interweb

Generated on 18/10/2026 12:13

KNOWLEDGE BASE · PUBLIC DOCUMENT

SMPP or an HTTP SMS API: what each one is and which your application needs

WhatsApp & APIs

CODE	VERSION	COUNTRY · LANGUAGE
RLX-MZ-KB-2152	v2026.10.16	Mozambique · English

Interweb Mozambique
RLX Holdings, S.A. · Group RLX Holdings
Tax ID (NUIT) 401 191 607 · Legal entity no. (NUEL) 105 045 974 · Economic activity code (CAE) 62022

https://www.meu.interweb.co.mz/knowledgebase/2152

The short answer (for developers): there are two usual ways for an application to hand SMS to a provider. With an **HTTP API**, your code makes one web request per send, with an API key, and gets back a message ID; the status arrives later by **callback** (a webhook to a URL of yours) or by asking. With **SMPP** (Short Message Peer-to-Peer), your program opens a TCP connection that stays open, logs in once (the **bind**) and sends and receives binary packets over that same connection: it is the protocol SMS centres (SMSC) and aggregators speak to each other. For most websites the HTTP API is enough. SMPP pays off when the provider offers it and the application sends all the time, needs replies on the same channel, or wants to switch providers without rewriting the integration.

The differences that matter

Aspect	HTTP API	SMPP
How it connects	One HTTPS request per send; nothing stays open in between.	A TCP connection that stays open for hours; you log in once (bind).
Where it can run	Anywhere that can make outgoing HTTPS requests: shared hosting, a VPS, cloud functions.	In a program that is always running: a VPS or server, as a service.
Format	JSON or form fields, different for every provider.	Standard packets (PDUs): submit_sm, deliver_sm and others. Version 3.4 is the most widespread.
Delivery reports	A callback to a URL of yours, or a provider endpoint you query.	A deliver_sm on the same connection, with the receipt inside.
Replies from recipients	A webhook to your URL, if the provider supports it.	A deliver_sm on a receiving bind (receiver or transceiver).
Switching provider	Rewrite the integration.	Mostly change host, port, credentials and a few parameters.
Effort	Low: cURL and a few lines.	Higher: connection state, windows, reconnects, encodings.

How to choose and where to start

1. **Ask the provider first what it offers.** Many only have an HTTP API; others have both. An SMPP account usually needs the provider to allow your server IP and, sometimes, a separate agreement.

2. **If you send from a website** (sign-up codes, order notices, reminders), pick the HTTP API: from PHP it is a cURL request, as in **calling a web API with cURL**.

3. **Pick SMPP** if you send continuously and the provider recommends it, if you want replies and receipts without exposing a public URL, or if you want to move between several providers that speak SMPP.

4. **For SMPP, get a VPS ready** with the client running as a service that restarts by itself, and handle reconnects: **SMPP binds and enquire_link**. Help choosing: **which plan for an application or an API**.

5. **Whichever route,** keep the credentials out of the code (**environment variables and secrets**) and store the ID the provider returns next to your own record, so delivery reports can be matched: **delivery reports (DLR)**.

6. **Test with your own number first,** with and without accents in the text, before sending to customers: **GSM-7 or Unicode**.

Never call the SMS provider from the browser (JavaScript). The API key would be visible to every visitor, and anyone could send SMS at your expense. Sending always starts on your server.

An open-source gateway can sit in the middle. Programs such as Kannel or Jasmin hold the SMPP connection to the provider and give your application a simple HTTP interface. Useful when you want SMPP without writing the protocol handling, but it is one more service to look after on the VPS.

Need help preparing the VPS that will hold the connection to your SMS provider? The SMS account is with the provider; the server we can look at with you.

Open a support ticket

RLX-LIMITES

In our plans

An SMPP client is a program that stays running and keeps one connection open, so it needs a server of its own, not shared hosting. Our VPS plans run on KVM with full SSH access, so you can install the client and keep it running as a service. A client that only holds a connection and sends messages is light, so VPS1 (1 vCPU, 2 GB RAM, 40 GB NVMe) is a reasonable start; size up if your volume or the rest of the application needs it, up to VPS6 (32 vCPU, 64 GB).

We do not state here whether a given outbound port is open. Test it from the VPS: try to connect to the provider's SMPP host and port (for example with nc -vz). If it does not answer and the provider says it should, open a ticket with the host, the port and the result.

/RLX-LIMITES

SEE ALSO

SMPP binds and enquire_link: keeping an SMS connection alive on a VPS

SMS delivery reports (DLR): how your application knows whether a message arrived

Calling a web API from PHP with cURL and reading the JSON reply

Which plan for a web application or an API: shared hosting, VPS or Cloud

WhatsApp, e-mail or SMS: advantages, disadvantages, risks and who each is for

Webhook or polling: how one system learns that something changed in another

interweb

Online version of this document

Interweb is a leading internet services company, with a mission to help businesses to be born and prosper through technology.

2020	37,000+	130+
FOUNDED	CLIENTS	SIGN-UPS A DAY

Interweb Mozambique

RLX Holdings, S.A. · Group RLX Holdings

TAX ID (NUIT)	401 191 607
LEGAL ENTITY NO. (NUEL)	105 045 974
ECONOMIC ACTIVITY CODE (CAE)	62022
REGISTERED OFFICE	Bairro 7 de Setembro Vilankulo 1304 · CEP 0411-02, Inhambane, Mozambique
CONTACT	info@interweb.co.mz · +258 84 635 1111 · +258 86 635 1111 · www.interweb.co.mz
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

RLX-MZ-KB-2152 · v2026.10.16 · Generated on 18/10/2026 12:13