MADDY DEVELOPERS

WhatsApp Service

Integration Guide for Partner Projects

Support
maddydevelops6@gmail.com
WhatsApp +92 302 6164892
md.wepure.site

1. What this service does

This service lets your project send WhatsApp messages from your own phone numbers. You link a number once by scanning a QR code, then send messages through a simple HTTP API. OTPs, alerts, notifications and replies all work the same way.

Your numbers, messages and delivery logs are private to your account and reachable only with your own token.

What you have been given

- **Username** — identifies your project.
- **API token** — your password. Shown once when issued, and never recoverable. If you lose it, ask us to rotate a new one.
- **Service URL** — `https://whatsapp-service.wepure.site`

The same username and token sign you into the dashboard in a browser and authenticate your API calls. One credential, one place to change it.

Single number or many

Your account is set to one of two modes. Check which one you are on by calling `/api/v1/me`.

Mode	What it means
SINGLE	You may link one WhatsApp number. Scan the QR on your dashboard and you are live.
MULTI	You may link several numbers, and create new ones straight from your own software using the API in section 4.

Need the limit raised? Contact us using the details on the last page.

2. Authentication

Send your token on every request using the `X-API-Token` header.

```
curl https://whatsapp-service.wepure.site/api/v1/me \
  -H "X-API-Token: YOUR_API_TOKEN"
```

`Authorization: Bearer <token>` works too if that suits your HTTP client better.

Always start here

Make this your first call when wiring up the integration. It proves the token works and tells you your mode and current usage, so you never have to guess.

GET `/api/v1/me`

Confirm credentials and read your allowance.

```
{
  "account":  { "username": "your-project" },
  "sessions": { "total": 2, "connected": 2 },
  "allowance":{ "mode": "MULTI", "used": 2,
                "limit": 5, "can_create": true }
}
```

Read `allowance.can_create` before offering an "add another number" button in your own UI, instead of finding out from an error.

3. Quick start

Five steps from nothing to a delivered message.

Step	Call	What happens
1	GET /api/v1/me	Confirm your token works
2	POST /api/v1/sessions	Create a number slot, get a session_id
3	GET /api/v1/sessions/{id}/qr	Get a QR image, show it to your user
4	(your user scans it)	The phone links to the service
5	POST /api/v1/messages/send	Send from that number

A SINGLE-mode project can skip steps 2 and 3 by scanning the QR on the dashboard instead, then go straight to step 5.

4. Connecting a WhatsApp number

A **session** is one linked WhatsApp number. Create it, show its QR to whoever owns the phone, and once they scan it you can send.

POST `/api/v1/sessions`

Create a new number slot. Nothing is started until you ask for the QR.

```
curl -X POST https://whatsapp-service.wepure.site/api/v1/sessions \
  -H "X-API-Token: YOUR_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{"label":"Sales line"}'

# -> { "session": { "id": "7c9e...", "state": "PENDING" } }
```

Keep that `id` — every later call uses it.

GET `/api/v1/sessions/{id}/qr`

Fetch the current QR code. Starts the session automatically on first call.

```
curl https://whatsapp-service.wepure.site/api/v1/sessions/SESSION_ID/qr \
  -H "X-API-Token: YOUR_TOKEN"

# -> { "qr": "...",
#      "qr_expires_in_sec": 14,
#      "session": { "state": "SCAN_QR" } }
```

Three rules for the QR

- **Poll every 2 seconds.** WhatsApp rotates the code about every 20 seconds.
- **Always render the newest image.** Showing a cached one makes the phone say "Unable to link device".
- **Stop when state becomes WORKING.** That means the number is linked and ready.

The `qr` value is a PNG data URL. In a web page you can use it directly: `<img src={qr} />`

Session states

State	Meaning	What to do
PENDING	Created, not started	Call the QR endpoint
STARTING	Coming up	Keep polling
SCAN_QR	Waiting for a scan	Show the QR to your user
WORKING	Linked and ready	You can send
STOPPED	Stopped	Call the QR endpoint to restart
FAILED	QRs went unscanned	Call the QR endpoint for a fresh code
LOGGED_OUT	Unlinked from the phone	Scan again to reconnect

GET `/api/v1/sessions`

List your numbers and see which are connected.

```
# -> { "count": 3, "connected": 2, "data": [ ... ] }
```

One call regardless of how many numbers you have. Good for a status panel in your own admin area.

