

Ticketon Integration

Connects **Ticketon** with the Newo AI agent so callers can discover events, choose a city, narrow available sessions by venue and time, and receive the selected Ticketon event page by WhatsApp.

Time to first success: about 15 minutes when the Ticketon connector is already available and the delivery team has access to **Newo Builder**. Allow another 10 minutes when TalkMe WhatsApp settings still need to be supplied and tested.

Before You Start

Make sure you have:

- access to **Newo Builder**, including permission to edit module settings and click **Publish All**;
- the **Ticketon connector** available in the project;
- an approved TalkMe WhatsApp sender and token if WhatsApp link delivery is part of the delivery scope;
- one supported test city, event or movie, and date with real on-sale sessions for acceptance testing.

Ticket purchase, seat selection, payment, refunds, and ticket cancellation are outside this integration. The agent discovers events and sessions, then directs the caller to the exact Ticketon event page.

What the AI can do

- **Search Ticketon events.** The caller can use a specific title, artist, movie, show, meaningful category, or date-based request.
- **Resolve the city before searching.** The caller chooses exactly one city; the agent never silently chooses a location.
- **Disambiguate event results.** When several events match, the agent presents numbered options and waits for a selection.
- **Fetch the complete on-sale session list.** The agent uses Ticketon data for venues, times, language, price, and availability.
- **Narrow sessions conversationally.** After sessions are fetched, the agent asks for venue and approximate time, then filters the existing session list without repeating the Ticketon search.
- **Offer the nearest available time.** If no session is close to the requested time, the agent clearly labels the nearest available option.
- **Handle cinema dates.** When a movie search needs a date, the agent asks for the date and keeps the previously supplied movie and city.
- **Send the selected Ticketon page through WhatsApp.** Delivery starts only after the agent has available sessions and the caller confirms the full destination number.

Features at a glance

Feature	Included
Event and movie search	✓
One-city selection before search	✓
City vocabulary refresh on publish and when the cached list expires	✓
Event option disambiguation	✓

Cinema session lookup by date and city	✓
Venue and approximate-time narrowing	✓
Nearest-session fallback	✓
WhatsApp event-link delivery	✓ when TalkMe is configured
Ticket purchase, payment, or seat reservation	✗ completed by the caller on Ticketon
Refunds or ticket cancellation	✗ route to a human

Setup

1. Prepare Ticketon and TalkMe

Ticketon access and TalkMe delivery credentials are configured on the **Ticketon connector**.

1. Confirm that the connector uses the required Ticketon language and country.
2. If WhatsApp delivery is in scope, add the approved **TalkMe API key** and **TalkMe source**.
3. Keep the default TalkMe service address unless support provides a different approved value.

2. Get your delivery credentials

No Ticketon credential is entered in the module settings. The connector is supplied by the platform team.

For WhatsApp delivery, obtain these values from the approved TalkMe sender account:

- **TalkMe API key** — the delivery token;
- **TalkMe source** — the approved WhatsApp sender or source.

Do not enable WhatsApp for a live project until both values are available and a real delivery test has passed.

3. Configure in Newo

1. Open the project in **Newo Builder**.
2. Find the **Module - Ticketon** settings group.
3. Keep **Enable event discovery** on for projects that should answer Ticketon event questions.
4. Keep **Enable WhatsApp link delivery** on only when TalkMe is configured and WhatsApp delivery is part of the agreed scope.
5. Keep **Install Ticketon scenarios** on unless the project deliberately uses a manually maintained canvas scenario.
6. Click **Publish All**.

4. Hit Publish All

After **Publish All**, the integration:

- creates or starts the Ticketon connector;
- registers event discovery, session narrowing, and WhatsApp delivery actions;
- refreshes the Ticketon city vocabulary;
- adds the **Event Discovery** intent and scenario when **Install Ticketon scenarios** is on.

How to verify:

- Ask *"Find concerts in Almaty this weekend."* The agent should search without asking for the city again.
- Complete a search to the session stage. The agent should ask for venue and approximate time before offering WhatsApp.
- If WhatsApp is enabled, confirm one delivery to a controlled test number.

