

vCX WhatsApp Onboarding & Messaging

Base URL

```
https://backend.admin.versalence.online/api/v2
```

All protected endpoints require:

```
Authorization: Bearer <JWT_TOKEN>
```

1. Authenticate the customer admin

The customer's admin user logs in with their vCX credentials. The returned JWT is used for all subsequent calls.

```
POST /api/v2/login
Content-Type: application/json

{
  "email": "admin@customer.com",
  "password": "CustomerAdminPassword"
}
```

Response:

```
{
  "success": true,
```

```
"message" "Sign-in successful",
"token": "<JWT_TOKEN>"
}
```

Notes:

- The user must have `email_verified = 'yes'`.
- The account must be `active`.
- The JWT expiry is controlled by the server (`JWT_EXPIRES_IN`).

2. Check WhatsApp integration status

Use this to decide whether the customer still needs to onboardApp.

```
GET https://backend.versalence.online/api/v2/getWaba
Authorization: Bearer <JWT_TOKEN>
```

Not integrated response:

```
{
  "success": true,
  "message": "no whatsapp data found"
}
```

Already integrated response:

```
{
  "success": true,
  "message": "whatsapp data found",
  "data": [
    {
      "app_id": "...",
      "waba_id": "...",
      "access_token": "[hidden]",
      "phone_id": "...",
      "phone_number": "919876543210",
    }
  ]
}
```

```
    "display_phone_number": "+91 98765 43210",
    "onboarding_mode": "standard",
    "is_active": true,
    "webhook_status": "subscribed"
  }
]
}
```

3. Check Coexistence status (optional)

If the partner wants the customer to use **Coexistence** mode (so the customer's existing WhatsApp Business app keeps working alongside vCX), check the feature flag first.

```
GET /api/v2/whatsapp-coexistence/status
Authorization: Bearer <JWT>
```

Response:

```
{
  "success": true,
  "data": {
    "uuid": "...",
    "coexistence_enabled": false,
    "onboarding_mode": "standard",
    "is_active": false,
    "waba_id": null    "phone_id": null
  }
}
```

Role requirement: Admin only.

4. Enable Coexistence mode

If the customer wants Coexistence onboarding, enable the feature flag **before** running embedded signup.

```
/api/v2/whatsapp-coexistence/enable
Authorization: Bearer <JWT_TOKEN>
Content-Type: application/json

{
  "enabled": true
}
```

Response:

```
{
  "success": true,
  "message": "Coexist enabled for tenant"
}
```

Role requirement: Admin only.

5. Launch Meta Embedded Signup from the mobile app

The mobile app must open Meta's Embedded Signup flow using the **vCX Meta App ID**. The recommended implementation is- Open a WebView / in-app browser.

- Load the Meta Embedded Signup SDK configured with vCX's `APP_ID`.
- Request the `whatsapp_business_management` permission.
- Ask the user to select or create a WhatsApp Business Account and phone number.
- Meta will redirect back the app with an **authorization** `code`.

Keep that `code` — you will send it to the vCX backend in the next step.

“ The backend uses `APP_ID` and `APP_SECRET` to exchange the code for a long-lived access, so the mobile app never needs to store Meta credentials.

6. Complete WhatsApp onboarding on the vCX backend

Use the **3-step onboarding flow** (recommended). It supports both `standard` and `coexistence` modes and keeps the access token server-side.

Step 6a — Exchange the Meta code for an access token

```
POST /api/v2/whatsappSign
Authorization: Bearer <JWT_TOKEN>
Content-Type: application/json

{
  "code": "<META_AUTH_CODE>",
  "mode": "coexistence"
}
```

Response:

```
{
  "success": true,
  "message": "Access token updated successfully",
  "session_id": "<SESSION_ID>",
  "mode": "coexistence",
  "data": "<ACCESS_TOKEN>"
}
```

Save `session_id` and discard `data` (legacy field).

Step 6b — Resolve WABA ID

```
POST /api/v2/whatsappWaba
Authorization: Bearer <JWT_TOKEN>
Content-Type: /json
```

```
{
  "session_id": "<SESSION_ID>"
}
```

Response:

```
{
  "success": true,
  "message": "WABA ID updated successfully",
  "data": {
    "waba_id": "<WABA_ID >",
    "session_id": "<SESSION_ID>"
  }
}
```

Step 6c — Resolve phone number and complete signup

```
POST /api/v2/whatsappPhone
Authorization: Bearer <JWT_TOKEN>
Content-Type: application/json
```

```
{
  "session_id": "<SESSION_ID>"
}
```

Response:

```
{
  "success": true,
  "message": "Whatsapp embedded signup complete with co-existence mode",
  "mode": "coexistence",
  "phone_number": "876543210",
  "phone_id": "<PHONE_ID>",
  "webhook_config": "Webhook configured successfully"
}
```

At this point the backend has:

- Registered the phone number with Meta.
- Configured the vCX webhook.
- Persisted WABA, phone ID, and access token.
- For Coexistence mode, requested history and app-state sync from Meta.

Role requirement: Admin only for all three steps.

7. Verify onboarding completed

Call the status endpoint again:

http GET /api/v2/getWaba Authorization: Bearer ``

Confirm `is_active` is `true` and `webhook_subscription_status` is `subscribed`.

8. Send messages via vCX

Once is complete, use the **separately documented messaging API** to send WhatsApp messages through the vCX platform.

The backend stores the access token and phone ID, so the send-message API only needs the vCX customer JWT (and the recipient/message payload---

Prerequisites

1. The customer account must already exist in vCX (created as a sub-account under the partner's master account via the admin/partner flow).
 2. The customer must have an admin user with verified email and active account status. 3
The mobile app must be able to launch Meta Embedded Signup using vCX's Meta App ID.
 3. Coexistence mode must be enabled before the 3-step onboarding if that mode is desired.
-

Role requirements summary

Endpoint	Required role
POST /v2/login	Any user
GET /v2/getWaba	Any authenticated user
GET /v2/whatsapp-coexistence/status	Admin
POST /v/whatsapp-coexistence/enable	Admin
POST /v2/whatsappSign	Admin
POST /v2/whatsappWaba	Admin
POST /v2/whatsappPhone	Admin

Important notes

- **Legacy endpoint** `POST /api/v2/embeddedSignup` **exists as a single-call onboarding flow, but it** always uses `standard` mode** and does not support Coexistence. Use the 3-step flow above for Coexistence.
- ****Token storage** The 3-step flow uses `session_id` to keep the Meta access token server-side. The mobile app only needs to store the vCX JWT.
- **No login to vCX web UI required:** The mobile app can perform the entire flow via these API calls.

Let me know if you want me to save this as a file in the repo (e.g., `docs/mobile-app-whatsapp-integration.md`).

Revision #2

Created 2026-07-24 11:29:49 UTC by Admin

Updated 2026-07-24 11:44:29 UTC by Admin