

Data platform setup checklist

Companion to the tutorial "Own Your Association's Data Platform"

(<https://associationai.ai/posts/own-your-association-data-platform/>). Print this, work through it in your Google Cloud console, and check each box as you finish. The build itself happens in your Cloud console; this page keeps you honest about finishing it.

1. Create the Cloud project

- ☐ Created with the association's Workspace account, never a staffer's personal Gmail (projects belong to their creator).
- ☐ Name is 4 to 30 characters and legible to a successor (e.g. `acme-association-data`, not `test-final-2`).
- ☐ Billing account picked "as applicable" (skip it for the sandbox in step 2).
- ☐ You hold the Project Creator role, or your IT contact created it (new Workspace and Cloud Identity domains hold the role by default).

2. Choose your on-ramp

Sandbox (the practice room)	Billing-enabled (the warehouse)
No credit card, no billing account	Card on file
10 GiB of lifetime storage	Standard free tier
1 TiB of query data per month	Everything unlocked
Tables and views auto-expire at 60 days	No expiry
No streaming inserts or DML	GA4 export and data agents need this

- ☐ Choice recorded: _____
- ☐ Reason recorded: _____ (The sandbox is for learning. Enable billing the week you connect real member data or build the data agent.)

3. Pick one branch

- ☐ Staff only: dashboards and plain-English answers on your own numbers. (Branch one: staff dashboards.)
- ☐ Members, reading: chapters see curated data without copying it. (Branch two: member data sharing.)
- ☐ Members, using: an app members open, like a data explorer or benchmarking tool. (Branch three: the member platform.)

Branch chosen: _____

The branches keep, so one never blocks the others. Pick one and finish it.

4. Build the branch you chose

Branch one (staff dashboards):

- ☐ Billing enabled on the project (data agents require it).
- ☐ Data agent built following the companion tutorial "Build the Data Agent Your Dashboard Chats With".
- ☐ Looker Studio selected as a publishing option; staff can open the chat page and ask questions against your BigQuery tables.

Branch two (member data sharing):

- ☐ Data exchange created in Analytics Hub.
- ☐ Listing added for the dataset; pilot subscriber invited.
- ☐ Authorized views or datasets in place; column- and row-level security set where needed.
- ☐ Egress restrictions reviewed so shared data cannot be copied, cloned, exported, or duplicated with CREATE TABLE AS. (Up to 1,000 linked datasets per shared dataset, at no additional cost for managing exchanges or listings.)

Branch three (member platform):

- ☐ AI Studio Build mode opened; the app described in plain language.
- ☐ React frontend and Node.js server runtime generated; Firebase Firestore and Authentication provisioned; API key kept as a server-side secret (never in client-side code).
- ☐ Deployed to Cloud Run; BigQuery warehouse connected behind the app.
- ☐ Costs reviewed: Cloud Run pricing may apply based on usage; the Firebase no-cost quotas cover most association volumes (50,000 monthly active users for Authentication; 1 GB stored with 50,000 reads and 20,000 writes per day for Firestore; 2 million function invocations per month; 10 GB of Hosting storage).

5. Result check

- ☐ The project lives on the association's Workspace account, not a personal Gmail.
- ☐ You recorded the sandbox-or-billing choice and the reason.
- ☐ One branch is finished: a dashboard staff can open, a listing a member can subscribe to, or an app a member can sign into.
- ☐ A second person holds Owner access. One keyholder is a bus factor.
- ☐ You reviewed who can export what: egress restrictions on shared data, no API keys in client-side code.

6. Mistakes to avoid

- ☐ The personal-Gmail project. The most common mistake and the hardest to unwind; migrate now while the project is small.
- ☐ Building all three branches at once. The branches keep; finish one, then expand.
- ☐ Treating the sandbox as production. Sixty days later your tables expire mid-board-meeting. The sandbox is the practice room.
- ☐ Sharing the raw dataset instead of a listing. The listing, the authorized view, and the egress block are the sharing; the dataset alone is exposure.
- ☐ Pasting the API key into frontend code. Build mode keeps it server-side for a reason; every member's browser would carry your key.

Your data, on your ground, under your keys.

From **AssociationAI.ai** — practical AI education for association staff. Nothing here is legal, financial, or compliance advice.