WISE NETWORK DOCUMENTATION

Messaging integration guidance

A starting point for connecting a messaging system: what Wise Network offers today, the main integration methods, and how a connection project usually runs.

> **Educational guide.** General information about messaging integrations. It does not describe a live Wise Network API, and examples are illustrative only.

Online version: https://wisenetwork.co/support/api/ · Content updated 28 October 2026

What is available today

Wholesale SMS connections are arranged individually with our team during onboarding. The technical approach for traffic exchange is described on SMPP connectivity.

Wise Network does **not** currently operate a public HTTP messaging API. There are no Wise Network API endpoints, API keys, code samples or webhooks to use, and this guide does not describe any. When this guide shows an example, it is labelled as illustrative.

What is and is not available

Item	Status
Discussing an SMS or voice interconnection with our team	**Available now** through the request form
Technical support for existing partners	**Available now** at support@wisenetwork.co
Public HTTP messaging API, SDKs and code samples	**Future service**
Delivery-report webhooks, balance and coverage endpoints	**Future service**

The two common integration methods

Most messaging systems connect to a provider in one of two ways.

SMPP

SMPP (Short Message Peer-to-Peer) is a binary protocol over TCP/IP. The client, called an ESME (external short message entity), opens a long-lived session called a bind and keeps it open. Messages are submitted with `submit_sm`, and delivery receipts come back on the same or a separate session as `deliver_sm`. SMPP suits high, continuous volumes. Read What is SMPP? for the full picture.

HTTP APIs

An HTTP API accepts one request per message or batch, usually over HTTPS with JSON or XML data. It is simpler to start with, works through ordinary web infrastructure and suits lower or irregular volumes. Delivery results are fetched by a status request or pushed to the client through a webhook. The messaging API concepts guide explains these patterns.

SMPP and HTTP compared

Question	SMPP	HTTP API
Connection	Persistent TCP session (bind)	Separate HTTPS request for each call
Typical volume	High and continuous	Low to medium, or irregular
Delivery receipts	Returned on the session as deliver_sm	Status request or webhook
Skills needed	SMPP client library and session handling	Any HTTP client
Typical access controls	System ID, password and source IP allow-list	Key or token, HTTPS, optional IP allow-list

How an integration project usually runs

1. **Scope.** Agree destinations, traffic type (for example one-time passcodes or notifications), expected volume and the connection method.
2. **Commercial and compliance checks.** Agree terms, complete company checks and confirm sender ID rules for each destination.
3. **Technical exchange.** Exchange connection details and the public IP addresses you will connect from, through a secure channel.
4. **Testing.** Send agreed test messages and check encoding, sender ID, delivery receipts and error handling.
5. **Go-live and monitoring.** Start live traffic with agreed limits, and keep contacts for incidents and changes.

The integration checklist lists everything to prepare for each step.

Next steps

- Ready to discuss a connection? Request an SMS interconnection.
- Already connected and need help? Email support@wisenetwork.co.

Questions? Email support@wisenetwork.co. This PDF is generated from the web page above; the online version is the reference.