم البطاقة

MM/\

CVC

ادفع

3.1.1. Link Webhook Url

Understanding webhooks is an important part of Wayl API integration. A webhook is a request that will be sent when a payment is completed to a service endpoint on your server.

This request will contain data about the order, payment method used, customer info, and address, and most importantly, a header signature to verify that this request has been sent from Wayl servers.

So, to verify that your customer has truly paid for your product, you have to create a service endpoint on your server that will receive the webhook request.

Note: for testing purposes, you can use <https://webhook.site>, which will give you an endpoint that will receive requests.

Below is an example of a received webhook request after a customer completed a payment:

Request Details & Headers

POSThttps://webhook.site/53a2472c-6a93-4e51-9601-b57

Host169.224.11.183

WhoisShodanNetifyCensysVirusTotal

Date05/21/2025 5:05:56 PM (2 hours ago)

Size482 bytes

Time0.000 sec

IDeef55d54-aeae-44f8-96da-f6d002c76650

NoteAdd Note

Query strings

None

hostwebhook.site

accept-encodinggzip, compress, deflate, br

content-length482

user-agentaxios/1.7.9

x-wayl-signature-2561b69bc002ac6cdf795757b78d8b754f2651cbbc616f6c92fffdc2a511e89e692

content-typetext/plain

acceptapplication/json, text/plain, */*

Form values

None

Webhook Signarure

Request Content

Raw Content

Format JSONWord-WrapCopy

Webhook Data

```
{
  "verb": "POST",
  "event": "order.created",
  "referenceId": "dassasasuaassasmmy-rze12faserasasenceas-id",
  "paymentMethod": "",
  "paymentStatus": "",
  "paymentProcessor": "",
  "total": 1000,
  "commission": 0,
  "code": "I94F590I",
  "customer": {
    "id": "cm8kktqzmz0000g00btwuo4ill",
    "name": "",
    "phone": "",
    "city": "iraq_al_basrah",
    "country": "IQ",
    "address": "Baghdad"
  },
  "items": [
    {
      "type": "increase",
      "label": "Basket Value",
      "amount": 1000
    }
  ],
  "id": "cmay0lfl2000ng0194x5r801o"
}
```

Webhook Signature

The webhook signature is a very important part of the Wayl API that you don't want to ignore; it verifies that the webhook request has been truly sent from Wayl servers after a customer completes a Payment.

Verifying the Signature on Your Server

On your webhook endpoint, you have to implement a signature verification function like this one :

Using Node-JS // Wayl Signature Verification

```
import crypto from "crypto";
function verifyWebhookSignature(data, signature, secret) {
  //data from webhook req.body
  //signature from req.headers "x-wayl-signature-256"
  //secret that was previously sent when creating the link, should be locally
  stored in your code or database
  //generate signature from data and secret
  const calculatedSignature = crypto.createHmac("sha256",
  secret).update(data).digest("hex");
  // Convert the signatures to buffers for security
  const signatureBuffer = Buffer.from(signature, "hex");
  const calculatedSignatureBuffer = Buffer.from(calculatedSignature, "hex");
  // Compare the created signature with the signature from the request
  if (signatureBuffer.length !== calculatedSignatureBuffer.length) {
    return false;
  }
  return crypto.timingSafeEqual(signatureBuffer, calculatedSignatureBuffer);
  // true if the signature is valid, false otherwise
}
verifyWebhookSignature(data, signature, secret)
```

Different Language
and Currency Display

Different Language and Currency Display

Link + &lang=ku¤cy=iqd
Link + &lang=en¤cy=usd
Link + &lang=ar¤cy=iqd

- 3.2 Retrieve Links
- 3.3 Retrieve a specific Link
- 3.4 Invalidate a specific Link