



# GETTING YOUR WHATSAPP API CREDENTIALS

Your Business is Verified - Let's Get API Access!

---

## CURRENT STATUS

- Business Verification:  Verified
  - WhatsApp Account:  Approved
  - **Next Step:** Get API credentials and integrate
- 

## STEP-BY-STEP GUIDE

### STEP 1: Go to Meta for Developers

1. **Open new tab and go to:** <https://developers.facebook.com>
  2. **Click "My Apps"** (top right corner)
  3. **Click "Create App"** (green button)
- 

### STEP 2: Create Your App

1. **Select Use Case:**
  - Choose: **"Other"**
  - Click "Next"
2. **Select App Type:**
  - Choose: **"Business"**
  - Click "Next"

#### 3. App Details:

App name: Ultraa Events API  
App contact email: [your business email]  
Business Account: [Select "Ketan InformationTechnologies Pvt.Ltd."]

4. **Click "Create App"**

## 5. Complete Security Check (if asked)

✓ App Created!

---

### STEP 3: Add WhatsApp Product to Your App

1. **In App Dashboard** (you'll see various products)
2. **Find "WhatsApp" card**
3. **Click "Set up"** button on WhatsApp card
4. **Select Your Business:**
  - Choose: "Ketan InformationTechnologies Pvt.Ltd."
  - Click "Continue"
5. **Select Your WhatsApp Account:**
  - You'll see your verified WhatsApp account
  - Click "Continue"

✓ WhatsApp Added to App!

---

### STEP 4: Get Your Phone Number ID

1. **You'll now be on "WhatsApp > Getting Started" page**
2. **Look for section:** "Send and receive messages"
3. **You'll see:**

From: [Your WhatsApp Business Number]  
Phone number ID: 123456789012345

4.  **COPY THIS PHONE NUMBER ID** - Save it in notepad!
- 

### STEP 5: Get Temporary Access Token (For Testing)

1. **On same page, look for:** "Temporary access token"
2. **Click "Copy"** button next to the token
3. **It looks like:**

EAAO...long string...ZD

4.  **SAVE THIS TOKEN** - Valid for 24 hours (for testing only)

⚠ **Note:** This expires in 24 hours. We'll create permanent token later.

---

## STEP 6: Test Your First Message!

Let's test if everything works!

1. **On same page, scroll to:** "Send a test message"
2. **You'll see:**

To: [Input field for phone number]

3. **Enter YOUR personal WhatsApp number:**

Format: 919876543210 (no + or spaces)


4. **Click "Send message"**
5. **Check your WhatsApp** - You should receive:

Hello World!

✅ **If received, API is working!**

---

## STEP 7: Get Your WhatsApp Business Account ID (WABA ID)

1. **Go to:** WhatsApp > API Setup (left sidebar)
  2. **Look for:** "WhatsApp Business Account ID"
  3. **Copy the number** (looks like: 123456789012345)
  4.  **SAVE THIS WABA ID**
-

## STEP 8: Get App ID and App Secret


1. Click "Settings" in left sidebar
2. Click "Basic"
3. You'll see:

App ID: 1234567890123456

App Secret: [Click "Show" button] xxxxxxxxxxxxxxxxx

4.  **COPY BOTH** - Save them securely!
- 

## STEP 9: Create System User (For Permanent Token)

 **Important:** Temporary token expires. System user token never expires!

1. Go back to: <https://business.facebook.com>
2. Click Settings (gear icon, bottom left)
3. Click "Users" > "System users"
4. Click "Add" button
5. System User Details:

System user name: Ultraa Events Bot

System user role: Admin

6. Click "Create system user"

 **System User Created!**

---

## STEP 10: Assign WhatsApp to System User

1. Click on "Ultraa Events Bot" (the system user you just created)
2. Click "Add Assets" button
3. Select "Apps" tab
4. Find your app: "Ultraa Events API"
5. Toggle switch to ON