How to test that everything works

Use a Ticketon event with known active sessions. Run the tests in this order so each step proves a separate delivery requirement.

Test event discovery

1. Ask: *"Find concerts in Almaty this weekend."*
2. The agent should make one short immediate search promise in the caller's language.
3. It should present Ticketon results or a specific recovery question, never event facts from general knowledge.

Test a missing city

1. Ask: *"I want tickets for Anacondaz."*
2. The agent should ask for one city and stop.
3. After the city is supplied, it should make the search promise without asking for the event again.

Test event disambiguation

1. Use a request that returns several matching events.
2. The agent should present numbered options.
3. Select one option. The agent should inspect that event and retrieve its sessions before asking for venue or time preferences.

Test cinema sessions

1. Ask for a movie in a supported city without a date.
2. If Ticketon requires a date, the agent should ask only for the preferred date.
3. Supply the date. The agent should reuse the movie and city and continue the lookup.

Test local session narrowing

1. Continue until the agent lists available venues and asks for a venue and approximate time.
2. Choose an offered venue and time, such as *"Arena Cinema around 7 PM."*
3. The agent should make a short promise to filter the fetched sessions, then present only matching sessions.
4. If no close time exists, it should clearly identify the nearest available session.
5. Choose an unknown venue. The agent should show the fetched venue choices again without starting a new event search.

Test stale-result isolation

1. Complete a lookup for one event.
2. Ask for a different event.
3. The agent must start a new lookup and must not reuse venues, sessions, or links from the previous event.

Test WhatsApp link delivery

1. Continue until matching sessions are presented and the agent offers WhatsApp.

2. Ask for the link without providing a number. The agent should ask only for the full destination number.
3. Provide a full number without confirmation. The agent should ask whether that exact number is correct and must not start delivery.
4. Confirm the number. The agent should make one short immediate send promise in the caller's language.
5. The agent should confirm delivery only after TalkMe returns a successful result.

Scenarios

The integration ships with one editable canvas scenario.

What gets added on Publish All

Intent	Scenario
Event Discovery	Event Discovery

This is a **reference implementation**. You can edit its tone, language, clarifying questions, and transfer wording in the canvas UI.

Updates do not overwrite your edits. If the Event Discovery intent or scenario was edited previously, an integration update will not replace it. To receive the newer session-narrowing behavior, delete the edited copy and click **Publish All**, then reapply approved custom wording to the new baseline.

Why action promises matter

The scenario uses three short, action-specific promises. They are semantic triggers, so the exact wording changes with the caller's language. Each promise must be one first-person sentence, describe the immediate action, contain no question or result, and stop after that sentence.

Scenario 22 — Event Discovery

Runs when the caller asks about events, movies, concerts, theatre, circus, sports, museums, master classes, sessions, venues, prices, or delivery of a selected event link.

1. Resolve the current event target. Ask only when the caller supplied no specific title, artist, movie, show, meaningful category, or date-based request.
2. Resolve exactly one city. Never choose a city for the caller or ask again when one city is already known.
3. Make a short immediate promise to search for matching event options and wait for the new Ticketon result.
4. Present only the facts supplied by that result. If several events match, ask the caller to choose one. If cinema needs a date, ask only for the date.
5. After Ticketon sessions are fetched, ask for the preferred venue and approximate time.
6. Make a separate short promise to filter the fetched sessions. Present matching sessions or the nearest available time; do not repeat the external search for a venue/time refinement.
7. Offer WhatsApp only after available sessions and the exact Ticketon event page are ready.
8. Collect and explicitly confirm the full destination number, then make a short immediate promise to send the selected event link by WhatsApp.
9. Confirm delivery only from the newest delivery result.

Triggered by intent: *Event Discovery*.

Action promises to preserve:

- **Search promise** — meaning: *"I will search for matching event options now."*
- **Session-filter promise** — meaning: *"I will filter the fetched sessions by your venue or time preference now."*
- **WhatsApp promise** — meaning: *"I will send the selected event link by WhatsApp now."*

These are meaning references, not fixed English scripts. Compose each promise directly in the caller's language.

