

## GuideSense: API Documentation

### 1. Authentication APIs

#### 1.1 Registration API

Purpose: Creates a new user account with email, password, and additional user details via Clerk.

Method: POST using Clerk via Full stack SDK

Process:

1. Input: User having two options to sign-up their account. Either manual enter the username, email, and password; or sign up by link to their Google account.
2. Output: A new user account is created in Clerk Authentication, and corresponding user details are stored in Clerk

Key Code: [Users: Create users](#)

```
export async function POST() {
  try {
    const user = await clerkClient.users.createUser({
      emailAddress: ['test@example.com'],
      password: 'password',
    })
    return Response.json({ message: 'User created', user })
  } catch (error) {
    console.log(error)
    return Response.json({ error: 'Error creating user' })
  }
}
```

## 2.2 Login API

Purpose: User able to login their accounts on two options.

Method: Enable the login options on Clerk API

Process:

1. Login with their username or email with password.

The image shows three panels of the Clerk API settings interface. The first panel, 'Contact information', has a description 'Specify whether your users should have email addresses or phone numbers'. It contains two toggle switches: 'Email address' (checked) and 'Phone number' (unchecked). The 'Email address' toggle has sub-options: 'Required' (checked), 'Used for sign-in' (checked), 'Verify at sign-up' (unchecked), and 'Email verification code' (unchecked). The 'Phone number' toggle has a 'Pro' label. The second panel, 'Username', has a description 'Specify whether your users have a unique username'. It contains one toggle switch: 'Username' (checked), with sub-options 'Required' (checked) and 'Used for sign-in' (checked). The third panel, 'Authentication strategies', has a description 'Select the authentication methods to present when a user signs in' and a link 'Learn more about authentication strategies'. It contains five toggle switches: 'Password' (checked), 'Passkeys' (unchecked, with a 'Pro' label), 'Email verification link' (unchecked), 'Email verification code' (unchecked), and 'SMS verification code' (unchecked). The 'SMS verification code' toggle has a message: 'Cannot be enabled because Phone number contact information is not used for sign-in'.

**Contact information**  
Specify whether your users should have email addresses or phone numbers

☒ **Email address**  
Users can add email addresses to their account  
☒ Required ☒ Used for sign-in ☐ Verify at sign-up ☐ Email verification code

☐ **Phone number** Pro  
Users can add phone numbers to their account

**Username**  
Specify whether your users have a unique username

☒ **Username**  
Users can set usernames to their account  
☒ Required ☒ Used for sign-in

**Authentication strategies**  
Select the authentication methods to present when a user signs in  
[Learn more about authentication strategies](#)

☒ **Password**  
Users can sign in with a password. Passwords are required during sign up unless the user signs up with a social connection or a Web3 wallet.  
☐ 8+ characters ☐ Reject compromised on sign-up & sign-in

☐ **Passkeys** Pro  
Users can sign in with a passkey.

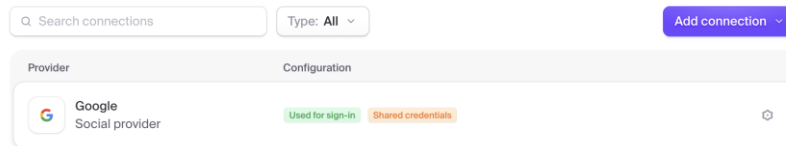
☐ **Email verification link**  
Users can sign in with an email verification link

☐ **Email verification code**  
Users can sign in with an email verification code

☐ **SMS verification code**  
Users can sign in with an SMS verification code  
⚠ Cannot be enabled because Phone number contact information is not used for sign-in

2. Login with “Continue with Google”, which requires user to login to their Google account.

#### SSO connections



## 2. FastAPI

<https://fastapi.tiangolo.com/reference/openapi/docs/>

To see documentation in Swagger UI, add /docs to the end of the URL.

To see documentation in ReDoc, add /redoc to the end of the URL.

## 3. Google Maps

- Go to the Google Cloud Console.
- Click the project dropdown (top bar) → "New Project".
- Name your project and click **"Create"**.
- Go to the API Library
- Search for and enable the following APIs depending on your needs:  
Maps JavaScript API Geocoding API, Directions API, Places API

## 4. Google Text to Speech

- Go to the Google Cloud Console.
- Visit the API Library.
- Search for “Text-to-Speech API”.
- Click Enable.
- Go to Credentials.

- Click "**Create Credentials**" → "**Service Account**".
- Fill in name/description and click **Create and Continue**.
- Grant the "**Project → Editor**" role (or more specific roles if needed).
- Click **Done**.
- In the credentials list, click your new service account → "**Keys**" tab → **Add Key** → **Create new key** → **JSON**.
- A .json key file will download. **Save this securely** — it's your authentication token,