6. **Select permission: Full control**
7. **Click "Save changes"**

## STEP 11: Generate Permanent Access Token

1. **Still on System User page**
2. **Click "Generate new token" button**
3. **Select Your App:** "Ultraa Events API"
4. **Select Permissions:** (Check these boxes)
  - ☒
  - ☒
5. **Token Duration: Never expire** (60 days or Never - choose Never)
6. **Click "Generate token"**
7. **⚠ IMPORTANT: Copy this token immediately!**

Token looks like: EAAxx

8.  **SAVE THIS TOKEN SECURELY** - You won't see it again!

 **Permanent Token Created!**

## STEP 12: Setup Webhook

Now connect your backend to receive messages.

1. **Go to Developers Dashboard:** <https://developers.facebook.com>
2. **Select Your App:** "Ultraa Events API"
3. **Go to:** WhatsApp > Configuration (left sidebar)
4. **Find "Webhook" section**
5. **Click "Edit" button**
6. **Enter Details:**

Callback URL: <https://your-app.railway.app/webhook/whatsapp>  
Verify Token: ultraa\_secure\_token\_123

(You can use any random string as verify token - remember it!)

## 7. Click "Verify and save"

---

### STEP 13: Add Webhook Code to Your Backend

Add this code to your `server.js`:

```
javascript
```

*// Webhook Verification (GET request)*

```
app.get('/webhook/whatsapp', (req, res) => {  
  const VERIFY_TOKEN = 'ultraa_secure_token_123'; // Must match what you entered in Meta  
  
  const mode = req.query['hub.mode'];  
  const token = req.query['hub.verify_token'];  
  const challenge = req.query['hub.challenge'];  
  
  if (mode === 'subscribe' && token === VERIFY_TOKEN) {  
    console.log('✅ Webhook verified!');  
    res.status(200).send(challenge);  
  } else {  
    console.log('❌ Webhook verification failed!');  
    res.sendStatus(403);  
  }  
});
```

*// Receive Messages (POST request)*

```
app.post('/webhook/whatsapp', async (req, res) => {  
  try {  
    const body = req.body;  
  
    if (body.object === 'whatsapp_business_account') {  
      if (body.entry &&  
        body.entry[0].changes &&  
        body.entry[0].changes[0] &&  
        body.entry[0].changes[0].value.messages &&  
        body.entry[0].changes[0].value.messages[0]) {  
  
        const message = body.entry[0].changes[0].value.messages[0];  
        const from = message.from; // Customer's phone number  
        const messageBody = message.text.body; // Message text  
  
        console.log('📱 Message from ${from}: ${messageBody}');  
  
        // Send reply  
        await sendWhatsAppMessage(from, `You said: ${messageBody}`);  
      }  
    }  
  
    res.sendStatus(200);  
  } catch (error) {  
    console.error('Webhook error:', error);  
    res.sendStatus(500);  
  }  
});
```

```
}  
});
```

## STEP 14: Subscribe to Webhook Fields

1. **In Configuration page** (where you added webhook)
2. **Scroll to "Webhook fields"**
3. **Click "Manage"** button
4. **Subscribe to these fields:**
  - ☒ messages
  - ☒ message\_template\_status\_update
5. **Click "Save"**

☒ **Webhook Connected!**

## STEP 15: Update Your .env File

Add all credentials to your `.env` file:

```
env  
  
# WhatsApp Cloud API Credentials  
WHATSAPP_PHONE_NUMBER_ID=123456789012345  
WHATSAPP_ACCESS_TOKEN=EAAxxxxxxxxxxxxx (permanent token)  
WHATSAPP_BUSINESS_ACCOUNT_ID=123456789012345  
WHATSAPP_VERIFY_TOKEN=ultraa_secure_token_123  
WHATSAPP_APP_ID=1234567890123456  
WHATSAPP_APP_SECRET=xxxxxxxxxxxxxxxxxxxx  
  
# Your Backend URL  
WEBHOOK_URL=https://your-app.railway.app
```

