

Programmatic Solution to Firebase Auth “CONFIGURATION_NOT_FOUND”

1. Context & Error

- **Issue:** When trying to enable Email/Password sign-in by PATCHing `identitytoolkit.googleapis.com/admin/v2/projects/.../config`, the error `CONFIGURATION_NOT_FOUND` is returned.
- **Root Cause:** The project's Firebase Authentication / Identity Platform config was never fully initialized, so the Identity Toolkit Admin API has no record of that config.

2. Key Requirements

1. Enable Required APIs

- `identitytoolkit.googleapis.com` (Identity Toolkit / Identity Platform API).
- If you need the full Firebase suite, also enable `firebase.googleapis.com` (Firebase Management API).

2. Have Correct IAM Permissions

- For initialization: `firebaseauth.configs.create` (often included in “Project Editor” or “Firebase Auth Admin”).
- For updates (enabling providers): `firebaseauth.configs.update`.

3. Billing

- If using Identity Platform's `initializeAuth`, the project must have a **billing account** (typically a Blaze plan).

3. Two Ways to Initialize Auth

Option A: Add Firebase (Firebase CLI or REST)

- Command: `firebase projects:addfirebase <PROJECT_ID>` (or `projects.addFirebase` via REST).
- This normally sets up the default Auth config automatically.
- If the project already “has” Firebase or an org policy blocks new service accounts, this might fail, or it might say `ALREADY_EXISTS`.

Option B: Call `initializeAuth` Directly

Endpoint:

POST

`https://identitytoolkit.googleapis.com/v2/projects/<PROJECT_ID>/identityPlatform:initializeAuth`

-
- Requires a billing-enabled project and correct IAM permissions.
- Returns `200` on success or an error if it’s already enabled or if billing is missing.

4. Enable Email/Password

After a successful Auth initialization, run:

PATCH

`https://identitytoolkit.googleapis.com/admin/v2/projects/<PROJECT_ID>/config?updateMask=signIn.email`

```
{
  "signIn": {
    "email": {
      "enabled": true,
      "passwordRequired": true
    }
  }
}
```

- This sets `"enabled": true` for email/password sign-in.
- Any subsequent calls to create users with email/password (e.g., `accounts:signUp`) will succeed.

5. Common Pitfalls

- **Project Already in Firebase** but missing Auth config.
 - `firebase projects:addfirebase` might fail with `ALREADY_EXISTS`.
 - If so, just use `initializeAuth`.
- **Org Policy** blocking new service account creation.
 - You'll need an admin to allow it, or use `initializeAuth` if you only need Auth.
- **Need to pass a "quota project"** header in Cloud Shell.
 - The Identity Toolkit v2 API often requires `-H "X-Goog-User-Project:<PROJECT_ID>"` or impersonating a service account that belongs to that project.
- **Already Enabled:** The `initializeAuth` call can return a 400 with `"INVALID_PROJECT_ID : Identity Platform has already been enabled for this project"`.
 - Treat that as success in your script.

6. Example Pipeline Fix

In a CI/CD script (like a Cloud Build YAML), you can:

1. **Retrieve the project ID.**
2. **Generate an access token** (`gcloud auth print-access-token`).
3. **Initialize Identity Platform** with a POST to `.../identityPlatform:initializeAuth`.
 - If you see `ALREADY_ENABLED` or `INVALID_PROJECT_ID`, skip.
4. **Patch the config** to enable Email/Password sign-in.
5. **Verify** by creating a test user via `POST /v1/accounts:signUp?key=YOUR_WEB_API_KEY`.