Hello World: Get Twilio SMS Working in 30 Minutes

Goal: Prove the basic SMS → n8n → Response flow works with minimal complexity

What You'll Build:

- Receive SMS with 2-3 parameters (From, Body, optional MessageSid)
- Simple pattern matching on message content
- Return static text responses
- No database, no queue mode, no complexity

Time Required: 20-30 minutes

Prerequisites Checklist

Before starting, you need:

- □ Twilio account (free trial works)
- □ Twilio phone number ($1-2/month)
- □ ngrok account with reserved domain ($8/month)
- □ VPS with Docker installed
- □ Basic terminal/SSH access

Step 1: Minimal docker-compose.yml (5 minutes)

Replace your entire `docker-compose.yml` with this minimal version:

yaml

```yaml
version: '3.8'

services:
  # n8n - Just the basics
  n8n:
    image: n8nio/n8n:latest
    container_name: n8n
    restart: unless-stopped
    ports:
      - "127.0.0.1:5678:5678"
    environment:
      # Minimal configuration
      - N8N_HOST=${N8N_HOST}
      - N8N_PORT=5678
      - N8N_PROTOCOL=https
      - WEBHOOK_URL=${WEBHOOK_URL}
      - GENERIC_TIMEZONE=America/New_York
      - N8N_LOG_LEVEL=info
    volumes:
      - /root/n8n-data:/home/node/.n8n

  # ngrok - Public access
  ngrok:
    image: ngrok/ngrok:alpine
    container_name: ngrok
    restart: unless-stopped
    command:
      - "http"
      - "n8n:5678"
      - "--domain=${NGROK_DOMAIN}"
    environment:
      - NGROK_AUTHTOKEN=${NGROK_AUTHTOKEN}
    ports:
      - "127.0.0.1:4040:4040"
    depends_on:
      - n8n
```

What we removed:

- PostgreSQL (using default SQLite)
- Redis (no queue mode yet)
- Advanced security (keeping it simple)

Step 2: Create Minimal .env File (2 minutes)

Create `/opt/ai-agent-platform/.env`:

bash

```
# ngrok
NGROK_AUTHTOKEN=your_ngrok_token_here
NGROK_DOMAIN=your-domain.ngrok.io

# n8n
N8N_HOST=your-domain.ngrok.io
WEBHOOK_URL=https://your-domain.ngrok.io/
```

Get your ngrok token:

1. Go to https://dashboard.ngrok.com/get-started/your-authtoken
2. Copy the token
3. Paste into `.env`

Get your ngrok domain:

1. Go to https://dashboard.ngrok.com/cloud-edge/domains
2. Create a new domain (or use existing)
3. Use format: `your-name.ngrok.io`

Step 3: Deploy the Stack (3 minutes)

bash

```
# SSH to your VPS
ssh agent-vps

# Navigate to project
cd /opt/ai-agent-platform

# Stop any existing containers
docker compose down

# Start the minimal stack
docker compose up -d

# Check status (wait ~30 seconds)
docker compose ps

# Both should show "Up" status:
# n8n      Up
# ngrok    Up
```

Verify ngrok tunnel:

bash

```
# Check ngrok is connected
curl http://localhost:4040/api/tunnels | jq

# You should see your domain active
```

Step 4: Access n8n UI (2 minutes)

1. **Open browser** to: `https://your-domain.ngrok.io`
   - (Yes, use https - ngrok provides SSL automatically)
2. **First-time setup:**
   - Email: your-email@example.com
   - Password: (create a secure password)
   - Click "Next"
3. **You're in!** You should see the n8n workflow canvas.

Step 5: Create "Hello World" Workflow (10 minutes)

5.1 Create New Workflow

1. Click **"Add workflow"** (top right)
2. Click the workflow name (top left) and rename to: `SMS Hello World`

5.2 Add Webhook Node

1. Click **"+" button** on canvas
2. Search for **"Webhook"**
3. Click it to add

Configure Webhook:

- **Webhook URLs:** Production
- **HTTP Method:** POST
- **Path:** `twilio-hello`
- **Response Mode:** `Respond to Webhook`
- **Response Code:** 200
- **Respond With:** Text

Click "Listen for Test Event" - leave this open

5.3 Add Code Node for Logic

1. Click **"+" button** to add another node
2. Search for **"Code"**
3. Click to add
4. Connect Webhook → Code (drag the arrow)

Configure Code Node:

Paste this code:

📋 ✓

javascript

```
// Get incoming SMS data
// Twilio sends: From, To, Body, MessageSid
const fromNumber = $input.item.json.body.From || 'Unknown';
const messageBody = $input.item.json.body.Body || '';
const messageSid = $input.item.json.body.MessageSid || 'No-ID';

// Simple pattern matching for responses
let response = '';

// Check message content (case-insensitive)
const msg = messageBody.toLowerCase().trim();

if (msg.includes('hello') || msg.includes('hi')) {
  response = '👋 Hello! Thanks for texting. This is a test bot.';

} else if (msg.includes('help')) {
  response = 'Try saying: hello, status, or info';

} else if (msg.includes('status')) {
  response = '✅ Bot is working! Current time: ' + new Date().toLocaleTimeString();

} else if (msg.includes('info')) {
  response = 'This is an n8n SMS bot. Message from: ' + fromNumber;

} else {
  response = 'Got your message: "' + messageBody + '". Try: hello, help, status, or info';
}

// Return the response
return {
  json: {
    // Original data
    from: fromNumber,
    body: messageBody,
    messageSid: messageSid,
    receivedAt: new Date().toISOString(),

    // Our response
    response: response,

    // Debug info
    params: $input.item.json.body
  }
};
```

5.4 Add Response Node

1. Click **"+" button** after Code node
2. Search for **"Respond to Webhook"**
3. Click to add
4. Connect Code → Respond to Webhook

Configure Response:

- **Response Body:** `{{$json.response}}`
- Click outside to save

5.5 Save Workflow

1. Click **"Save"** button (top right)
2. Click **"Activate"** toggle to ON (top right)

Your webhook URL is now:

```
https://your-domain.ngrok.io/webhook/twilio-hello
```

Copy this URL - you'll need it for Twilio!