**Replace with your actual values!**

## STEP 16: Create Send Message Function

Add this complete function to your `server.js`:

```
javascript
```



```

const axios = require('axios');

// Send WhatsApp Message Function
async function sendWhatsAppMessage(to, message) {
  const PHONE_NUMBER_ID = process.env.WHATSAPP_PHONE_NUMBER_ID;
  const ACCESS_TOKEN = process.env.WHATSAPP_ACCESS_TOKEN;

  const url = `https://graph.facebook.com/v18.0/${PHONE_NUMBER_ID}/messages`;

  try {
    const response = await axios.post(url, {
      messaging_product: 'whatsapp',
      recipient_type: 'individual',
      to: to, // Format: 919876543210 (no + or spaces)
      type: 'text',
      text: {
        preview_url: false,
        body: message
      }
    }, {
      headers: {
        'Authorization': `Bearer ${ACCESS_TOKEN}`,
        'Content-Type': 'application/json'
      }
    });

    console.log('✅ Message sent successfully:', response.data);
    return response.data;
  } catch (error) {
    console.error('❌ Error sending message:', error.response?.data || error.message);
    throw error;
  }
}

// Export for use in other files
module.exports = { sendWhatsAppMessage };

```

## STEP 17: Test Sending Message from Code

Add this test route to your `server.js`:

```
javascript
```

```
// Test route to send message
app.get('/test-whatsapp', async (req, res) => {
  try {
    // Replace with YOUR phone number
    await sendWhatsAppMessage('919876543210', 'Hello from Ultraa Events! 🇧🇷 This is a test message from our API.');
```

```
    res.json({ success: true, message: 'WhatsApp message sent! Check your phone.' });
  } catch (error) {
    res.status(500).json({ success: false, error: error.message });
  }
});
```

### Test it:

1. Restart your server: `node server.js`
  2. Open browser: `https://your-app.railway.app/test-whatsapp`
  3. Check your WhatsApp!
- 

## STEP 18: Send Interactive Button Messages

Add this function for buttons:

```
javascript
```

*// Send Button Message*

```
async function sendButtonMessage(to, bodyText, buttons) {  
  const PHONE_NUMBER_ID = process.env.WHATSAPP_PHONE_NUMBER_ID;  
  const ACCESS_TOKEN = process.env.WHATSAPP_ACCESS_TOKEN;  
  
  const url = `https://graph.facebook.com/v18.0/${PHONE_NUMBER_ID}/messages`;  
  
  const buttonData = buttons.map((btn, idx) => ({  
    type: 'reply',  
    reply: {  
      id: btn.id || `btn_${idx}`,  
      title: btn.title  
    }  
  }));  
  
  try {  
    await axios.post(url, {  
      messaging_product: 'whatsapp',  
      recipient_type: 'individual',  
      to: to,  
      type: 'interactive',  
      interactive: {  
        type: 'button',  
        body: {  
          text: bodyText  
        },  
        action: {  
          buttons: buttonData  
        }  
      }  
    }, {  
      headers: {  
        'Authorization': `Bearer ${ACCESS_TOKEN}`,  
        'Content-Type': 'application/json'  
      }  
    });  
  
    console.log('✅ Button message sent');  
  } catch (error) {  
    console.error('❌ Error:', error.response?.data || error.message);  
  }  
}
```

*// Usage:*

```
await sendButtonMessage('919876543210',  
  'How can we help you today?',
```

```
[  
  { id: 'view_events', title: 'View Events' },  
  { id: 'contact_support', title: 'Contact Support' }  
]  
);
```

