

THE BROKER TIMES

GUIDE 5 OF 5

Troubleshooting

AI WhatsApp Agent Engine · thebrokertimes.com.au

In rough order of how often these actually happen.

Nothing happens when I message the number

Check in this order:

1. Is the scenario toggled ON?
 2. Meta app → WhatsApp → Configuration → Webhook fields: is **messages** subscribed? Being verified is not the same as being subscribed, and this is the most common miss.
 3. Is your mobile in the test number's allowed recipient list?
 4. Make → the scenario → **History**: is there any run at all? If yes, the webhook is fine and the problem is downstream. If no, it's Meta-side.
-

It worked yesterday and stopped today

Your access token was the temporary one. Guide 1, Step 3 — generate a permanent System User token and update the connection in Make.

"Module 33: Invalid JSON" / parse errors

The model returned something that isn't clean JSON — usually it wrapped the response in a markdown code fence.

- Confirm `"response_format": { "type": "json_object" }` is still in module 32's body
 - Add a line to the end of your system prompt: "Output raw JSON only. Never wrap it in markdown code fences."
 - Some cheaper models are unreliable at this. If it recurs, change model.
-

Module 32 returns 401

The OpenRouter key. Check the header reads `Bearer sk-or-...` — the word Bearer, one space, then the key. A pasted newline at the end of the key does this too.

Module 32 returns 400

Malformed request body. Two causes, in order of likelihood:

1. Someone pasted the system prompt directly into module 32 instead of module 30. The prompt goes in module 30 — that module exists to escape it safely.
 2. The quote marks around `{{30.json}}` or `{{31.json}}` in module 32's body were removed. They are required. Transform to JSON escapes the text but does not add the surrounding quotes, so the body has to supply them.
-

The first message from a new contact does nothing, but later ones work

Modules 3 and 28 need **"Continue the route when there are no results"** switched on. With no existing records, the search returns nothing and the flow stops dead.

A client sent a photo and the agent said nothing

Module 2's filter has four branches: text, reply button, list row, and everything else. If the fourth is missing, media messages are dropped before the scenario runs. Open module 2, check the filter, and confirm the fourth condition group is there. The scenario history will show no run at all for that message, which is the giveaway.

It keeps asking my name over and over

The stage isn't being written, so every message looks like the name answer again.

1. Module 60 must point at `wa_contacts`, not `wa_messages`. This is the most common mis-map in the build.
 2. Check the `stage` field exists in the `wa_contacts` data structure, spelled exactly `stage`, type Text.
 3. Open the contact's row in `wa_contacts`. After the name answer it should read `AWAITING_TYPE`. If it still reads `AWAITING_NAME`, module 60 didn't write.
-

The loan-type list never appears

Module 62 either has no connection selected, or a row value is over Meta's limit. Row title max 24 characters, description max 72, button label max 20. Over-length values are rejected by Meta at send time, so the module shows green in Make and the client sees nothing. Count the characters on any row you edited.

It replies to someone who sent STOP

Check module 3 is pointed at `wa_contacts` and that `opted_out` is a **Boolean** field, not Text. A Text field containing the word "true" does not satisfy the router's boolean filter, so the AI route stays open to them.

It has no memory / re-asks the same questions

- Is `created_at` in `wa_messages` a **Date** field? If it's Text, the sort is alphabetical and you're feeding it a random 15 messages.
 - Is module 28's sort set to `created_at` **descending**?
 - Are modules 35 and 36 actually writing? Check the data store has rows.
 - Is module 29 (Text aggregator) sourced from module 28?
-

It replies twice to one message

Meta retries when it doesn't get a fast acknowledgement. Turn on **Sequential processing** in scenario settings. If it persists, the scenario is running too slowly — reduce history length and use a faster model.

Replies take ten seconds or more

- Model choice is usually the culprit. Try a faster one.
 - History length: 15 messages of context on every turn adds up.
 - A large knowledge base is re-sent every single message. Trim it.
-

It quoted a rate

Stop. Turn the scenario off.

1. Find the exact exchange in `wa_messages`
2. Strengthen the HARD RULES section with that specific phrasing as an example
3. Re-run every test in Guide 4, Step 2
4. Consider a better model — this is exactly where cheap models fail
5. Only then turn it back on

If it happened with a real client, tell your licensee. Don't sit on it.

The data store is full

`wa_messages` grew past its allocation. Either increase the size in the data store settings, or add the pruning scenario described in `data-structures/data-stores.pdf`.

Where to look first, always

Make → your scenario → **History**. Click the run, click the module with the red mark, read the input and output bundles. The answer is nearly always visible there, and it beats guessing.