Google Nest Thermostat — Crestron Home Setup Guide

Zentura Home · v2.0 · https://zenturahome.com/drivers/google-nest-thermostat

---

Before you start

Nest devices are controlled through Google's **Smart Device Management (SDM) API**. Each installation uses its own free Google Cloud project plus a one-time **$5 Device Access** registration fee paid directly to Google. Once configured, the driver runs unattended; your thermostats appear as native Crestron Home Climate tiles and react to scene changes, schedules, and Sara voice commands in real time.

You'll need:

- A **personal `@gmail.com` account** (Google Workspace accounts are not supported)
- A **$5 one-time fee** paid directly to Google (not Zentura)
- About **25 minutes** of guided setup
- One **Zentura Home Nest Thermostat license** ($129, one-time, MAC-locked)

You'll come away with:

- Each Nest thermostat appearing as a **native Crestron Home Climate tile**
- Full support for **Climate Scenes** and **scheduled events**
- **Sub-second push updates** via Google Cloud Pub/Sub (recommended) or 60s polling fallback
- Voice control through **Crestron Sara**

Step 1 — Register for Google Device Access

Go to **https://console.nest.google.com/device-access** and accept the Sandbox terms. Pay the **$5 one-time** registration fee with a credit card.

> ***Note:*** *This fee is paid directly to Google. Zentura does not receive any portion of it. You will not be*

> *charged again — the $5 grants your Google account lifetime SDM API access.*

Step 2 — Create a Device Access Project

From the Device Access Console:

1. Click **Create Project**.
2. Name the project (e.g. *Crestron Home*) and click **Next**.
3. **Skip** the OAuth Client ID prompt for now — you'll wire it up in step 6.
4. **Leave Enable Events set to Disabled** for now — you'll turn it on in step 10 once you have a Pub/Sub topic to point it at.
5. Open the project page and **copy the Project ID** (a UUID like `7a9c14ab-…`). You'll reference it several times below and again in Crestron Home in step 11.

Step 3 — Create or select a Google Cloud project

Open **https://console.cloud.google.com/projectcreate** and create a new project (any name; e.g. *Zentura Nest*). Copy the **Project ID** Google assigns — this is your *Google Cloud Project ID*, distinct from the Device Access Project ID from step 2. If you already have a personal Cloud project you'd like to reuse, you can pick that one instead.

Step 4 — Enable the required Google Cloud APIs

With your Google Cloud project selected, open **https://console.cloud.google.com/apis/library** and enable **both** of the following (search by name, then click **Enable** on each):

- **Smart Device Management API** — lets the driver read and control your thermostats.
- **Cloud Pub/Sub API** — lets the driver receive real-time mode / setpoint / humidity / connectivity updates within ~1 second of a change.

> *If you skip the Pub/Sub API the driver still works in polling-only mode and the install completes successfully — trait changes simply appear within one polling interval (default 60s) instead of within a second.*

Step 5 — Configure the OAuth Consent Screen and publish to production

1. Open **https://console.cloud.google.com/apis/credentials/consent**.
2. **User type:** `External` (the only option for personal Gmail accounts).
3. Fill in the **App name** (any name, e.g. *Crestron Home Nest*), **User support email**, and **Developer contact email** (your own).
4. Add **both** scopes:
   - `https://www.googleapis.com/auth/sdm.service`
   - `https://www.googleapis.com/auth/pubsub`
5. **Publishing status → In production.**

> ⚠️ ***Critical step.*** *If you leave the consent screen in Testing mode, Google expires refresh tokens after **7 days** and the driver stops working. Publishing is free; personal-use apps with only the SDM and Pub/Sub scopes are not subject to Google's brand-review process.*

Step 6 — Create an OAuth 2.0 Client ID and link it to the Device Access project

1. Go to **https://console.cloud.google.com/apis/credentials** (make sure the project selector at the top shows the Cloud project from step 3).
2. Click **Create Credentials → OAuth client ID**.
3. **Application type:** `Web application`

4. Under **Authorized redirect URIs** add **exactly**:

   ```
   https://zenturahome.com/oauth/nest/callback.html
   ```

   (This is a static Zentura-hosted page that simply displays the auth code in your browser; the code never leaves your computer.)

5. Click **Create**. Copy the **Client ID** and **Client Secret** that Google displays. Treat the secret like a password.

6. Return to the **Device Access Console** (the project from step 2) and paste the new **Client ID** into the **OAuth Client ID** field on the project page. (If you skipped this in step 2, this is where you wire it up.)

Step 7 — Authorize your Nest devices via the Zentura sign-in page

Open **https://zenturahome.com/oauth/nest/callback.html** in a new browser tab. The page contains a small form:

1. Paste your **SDM Project ID** (from step 2).
2. Paste your **OAuth Client ID** (from step 6).
3. Leave **Enable real-time events** *checked* (default).
4. Click **Authorize with Google**.
5. Sign in with the **same Google account** that owns your Nest devices.
6. Tick the **box next to each thermostat** Crestron Home should control.
7. Click **Next** → **Allow**.

Google redirects you back to the same callback page.

Step 8 — Copy the Authorization Code

On the callback page, a single-use **Authorization Code** appears.

- Click **Copy** within **5 minutes** — codes expire quickly.
- If you see an error banner instead, return to step 7 and try again.

