Luma distribution: manual links, chapter accounts and HQ-managed events

Status: Current product help

AI Tinkerers uses Luma for event discovery and distribution. AI Tinkerers remains the source of truth for registration, screening, acceptance, event updates, admission QR codes and check-in. A Luma RSVP alone is not acceptance to an AI Tinkerers event.

Choose the setup offered for your chapter

Mode	What you need	What it does
Manual Luma link	An existing or manually created Luma event	Copy the prepared event details and AIT registration link into Luma, then save the Luma URL in AIT. This mode does not provide API guest sync.
Chapter-owned Luma API	HQ enablement plus the chapter calendar's valid Luma Plus API key	Connect or create the Luma page and synchronize supported guest records into AIT for review.
HQ-managed Luma	HQ has made this mode available for your chapter/event	Review and confirm publication using the shared account without supplying your own calendar key. Availability depends on the enabled program.

Open the exact event's **Manage > Luma Integration** tool. Its available controls and current connection status determine which mode you can use. Changing an account preference for new events does not migrate every existing connection.

Manual distribution without a chapter Plus key

Choose **Manual link**, copy the prepared title, description, image and AIT registration URL into the Luma page, then paste the published Luma event URL back into AIT and select **Save manual link**. Review the Luma page so visitors are directed to complete registration in AIT. If the Luma page already exists, reuse it rather than creating another.

This is a distribution workflow. Guest registration imports, periodic synchronization and automatic link repair require their applicable connected/API mode. Switching an API-connected event to manual mode can pause its sync, so read the confirmation before changing it.

Connect your chapter’s Luma Plus calendar

- Ask HQ to enable **custom Luma API support** for your chapter. Existing enabled chapters retain their access.
- Confirm the Plus subscription is attached to the chapter’s Luma calendar, not only to an organizer’s personal calendar. Luma applies Plus and API access at the calendar level.
- Create and publish the event in AI Tinkerers first. Complete the title, date, time, public summary, image, location policy, screening settings, and RSVP page before making its Luma page.
- Decide whether this event already has a Luma page. Do not create a duplicate.

HQ-managed availability

When HQ confirms this mode is available for your chapter and event, use **Review Luma publication** and review the exact destination account, page, registration handling and provider notifications before confirming. A chapter using the shared HQ account does not need its own key. A control described here is not a promise that every chapter has this mode enabled; ask HQ if it is not offered.

VIP dinners and private/invitation-only events cannot publish to Luma. Creating an AIT event never automatically creates a Luma page. Existing Luma pages, custom keys, and manual/CSV imports remain connected as before.

Chapter account preferences affect new connections only. In **Settings → Meetups → Luma account for new events**, choose the chapter-owned account to use the custom setup below. Custom selection requires the separate HQ permission. An invalid custom key does not fall back to HQ.

HQ-managed pages have their own controls to close registration, review reopening and prepare permanent removal where enabled. Review the final-sync and Luma-notification warnings before confirming removal. These operations preserve the AIT event and registrations and do not apply to existing custom or no-key integrations.

HQ configures the shared calendar’s webhook once under **Admin → HQ Luma settings → Shared HQ webhook setup**. Chapters using HQ do not need to configure their own webhook. A **Yes** beside the secret status confirms that a secret is saved, while delivery must still be verified with Luma. Chapters using their own paid account should follow the separate webhook instructions below.

Connect the calendar

1. In Luma, open the chapter calendar and go to **Settings → Developer**. Copy its calendar-specific API key. Do not send the key by email or chat, paste it into a shared document, or put it in source control.
2. In the AI Tinkerers event, open **Manage → Luma Integration** and choose **Luma API key**.
3. Paste the API key. Normally leave **Override calendar identifier** blank; AI Tinkerers detects the calendar when the key is tested.
4. If this event already exists in Luma, paste its `evt-...` Event ID from the Luma event’s Manage URL. If no Luma page exists, leave the Event ID blank.

5. Keep **Sync event registrations from Luma** on. Luma registrants will enter AI Tinkerers as pending registrations and still need to complete the AI Tinkerers profile and screening flow.
6. Leave **Sync subscribers from Luma** off unless the chapter has a clear, consent-based reason to add those people to its AI Tinkerers email list.
7. Add only the organizers who genuinely need Luma event-management access under **Trusted Luma manager emails**, one address per line.
8. Click **Test API connection**. Continue only when the page says the key is valid, then click **Save settings**.

Create or attach the event

- **No Luma page yet:** review the displayed event and destination account, then confirm **Create new Luma page** once. AI Tinkerers will create it from the event content and save the returned Event ID.
- **Luma page already exists:** do not create another one. Save its Event ID, then use **Check now** to verify that the Luma page points visitors to the AI Tinkerers registration page. Use **Add AIT link to Luma page** if the check finds the link missing.

Before promoting the event, open the public Luma page and confirm that its primary registration direction leads people to the AI Tinkerers event page. A Luma RSVP by itself is not event acceptance and does not grant venue details or a valid AI Tinkerers check-in QR code.

Turn on real-time updates

1. After saving the API connection, copy the **Luma webhook endpoint URL** shown in AI Tinkerers.
2. In Luma, return to **Settings** → **Developer** → **Webhooks** and create a webhook with that endpoint.
3. Select **Guest Registered** and **Guest Updated**, or **All Actions** if that is simpler for the chapter.
4. Luma will show a signing secret beginning with `whsec_`. Paste it into **Webhook signing secret** in AI Tinkerers and save it. Treat it like the API key and never share it publicly.

After launch

- Use **Sync now** after setup and review the Sync and Import report.
- Review and approve or waitlist attendees in AI Tinkerers, not Luma. AI Tinkerers sends the authoritative acceptance and QR code.
- If the report flags an unexpected calendar admin, review that person directly in Luma. AI Tinkerers can flag calendar admins but cannot demote them through Luma's API.
- If the API key is replaced, the Plus subscription moves to another calendar, or the integration reports an error, stop creating or syncing pages until the connection tests successfully again.

Paused sync and incomplete imports

A saved API key, a valid connection and active guest sync are different states. If the page says **Sync paused**, use the available **Resume sync** flow to validate the saved connection. For **API validation failed**, review the displayed error, edit/test the correct calendar key and confirm the sync setting. Use the Sync and Import report to check the last successful run and any remaining errors.

For supported CSV imports, review the detected columns before importing. LinkedIn/GitHub links can come from recognized custom-question columns, and existing profile links are preserved. A fixed importer does not automatically backfill missing links from an earlier import. Include the import/event reference and affected field in private support when historical data needs checking.

Do not share API keys or webhook secrets in a support post. A stale sync time does not by itself show whether the key, subscription, connection mode or a product defect caused the problem.

Luma references: API keys, webhooks, Luma Plus, and calendar ownership and transfers.