POST `/api/v1/sessions/{id}/logout`

Unlink the phone. The slot stays, so the same id can be linked again later.

DELETE `/api/v1/sessions/{id}`

Remove the number completely.

5. Sending a message

POST `/api/v1/messages/send`

Send a text message from one of your connected numbers.

```
curl -X POST https://whatsapp-service.wepure.site/api/v1/messages/send \
  -H "X-API-Token: YOUR_TOKEN" \
  -H "Idempotency-Key: otp-user-8821" \
  -H "Content-Type: application/json" \
  -d '{
        "session_id": "7c9e...",
        "to": "03001234567",
        "message": "Your code is 418290"
      }'

# -> { "to": "923001234567", "message_id": "3EB0...",
#      "status": "SENT",
#      "usage": { "minute_used": 3, "minute_limit": 20,
#                 "day_used": 47, "day_limit": 215 } }
```

Always send an Idempotency-Key for OTPs

If your request times out and you retry, the same key returns the original result instead of delivering a second code. Without it, a retry sends the message twice. Use anything unique to the attempt, such as `otp-<user-id>-<timestamp>`.

Phone number format

Numbers are cleaned up for you: everything that is not a digit is stripped, and a leading 0 is replaced with the country code (92 by default). So `0300-1234567`, `03001234567` and `+923001234567` all reach the same phone.

GET `/api/v1/messages`

Your message history, newest first.

```
GET /api/v1/messages?session_id=...&status=SENT&limit=25&page=1
```

Message status moves PENDING to SENT to DELIVERED to READ as receipts arrive from WhatsApp, or FAILED if it could not be delivered.

6. Handling errors

Every error returns JSON with a human-readable `message`. Handle these four and you have covered almost everything.

Code	Meaning	What to do
400	Bad input, or the number is not on WhatsApp	Fix the request. Do not retry unchanged.
401	Token missing, wrong, or revoked	Check the header. Ask us to rotate if needed.
403	This action is not permitted for your account	Read the message; it names the reason.
404	No session with that id	Check the session_id.
409	The number is not connected right now	Fetch the QR and have it scanned again.
429	Rate limit reached	Back off. Honour the Retry-After header.

A 409 comes back immediately rather than waiting, so your request never hangs because someone has not scanned a code yet.

7. Keeping your number safe

WhatsApp can ban numbers that behave like spam. The service enforces sending limits and warms up new numbers automatically, but how you use it matters more than anything we can control.

What we do for you

- **Rate limits** — a cap per minute and per day on every number.
- **Warm-up** — a new number starts at a low daily cap and rises over about two weeks. This is on by default.
- **Idempotency** — retries never double-send.

What you must do

- Only message people who contacted you first or signed up for it.
- Vary your message text. Identical messages to many numbers is the clearest spam signal there is.
- Never bulk-send from a number you linked today.
- Watch for recipients blocking you, and stop if it climbs.
- Keep a spare number linked so a problem is not an outage.

A banned number cannot usually be recovered. If delivery absolutely cannot fail, talk to us about using WhatsApp's official API for those messages instead.

8. Support

We are happy to help with integration, limits or anything else.

Email	maddydevelops6@gmail.com
WhatsApp	+92 302 6164892
Website	md.wepure.site
Service URL	`https://whatsapp-service.wepure.site`
Dashboard	`https://whatsapp-service.wepure.site/login`

For your engineers and AI assistants

Two machine-readable versions of this guide are available. Point your developer or coding assistant at either and it can wire up the integration without reading this document.

File	Use it for
`https://whatsapp-service.wepure.site/docs/openapi.json`	OpenAPI 3.1 spec. Import into Postman or Insomnia, or generate a typed client.
`https://whatsapp-service.wepure.site/docs/integration.md`	The whole guide in Markdown, written for an AI assistant to read.

Ask us to rotate your token the moment you suspect it has leaked. Rotation takes effect on the very next request.