> ⚠️ ***Verify the consent screen showed BOTH scopes.*** *During the authorization flow Google should have asked you to grant:*
>
> - *"See and/or control the devices that you selected" (the SDM scope), AND*
> - *"View and manage Pub/Sub topics and subscriptions" (the Pub/Sub scope)*
>
> *If you only saw the SDM permission, the* `pubsub` *scope is **not** on your OAuth Consent Screen (step 5). Add it there and re-do steps 7–8 — otherwise the driver will hit* `403 insufficient scope` *errors on every Pub/Sub pull.*

Step 9 — Create a Pub/Sub topic and subscription in your Google Cloud project

> ***Optional but strongly recommended.*** *Skip this step (and step 10) only if you want polling-only operation. With Pub/Sub configured the driver receives mode / setpoint / humidity changes from the thermostat within ~1 second instead of waiting for the next poll.*

Unlike most Google APIs, SDM does **not** publish events to a topic in its own project that you can subscribe to. Instead, **you create the topic in your own GCP project**, register it in the Device Access Console (step 10), and Google then publishes events to it. The driver subscribes to that same topic.

In your Google Cloud project, open **https://console.cloud.google.com/cloudpubsub/topic/list** and click **Create Topic**.

1. **Topic ID:** `nest-events` (or anything memorable).
2. **Add a default subscription:** leave this **checked** — Google will auto-create `nest-events-sub` as a Pull subscription on the topic, which is exactly what the driver needs.
3. Leave all other defaults. Click **Create**.

From the topic detail page, copy **both** of the following names — you'll need each one in a different later step:

```
Topic:        projects/{your-gcp-project-id}/topics/nest-events
Subscription: projects/{your-gcp-project-id}/subscriptions/nest-events-sub
```

- The **topic** name goes in the Device Access Console (step 10).
- The **subscription** name goes in Crestron Home (step 11).

Step 10 — Register the Pub/Sub topic in the Device Access Console

Return to https://console.nest.google.com/device-access and open your project. Set:

- **Enable Events:** `Enabled`
- **Pub/Sub topic:** paste the topic name from step 9, e.g. `projects/{your-gcp-project-id}/topics/nest-events`

Click **Add & Validate** (or **Save**). The console verifies that Google's SDM service account can publish to your topic and, if needed, grants itself the `Pub/Sub Publisher` role automatically.

> ***If validation fails*** *with a permissions error: open the topic in* ***https://console.cloud.google.com/cloudpubsub/topic/list****, click the topic, go to the* ***Permissions*** *tab → **Grant Access**, paste* `sdm-publisher@googlegroups.com` *as the principal (this is a Google-managed group, not a service account — the Device Access docs use this group to represent the SDM publisher identity), choose role* `Pub/Sub Publisher`*, click Save, then retry **Add & Validate** in the Device Access Console.*

Step 11 — Pair the driver in Crestron Home

1. In the Crestron Home setup app open **Settings** → **Drivers** → **Add Driver**.

2. Pick **Nest Thermostat (by Zentura Home)**.

3. Paste the values you collected:

Field	Value source
Project ID	Step 2
OAuth Client ID	Step 6
OAuth Client Secret	Step 6
Authorization Code	Step 8
Pub/Sub Subscription Path	Step 9 — the *subscription* name *(optional)*

4. (Optional) Paste your **Zentura Home License Key**.

5. Tap **Apply**.

The driver exchanges the code for a long-lived refresh token, discovers your thermostats, and adds a **native Climate tile** for each one. The *Authorization Code* field clears itself automatically — it's only needed once. From this point forward the driver runs unattended; if you ever revoke access in Google or change OAuth credentials, simply repeat steps 7 → 11.

Troubleshooting

Symptom	Fix
`invalid_grant` after pairing	The Authorization Code expired or was already used. Repeat steps 7–8 to get a fresh code, then re-apply the driver.
Thermostats disappear after 7 days	Your OAuth consent screen is still in *Testing*. Return to step 5 and publish to *In production*.
Add & Validate fails in Device Access Console with `PERMISSION_DENIED`	Grant the SDM service account `Pub/Sub Publisher` on your topic manually. See the note at the end of step 10.

`403 PERMISSION_DENIED` or `insufficient scope` on Pub/Sub pull	The OAuth token Google issued does not include the `pubsub` scope. **Most common cause:** the `pubsub` scope is not listed on your OAuth Consent Screen (step 5). Add it under **Scopes for Google APIs**, click **Save and Continue**, then **re-pair the driver from scratch** (re-do steps 7–8 — a token's scope set is fixed at issue time and can only be changed by re-authorizing). Verify the consent screen shows you a checkbox for "*View and manage Pub/Sub topics and subscriptions*" during the authorization flow — if not, the scope still isn't registered.
Driver pairs but no real-time updates	Verify the topic name in the Device Access Console **exactly matches** the topic from step 9 (not the subscription). The subscription name (ending `.../subscriptions/...`) goes in Crestron Home, the topic name (ending `.../topics/...`) goes in the Device Access Console.
Subscription path rejected by driver	It must be the full `projects/{cloud}/subscriptions/{name}` form, not just the subscription ID. Copy it from the subscription detail page.
`429 Too Many Requests` in driver logs	Increase **Polling Interval** in driver settings (default 60s, max 300s). Pub/Sub updates are unaffected.
Setpoint changes briefly revert	Nest is performing an internal sync; the next push (or poll) will reflect the new value.
Can't see thermostat in Crestron Home Climate Scenes	Confirm the thermostat appears in the Crestron Home app **Climate** room before building scenes.

Support

- Documentation: https://zenturahome.com/drivers/google-nest-thermostat
- Contact: https://zenturahome.com/contact