DayBoard External Setup Instructions
==========

Overview

Before running the DayBoard application, you need to set up accounts and projects with several external services. These services provide the calendar data, subscription detection, commute estimates, and hosting necessary to support the app. This document outlines the steps required to create these accounts and gather the credentials you will later add to your `.env` file.

Services to Configure
---

1. **Supabase (PostgreSQL database hosting)**
   - Sign up at https://supabase.com/ using your preferred sign-in method (email or social login).
   - Create a new project named "dayboard" (or any name you prefer).
   - Choose the free tier if available and select a strong database password.
   - Once the project is created, go to "Project Settings" → "Database" and note down the **Connection String** (formatted like `postgres://username:password@host:port/dbname`). This string will be used as `DATABASE_URL` in your backend's `.env` file.

2. **Google Cloud (Calendar API and Distance Matrix)**
   - Visit https://console.cloud.google.com/ and sign in with your Google account.
   - Create a new project called "DayBoard".
   - In the APIs & Services dashboard, enable the following APIs:
     * Google Calendar API
     * Google Maps Distance Matrix API
   - Under "Credentials", create an **OAuth client ID** of type **Web Application**. Set the authorised redirect URI to `https://<your-backend-domain>/auth/google/callback` (replace with your deployed backend domain or local URL during development).
   - Copy the **Client ID** and **Client Secret**; these will populate `GOOGLE_CLIENT_ID` and `GOOGLE_CLIENT_SECRET` in your `.env` file.
   - Also create an **API key** for the Distance Matrix API and note it for `DISTANCE_MATRIX_API_KEY`.

3. **Microsoft Azure (Outlook Calendar API)**
   - Navigate to https://portal.azure.com/ and sign in.
   - Go to **Azure Active Directory** → **App registrations** and click **New registration**.
   - Name your app (e.g. "DayBoard"), choose **Accounts in any directory**.
   - Under "Redirect URI", select **Web** and enter `https://<your-backend-domain>/auth/microsoft/callback`.
   - After registration, note the **Application (client) ID** and create a **Client Secret** under **Certificates & secrets**. These values become `MICROSOFT_CLIENT_ID` and `MICROSOFT_CLIENT_SECRET`.
   - Configure API permissions: add delegated permission to `Calendars.Read`.

4. **Plaid (Subscription detection)**
   - Go to https://dashboard.plaid.com/signup and create a developer account.
   - After verifying your email, create a new application and choose the **Sandbox** environment for development.
   - Copy your **client_id** and **secret**; populate `PLAID_CLIENT_ID` and `PLAID_SECRET` in your `.env` file. Set `PLAID_ENV=sandbox`.
   - You do not need production access for development.

5. **Hosting Platform (Fly.io, Render, Railway, or Cloud Run)**
   - Choose a platform to deploy your Go backend. This guide assumes Fly.io.
   - Sign up at https://fly.io/ and install the Fly CLI.

   - Run `fly launch` in your backend directory when ready to deploy.
   - Note your app’s URL (e.g., `https://dayboard.fly.dev`) for use as your OAuth redirect base.
   - Set environment variables (`DATABASE_URL`, `GOOGLE_CLIENT_ID`, etc.) on your Fly app using `fly secrets set` or the web dashboard.

6. **Apple Developer Account (Optional)**
   - For iOS distribution and TestFlight, you need an Apple Developer Program membership ($99/year). Enroll at https://developer.apple.com/ if you plan to distribute beyond your own devices.

After you complete the account setup, update the `.env` file in `dayboard/backend` with the values you collected. Do not commit secret keys to source control. Instead, use environment variables or deployment secret management provided by your hosting platform.