interweb

Generated on 13/11/2026 18:16

KNOWLEDGE BASE · PUBLIC DOCUMENT

Generating an API client for your language from our OpenAPI specification

PHP & Development

CODE	VERSION	COUNTRY · LANGUAGE
RLX-AO-KB-2738	v2026.11.13	Angola · English

Interweb Angola
RLX Angola, Lda. · Group RLX Holdings
Tax ID (NIF) 500 260 5724 · Commercial registration no. 26464-25/250724

https://www.meu.interweb.ao/knowledgebase/2738

Generating an API client for your language from our OpenAPI specification

PHP & Development Created on **November 13, 2026** 2 min read



The short answer: a client generator reads our OpenAPI specification and writes the code that calls the API in your language, with one function per operation. It saves work when you use many operations. The best known is **OpenAPI Generator**, which produces clients for PHP, Python, TypeScript, Java, C#, Go and many more.

With OpenAPI Generator

1. **Download the specification** into your project folder, named `openapi.json`.
2. **Get the generator ready:** the npm command-line version needs Java installed; alternatively, it runs in Docker with nothing else to install.
3. **Generate the client** in the language you want, with one of these commands:
4. **Configure the generated client** with the API address and the Bearer key, read from the `RLX_API_KEY` variable, as in the documentation examples.

```
npx @openapitools/openapi-generator-cli generate -i openapi.json -g python -o client-python
npx @openapitools/openapi-generator-cli generate -i openapi.json -g php -o client-php
docker run --rm -v "${PWD}:/local" openapitools/openapi-generator-cli generate -i /local/openapi.json -g typescript-fetch -o /local/client-ts
```

Language	Value for `-g`
PHP	`php`

Language	Value for `-g`
Python	`python`
TypeScript or JavaScript	`typescript-fetch` or `javascript`
Java	`java`
C#	`csharp`
Go	`go`

Before you trust the generated code

The generator knows the shape of the API, not its rules.
It does not know it should send `Idempotency-Key` on purchases and payments, wait for `Retry-After` after a `429`, or that lists come in pages. That is in the **API rules** and you need to add it.

1. **Use a recent version of the generator.** Support for OpenAPI 3.1 is newer than for 3.0 and improves from release to release.

2. **Test everything with a `sandbox_` key** before moving to `prod_`.

3. **Generate again when the specification changes**, and do not edit the generated files by hand: your changes are lost on the next run.

Generate or write by hand?

Situation	What pays off
You use two or three operations	Write it by hand, from the documentation examples: less code, fewer dependencies.
You use many operations, across several areas	Generate the client and add your own layer on top with the API rules.
A team is working on the integration	Generate it, so everyone uses the same names and types.

Just want to try things without code?
The Playground and API clients make the same requests: see **Postman**.

RECOMMENDED

Start by downloading the specification.

Download the specification →

SEE ALSO

- The OpenAPI specification of our API: what it is, where to download it and what it contains →
- Code examples in our API docs: curl, PHP, Python and JavaScript, ready to copy and adapt →
- Importing our API into Postman from the OpenAPI specification, step by step →
- Importing our API into Insomnia and other API clients, and setting the Bearer key →
- Calling a web API from PHP with cURL and reading the JSON reply →

interweb

Interweb Angola

RLX Angola, Lda. · Group RLX Holdings

TAX ID (NIF)	500 260 5724
COMMERCIAL REGISTRATION NO.	26464-25/250724
REGISTERED OFFICE	Rua Direita do Patriota, Bairro Dangereux Belas, Luanda, Angola
CONTACT	info@interweb.ao · +244 942 011 191 · www.interweb.ao
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-AO-KB-2738 · v2026.11.13 · Generated on 13/11/2026 18:16