What the agent reads to know the result:

- **Ticketon cities panel** — the current city vocabulary used to recognise caller-supplied city names.
- **Latest Ticketon result** — a self-contained instruction containing only the event options, missing detail, venues, sessions, nearest time, exact event page, delivery outcome, or safe recovery message for the current action.

Safe customization in the canvas UI:

-  Change greetings, tone, and language around action promises.
-  Add approved transfer wording for purchase, refund, or cancellation requests.
-  Adjust how many supplied options the agent reads when the Ticketon result permits it.
-  Combine an action promise with a question, explanation, event fact, city, title, venue, time, price, or link.
-  Remove the city gate, venue/time narrowing step, or full-number confirmation.
-  Allow the agent to reuse an earlier result after the caller names a different event.
-  Promise ticket purchase, reservation, payment, refund, or cancellation.

Action promises reference

Meaning to preserve	Starts	Required before the promise
<i>"I will search for matching event options now."</i>	New event lookup or selected-event inspection	Current target and exactly one city
<i>"I will filter the fetched sessions by your venue or time preference now."</i>	Local narrowing of already fetched sessions	Current session choices plus a venue or approximate-time preference
<i>"I will send the selected event link by WhatsApp now."</i>	WhatsApp delivery	Current Ticketon page, available sessions, full phone number, explicit confirmation

Example dialogs

Example 1 — direct event search

Caller: Find concerts in Almaty this weekend. **Agent:** I will search for matching event options now.
Agent: I found these Ticketon events... Which one do you mean?

Example 2 — local session narrowing

Agent: This event has sessions at Arena Cinema and Mall Cinema. Which venue and approximate time do you prefer? **Caller:** Arena Cinema around 7 PM. **Agent:** I will filter the fetched sessions by your venue

and time now. **Agent:** The nearest available Arena Cinema session is at 7:30 PM...

Example 3 — WhatsApp confirmation gate

Caller: Send the link to WhatsApp. **Agent:** What full WhatsApp number should I use? **Caller:** +7 700 000 00 00. **Agent:** Is +7 700 000 00 00 correct? **Caller:** Yes. **Agent:** I will send the selected event link by WhatsApp now. **Agent:** The link has been sent.

Example 4 — unsupported purchase request

Caller: Reserve two seats and pay for them. **Agent:** I can help find the event and send its Ticketon page, but ticket purchase and seat reservation must be completed on Ticketon.

Customizing the agent

The shipped scenario is a reference implementation. Safe changes include approved brand tone, translations, shorter option presentation, and business-specific transfer wording.

Always preserve the three action meanings, phrase-only behavior, target-before-city order, local venue/time narrowing, and WhatsApp confirmation gate.

Upgrade reminder: edited intents and scenarios are not overwritten. After a Ticketon update, compare this changelog with the live canvas. Delete the edited copy and click **Publish All** when the project needs the newer shipped behavior.

FAQ

Q. Does the agent buy tickets?

No. It provides event and session information and can send the exact Ticketon event page. The caller completes purchase on Ticketon.

Q. Why does the agent ask for a city before searching?

Ticketon results are location-specific. Exactly one caller-selected city is required for a safe lookup.

Q. Why does cinema sometimes need a date?

Movie sessions are date-specific. The agent keeps the movie and city, asks for the missing date, and continues.

Q. Why does the agent ask for venue and approximate time after fetching sessions?

The integration first obtains the complete on-sale session list, then narrows that list locally. This avoids repeated external searches during venue/time refinement.

Q. What happens when no session is close to the requested time?

The agent presents the nearest available time and labels it as the nearest option.

Q. Can WhatsApp work without TalkMe settings?

No. The module switch enables the behavior, but the connector still needs the approved TalkMe token and sender.

Q. What happens if the city vocabulary is unavailable?

The agent still asks for the city. If the city cannot be resolved, it asks for a clearer city name rather than searching without a location.

Q. When is the city vocabulary refreshed?

It refreshes on **Publish All** and when a session starts after the cached vocabulary has expired. The default cache lifetime is 24 hours.