Step 6: Configure Twilio (5 minutes)

6.1 Get Twilio Phone Number

If you don't have one:

1. Go to https://console.twilio.com/
2. **Phone Numbers → Manage → Buy a number**
3. Choose a toll-free number (~$2/month)
4. Buy it

6.2 Configure Webhook

1. Go to **Phone Numbers → Manage → Active Numbers**
2. Click your phone number
3. Scroll to **Messaging Configuration**
4. Under **"A MESSAGE COMES IN":**
    - **Webhook:** `https://your-domain.ngrok.io/webhook/twilio-hello`

- **HTTP Method:** POST

5. Click **Save**

Step 7: Test It! (3 minutes)

7.1 Send Test SMS

From your personal phone, text your Twilio number:

```
hello
```

Expected Response (within 2-5 seconds):

```
👋 Hello! Thanks for texting. This is a test bot.
```

7.2 Try Other Commands

```
SMS: "help"
Response: "Try saying: hello, status, or info"

SMS: "status"
Response: "✅ Bot is working! Current time: 2:45:30 PM"

SMS: "info"
Response: "This is an n8n SMS bot. Message from: +15551234567"

SMS: "blah blah blah"
Response: "Got your message: "blah blah blah". Try: hello, help, status, or info"
```

Step 8: Debug if Not Working (5 minutes)

Check n8n Execution Log

1. In n8n UI, click **"Executions"** (left sidebar)
2. You should see your webhook executions
3. Click one to see the data flow
4. Check each node for errors

Check ngrok Status

bash

```bash
# On your VPS
curl http://localhost:4040/api/tunnels | jq

# Should show:
# - public_url: "https://your-domain.ngrok.io"
# - status: "online"
```

Check n8n Logs

bash

```bash
# On your VPS
docker compose logs n8n --tail=50

# Look for:
# ✅ "Webhook waiting for requests"
# ✅ "Server ready on port 5678"
```

Common Issues

❌ "Webhook not found"

- Solution: Make sure workflow is Activated (toggle ON)
- Solution: Check webhook path matches Twilio config exactly

❌ No response received

- Solution: Check n8n logs for errors
- Solution: Verify ngrok tunnel is active

- Solution: Test webhook directly with curl

❌ "Invalid request"

- Solution: Webhook must be POST method
- Solution: Check Response Mode is set correctly

Test Webhook Directly

bash

```bash
# From your local machine
curl -X POST https://your-domain.ngrok.io/webhook/twilio-hello \
  -d "From=+15551234567" \
  -d "Body=hello" \
  -d "MessageSid=test123"

# Should return:
# 👋 Hello! Thanks for texting. This is a test bot.
```

Success Checklist

You know it's working when:

- ☑ n8n UI accessible at your domain
- ☑ Workflow shows "Active" (green toggle)
- ☑ ngrok tunnel shows "online" status
- ☑ Twilio webhook configured with your URL
- ☑ Test SMS receives response within 5 seconds
- ☑ Different messages trigger different responses
- ☑ Execution log shows successful runs

What You've Proven

✅ Twilio → n8n connection works ✅ Webhook receives SMS parameters ✅ Code can process message content ✅ Response returns to sender ✅ Basic routing logic functions

---

Next Steps (After Hello World Works)

Once you've proven the basic flow:

1. **Add signature validation** (prevent spoofing)
2. **Store conversations** (add PostgreSQL)

3. **Add AI responses** (integrate LLM)
4. **Enable queue mode** (handle load)
5. **Implement buffering** (better UX)

Refer to the **Production Baseline** artifact for the full implementation.

Quick Reference

Your URLs

- **n8n UI:** `https://your-domain.ngrok.io`
- **Webhook:** `https://your-domain.ngrok.io/webhook/twilio-hello`
- **ngrok Inspector:** `http://localhost:4040` (when SSH'd to VPS)

Quick Commands

bash

```bash
# Restart everything
docker compose restart

# View logs
docker compose logs -f

# Check status
docker compose ps

# Stop everything
docker compose down

# Start everything
docker compose up -d
```

Twilio Quick Links

- **Console:** https://console.twilio.com/
- **Phone Numbers:** https://console.twilio.com/phone-numbers/incoming
- **Logs:** https://console.twilio.com/monitor/logs/sms

Support

If you're stuck:

1. Check execution logs in n8n
2. Check ngrok tunnel status
3. Check Twilio logs for delivery errors
4. Test webhook with curl directly
5. Verify all services are "Up" status

Common "Gotchas":

- Workflow must be **Activated** (toggle ON)
- Webhook path is **case-sensitive**
- ngrok domain must match .env exactly
- Twilio webhook must be **HTTPS** (not HTTP)

You're Done! 🎉

Your basic SMS bot is now working. When you're ready for production features, move to the Production Baseline architecture with PostgreSQL, Redis, queue mode, and all the bells and whistles.

Time to celebrate with a working prototype! 🚀 📱