---

## STEP 19: Send List Messages (Event List)

```
javascript
```

*// Send List Message*

```
async function sendListMessage(to, bodyText, buttonText, sections) {
  const PHONE_NUMBER_ID = process.env.WHATSAPP_PHONE_NUMBER_ID;
  const ACCESS_TOKEN = process.env.WHATSAPP_ACCESS_TOKEN;

  const url = `https://graph.facebook.com/v18.0/${PHONE_NUMBER_ID}/messages`;

  try {
    await axios.post(url, {
      messaging_product: 'whatsapp',
      recipient_type: 'individual',
      to: to,
      type: 'interactive',
      interactive: {
        type: 'list',
        body: {
          text: bodyText
        },
        action: {
          button: buttonText,
          sections: sections
        }
      }
    }, {
      headers: {
        'Authorization': `Bearer ${ACCESS_TOKEN}`,
        'Content-Type': 'application/json'
      }
    });

    console.log('✅ List message sent');
  } catch (error) {
    console.error('❌ Error:', error.response?.data || error.message);
  }
}
```

*// Usage - Send Event List:*

```
await sendListMessage('919876543210',
  'Here are our upcoming events:',
  'View Events',
  [
    {
      title: 'Upcoming Events',
      rows: [
        {
          id: 'event_1',
```

```
    title: 'Neon Nights Festival',
    description: '15 Dec 2025, 8:00 PM'
  },
  {
    id: 'event_2',
    title: 'Electronic Dreams',
    description: '22 Dec 2025, 9:00 PM'
  },
  {
    id: 'event_3',
    title: 'New Year Bash 2026',
    description: '31 Dec 2025, 10:00 PM'
  }
]
}
]
);
```

---

## YOUR COMPLETE CREDENTIALS CHECKLIST

Make sure you have all of these saved:

- ☒ Phone Number ID: \_\_\_\_\_
- ☒ Access Token (Permanent): \_\_\_\_\_
- ☒ WABA ID: \_\_\_\_\_
- ☒ App ID: \_\_\_\_\_
- ☒ App Secret: \_\_\_\_\_
- ☒ Verify Token: \_\_\_\_\_
- ☒ Webhook URL: \_\_\_\_\_

---

## TESTING YOUR SETUP

### Test 1: Send Text Message

```
bash
```

```
curl -X POST \
  "https://graph.facebook.com/v18.0/YOUR_PHONE_NUMBER_ID/messages" \
  -H "Authorization: Bearer YOUR_ACCESS_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "messaging_product": "whatsapp",
    "to": "919876543210",
    "type": "text",
    "text": {
      "body": "Hello from Ultraa Events!"
    }
  }'
```

## Test 2: Receive Message

1. Send a WhatsApp message to your business number
2. Check Railway logs: [railway logs](#)
3. You should see: " 🧩 Message from..."

---

## ⚠️ IMPORTANT NOTES

### Message Limits:

- **Tier 1** (New accounts): 50 conversations/day
- **Tier 2** (after 7 days): 250 conversations/day
- **Tier 3** (after quality rating): 1,000 conversations/day
- **Tier 4** (verified business): Unlimited

### Conversation Windows:

- Customer-initiated: 24-hour window (free messages)
- Business-initiated: Must use approved templates

### Rate Limits:

- 80 messages per second
  - 200,000 messages per day (Cloud API)
-

## NEXT STEPS

Now that you have API access:

1. ☒ Integrate with your existing backend code
  2. ☒ Create message templates for:
    - Welcome message
    - Payment confirmation
    - Ticket delivery
    - Event reminders
  3. ☒ Test complete flow
  4. ☒ Deploy to production
- 

## SUPPORT LINKS

- **API Reference:** <https://developers.facebook.com/docs/whatsapp/cloud-api/reference>
  - **Message Templates:** <https://business.facebook.com/wa/manage/message-templates>
  - **API Status:** <https://developers.facebook.com/status/>
  - **Support:** <https://developers.facebook.com/support/bugs/>
- 

Ready to integrate? Let me know which step you're on and I'll help! 