Common errors and recovery

The updated venue/time step does not appear after upgrade

Likely cause: the project has an edited Event Discovery scenario, so the update preserved the older canvas copy.

How to recover:

1. Save any approved custom wording outside the canvas.
2. Delete the edited Event Discovery scenario and intent.
3. Confirm **Install Ticketon scenarios** is on.
4. Click **Publish All**.
5. Reapply approved wording without removing the action promises or gates.

How to verify: after sessions are fetched, the agent asks for venue and approximate time before offering WhatsApp.

The agent starts a new search after the caller chooses a venue or time

Likely cause: the session-filter promise is missing or the live canvas still contains the older scenario.

How to recover: restore the new Event Discovery scenario as described above. Preserve a separate immediate promise to filter the fetched sessions.

How to verify: choosing a venue/time produces filtered or nearest sessions from the current event without showing a new event list.

The agent asks to repeat the event during session narrowing

Likely cause: the fetched session list expired, was cleared at the end of a conversation, or belongs to another event.

How to recover: start a fresh event lookup with the current event and city.

How to verify: the new lookup reaches venue/time selection and then filters normally.

The agent keeps asking for the city

Likely cause: the city name did not resolve to a supported Ticketon city.

How to recover:

1. Ask for a more specific city name.
2. Test with a major supported city such as Almaty or Astana.
3. Click **Publish All** if the city vocabulary may be stale.

How to verify: the next request proceeds to event options instead of asking for the city again.

WhatsApp link is not sent

Likely cause: WhatsApp is disabled, TalkMe settings are missing, no current on-sale session or event page is available, or the full phone number was not explicitly confirmed.

How to recover:

1. Confirm **Enable WhatsApp link delivery** is on.
2. Confirm **TalkMe API key** and **TalkMe source** are present on the connector.
3. Run a new event lookup through venue/time narrowing.
4. Provide and explicitly confirm a full test number.

How to verify: the agent confirms delivery only after the send promise and a successful delivery result.

Event discovery is unavailable

Likely cause: **Enable event discovery** is off or the project was not republished after configuration.

How to recover: turn **Enable event discovery** on and click **Publish All**.

How to verify: a target-and-city request produces a search promise and a new Ticketon result.

Limitations

- Ticket purchase, payment, seat selection, refunds, and ticket cancellation are outside the integration.
- WhatsApp sends the selected Ticketon event page, not a ticket or a payment link generated by Newo.
- One lookup uses one Ticketon city at a time.
- Venue/time narrowing applies only to the most recently fetched session list in the current conversation.
- The integration may offer the nearest session when no time falls close to the caller's request.

Delivery manager handoff

Target release: **Ticketon 0.3.2**.

Delivery decisions to record

Decision	Record for handoff
Event discovery	Enabled or disabled
WhatsApp delivery	In scope or out of scope
TalkMe sender	Approved sender name; never copy the token into the handoff document
Canvas ownership	Shipped scenario or customer-maintained edited copy
Supported languages tested	Languages and channels used in acceptance
Test data	City, event/movie, date, venue, and expected session time

Go-live acceptance

Event discovery is ready only when all mandatory checks pass:

- ☐ target and city in one message start a search without redundant questions;
- ☐ missing target is requested before missing city;
- ☐ multiple events require a caller selection;
- ☐ cinema can request a missing date while preserving movie and city;
- ☐ sessions lead to venue/time selection and local narrowing;
- ☐ an unknown venue leads back to the fetched venue options;
- ☐ a new event never reuses the previous event's sessions or link;
- ☐ responses and action promises use the caller's language;
- ☐ unsupported purchase/refund/cancellation requests are not promised.

When WhatsApp is in scope, these checks are also mandatory:

- ☐ TalkMe settings are present;
- ☐ a phone number by itself does not start delivery;
- ☐ explicit confirmation of the full number is required;
- ☐ delivery is confirmed only after a successful result;
- ☐ failure does not produce a false success claim.

