

✓ Google API Setup for Laravel – Step-by-Step Roadmap

❖ login.php.blade/signup.php.blade.php

```
• <a href="{ route('google.login') }" >
• <imgsrc="https://developers.google.com/identity/images
  /btn_google_signin_dark_normal_web.png"
• alt="Sign in with Google">
• </a>
```

❖ Route

```
•
• Route::get('/auth/redirect/google', [UserController::class, 'redirectToGoogle'])->name('google.login');
• Route::get('/auth/callback/google', [UserController::class, 'handleGoogleCallback']);
•
```

❖ Controller

```
• use Laravel\Socialite\Facades\Socialite;
• use Illuminate\Support\Facades\Auth;
• // *****GOOGLE AUTHENTICATION*****
• public function redirectToGoogle()
• {
•     return Socialite::driver('google')->redirect();
• }
•
• public function handleGoogleCallback()
• {
•     try {
•         $googleUser = Socialite::driver('google')->stateless()->user();
•
•         // Check if user already exists
•         $user = User::where('email', $googleUser->getEmail())->first();
•
•         if (!$user) {
•             // Create new user
•             $lastuser = User::orderBy('id', 'desc')->first();
•             $number = $lastuser->id + 1000;
•             $userid = 'IRCTC' . $number;
•
•             $user = User::create([
•                 'name' => $googleUser->getName(),
•                 'email' => $googleUser->getEmail(),
•                 'userid' => $userid,
•                 'password' => bcrypt('google_dummy_password'),
•             ]);
•         }
•         Auth::login($user);
•
•         return redirect('dashboard')->with('success', 'Google Login Successful! Welcome: ' .
• $user->name);
•     } catch (\Exception $e) {
•         return redirect('/login')->with('error', 'Google Login Failed: ' . $e->getMessage());
•     }
• }
```

✓ 1. Install Laravel Socialite

If you haven't already:

```
Bash cmd ~  
composer require laravel/socialite
```

✓ 2. Update config/services.php

Add this:

```
'google' => [  
    'client_id' => env('GOOGLE_CLIENT_ID'),  
    'client_secret' => env('GOOGLE_CLIENT_SECRET'),  
    'redirect' => env('GOOGLE_REDIRECT_URI'),  
],
```

✓ 3. And in your .env file, add:

```
GOOGLE_CLIENT_ID=your_google_client_id  
GOOGLE_CLIENT_SECRET=your_google_client_secret  
GOOGLE_REDIRECT_URI=http://127.0.0.1:8000/auth/google/callback
```

👉 For generate **Client ID & Client Secret**

✓ Step 1: Go to Google Cloud Console

🔗 Visit: <https://console.cloud.google.com/>

✓ Step 2: Create a New Project (if not already)

1. Click on the **dropdown** in the top-left corner (next to “Google Cloud” logo).
2. Click **"New Project"**.
3. Give it a **name** (e.g., RailSeva Google Login) and click **"Create"**.
4. After creation, make sure to **select your new project**.

✓ Step 3: Enable the Google+ / OAuth2 API

1. In the sidebar, go to **"APIs & Services" > "Library"**.
2. Search for **"Google People API"** (for profile data).
3. Click it, then click **"Enable"**.

✓ Step 4: Create OAuth 2.0 Credentials

1. Go to **"APIs & Services" > "Credentials"**.
2. Click **"Create Credentials" > "OAuth client ID"**.

If you get a popup to configure a consent screen:

- Click **"CONFIGURE CONSENT SCREEN"**
- Choose **"External"**, click **Create**.
- Fill basic info: App Name, User Support Email.
- Scroll down and click **Save and Continue** (you can skip Scopes for now).
- On the last screen, click **Back to Dashboard**.

3. Now again click "**Create Credentials**" > "**OAuth client ID**".

✅ **Step 5: Choose Application Type**

- Select "**Web application**"
- Name it something like Laravel Login

Under "**Authorized redirect URIs**", add this:

text

CopyEdit

`http://127.0.0.1:8000/auth/google/callback`

👉 (You can change this to your domain later in production.)

✅ **Step 6: Get Client ID and Secret**

- After submitting, you will see:
 - **Client ID**
 - **Client Secret**

Click "**Download JSON**" or copy-paste these into your .env:

env

CopyEdit

`GOOGLE_CLIENT_ID=your-client-id-here`

`GOOGLE_CLIENT_SECRET=your-client-secret-here`

`GOOGLE_REDIRECT_URI=http://127.0.0.1:8000/auth/google/callback`