Escalation package

If acceptance fails, send support:

- project name and deployed Ticketon version;
- channel and caller language;
- exact caller messages and agent replies from the failed step;
- test city, event/movie, date, venue, and approximate time;
- whether the Event Discovery scenario was edited before the upgrade;
- whether **Enable event discovery**, **Enable WhatsApp link delivery**, and **Install Ticketon scenarios** are on;
- for WhatsApp issues, whether the approved sender was configured and the full number was explicitly confirmed.

Never include API keys, TalkMe tokens, real customer phone numbers, or other credentials in the handoff.

All settings reference

The table below lists the Ticketon module settings. Hidden extraction settings ship with working defaults and should be changed only by support.

Name	Description	Required	Default	Hidden
ticketon_documentation_url	Read-only Ticketon support reference.	No	https://ticketon.kz	No
ticketon_enable_event_discovery	Enables event	Yes	on	No

	search, city resolution, session retrieval, and local venue/time narrowing.			
ticketon_enable_whatsapp_link_delivery	Enables delivery of the selected Ticketon page after session and phone confirmation.	No	on	No
ticketon_setup_scenarios	Adds the shipped Event Discovery intent and scenario on publish.	No	on	No
ticketon_cities_cache	Stores the Ticketon city vocabulary.	No	(empty)	Yes
ticketon_cities_cache_updated_at	Stores the last successful city-vocabulary refresh time.	No	(empty)	Yes
ticketon_cities_cache_range_seconds	Controls city-vocabulary lifetime.	No	86400	Yes
ticketon_event_discovery_tool_schema	Defines the structured event and session-preference fields.	No	shipped configuration	Yes

ticketon_event_discovery_tool_instructions	Defines how event, city, date, venue, and time are extracted.	No	shipped configuration	Yes
ticketon_link_delivery_tool_schema	Defines the WhatsApp number and confirmation fields.	No	shipped configuration	Yes
ticketon_link_delivery_tool_instructions	Defines how the confirmed WhatsApp destination is extracted.	No	shipped configuration	Yes
ticketon_setup_persona_id	Identifies the internal setup record used to install actions.	No	(empty)	Yes

Connector settings used by the delivery team:

Name	Description	Required	Default	Hidden	Edit
talkme_api_key	TalkMe delivery token.	Yes for WhatsApp	(empty)	Connector-controlled	⚠
talkme_source	Approved TalkMe WhatsApp sender or source.	Yes for WhatsApp	(empty)	Connector-controlled	⚠
talkme_base_url	Approved TalkMe service address.	No	https://lcab.talk-me.ru	Connector-controlled	⚠
accept_language	Preferred Ticketon response language.	No	ru-KZ	Connector-controlled	⚠
country_code	Ticketon country scope.	No	KAZ	Connector-controlled	⚠

Changelog

0.3.2 — Readable localized results

- **Localized Ticketon data stays readable in agent instructions.** Event discovery and WhatsApp delivery results now preserve names and messages in their original scripts instead of converting them to escaped Unicode sequences.

0.3.1 — Delivery-ready event discovery

- **Sessions are narrowed after one complete fetch.** The agent asks for venue and approximate time, filters the fetched sessions, and labels the nearest available time when there is no close match.
- **New event requests are isolated from earlier results.** A new title cannot inherit the previous event's venues, sessions, or link.
- **Results are self-contained and language-aware.** The agent follows the newest Ticketon result and answers in the caller's language without relying on stale canvas state.
- **WhatsApp validation is independent and stricter.** A current Ticketon page, available session, full number, and explicit confirmation are all required before delivery.
- **City vocabulary uses a 24-hour cache.** It refreshes on publish and when an expired cache is encountered at session start, while overlapping refreshes are prevented.
- **Delivery note:** edited canvas scenarios are preserved. Delete the edited Event Discovery copy and click **Publish All** to adopt this release's venue/time narrowing behavior.

0.2.1 — Safer search and session selection

- Event search requires a meaningful target and one city.
- Movie lookups ask for a missing date without losing the selected title or city.
- WhatsApp is offered only after a selected event has an exact Ticketon page and on-sale sessions.