

⚠️ No hay “login directo” vía API

Si tu caso es que **quieres que tus usuarios inicien sesión y vean sus datos de ClickUp**, necesitas implementar el **flujo OAuth2** para que ClickUp les pida el login y te devuelva un token.

¿Quieres que te ayude a armar un flujo básico de OAuth2 para ClickUp? 🚀 O si lo que buscas es cómo estructurar las llamadas una vez que tienes el token, ¡también te ayudo!

Tú dijiste:

si quisiera crear una aplicación que hiciese de intermediario entre clickup un un usuario que ya tiene cuenta en clickup, voy a necesitar generar mi base de datos de usuarios que incluyan sus datos de logeo en mi aplicacion, y ademas el token del usuario que usa para gestionar clickup. Quiero saber si esta forma de proceder es correcta y factible o por el contrario no es posible llevarla a cabo. Y si fuese posible, como lo haría, incluyendo como obtengo el token del usuario para guardarlo en mi base de datos .

ChatGPT Plus

¡Buena pregunta! Vamos a analizarlo paso a paso para que veas si tu idea es viable y cómo la llevarías a cabo. 🚀

🎯 Tu objetivo

Quieres que tu app:

- ✓ Gestione usuarios propios (con su propio login en **tu app**).
- ✓ Permita a esos usuarios interactuar con **ClickUp** (con su cuenta de ClickUp).

👉 Para que eso funcione, necesitas:

- 1) Una **base de datos de usuarios** de **tu app**.
 - 2) El **token de acceso a ClickUp** de cada usuario.
 - 3) Un sistema para **obtener** ese token de forma segura (OAuth2).
-

⚠️ ¿Es factible?

- ✓ **Sí**, es exactamente el modelo de la mayoría de integraciones con APIs externas.
 - ✓ Se usa un flujo **OAuth2** estándar para que el usuario autorice a tu app a acceder a su cuenta de ClickUp.
 - ✓ Tu app **NO necesita** las credenciales de ClickUp del usuario (ni email ni contraseña de ClickUp).
-

💡 Cómo se hace paso a paso

1 Tu base de datos de usuarios

Crea un modelo de usuario en tu app, por ejemplo:

json

```
{
  "_id": "usuario123",
  "email": "usuario@email.com",
  "passwordHash": "hash_de_su_contraseña",
  "clickup": {
    "access_token": "token_de_clickup",
    "refresh_token": "refresh_token_si_hay",
    "team_id": "opcional",
    "expires_at": "opcional"
  }
}
```

 **NUNCA guardes contraseñas de ClickUp** en tu base de datos.

2 Flujo OAuth2 para obtener el token

ClickUp usa OAuth2 estándar. Te doy el flujo completo:

a) Registra tu app en ClickUp

- Ve a: ClickUp OAuth Apps
- Obtendrás:
 - `client_id`
 - `client_secret`
 - URL de redirección (`redirect_uri`)

b) Redirige al usuario para que autorice

Cuando un usuario tuyo quiera “conectar su cuenta de ClickUp”, mándalo a:

`https://app.clickup.com/api?client_id=CLIENT_ID&redirect_uri=REDIRECT_URI`

✅ Ahí ClickUp le pedirá que inicie sesión y autorice a tu app.

🟡 **c) ClickUp te devuelve un código**

ClickUp redirige al usuario de vuelta a tu `redirect_uri`, así:

`https://tusitio.com/oauth/callback?code=CODE`

🟡 **d) Tu backend intercambia el código por un token**

En tu backend, haces un POST:

http

POST `https://api.clickup.com/api/v2/oauth/token`
Content-Type: `application/json`

```
{
  "client_id": "CLIENT_ID",
  "client_secret": "CLIENT_SECRET",
  "code": "CODE",
  "redirect_uri": "REDIRECT_URI"
}
```

ClickUp te devolverá:

json

```
{
  "access_token": "TOKEN",
  "token_type": "Bearer",
  "expires_in": 2592000, // segundos
  "refresh_token": "REFRESH_TOKEN"
}
```

● Aquí es donde guardas el token en tu base de datos del usuario.

3 Llamadas a la API

Cuando necesites llamar a la API de ClickUp para ese usuario:

http

```
GET https://api.clickup.com/api/v2/team
Authorization: Bearer ACCESS_TOKEN_DEL_USUARIO
```

■ ¿Qué gana el usuario?

- ✓ Puede usar tu app con su propio login.
 - ✓ Puede autorizar a tu app para interactuar con su cuenta de ClickUp, sin darte sus credenciales.
 - ✓ Tú solo usas el `access_token` de ClickUp cuando lo necesitas.
-

■ ¿Qué debes tener en cuenta?

- ✓ **Seguridad:** Guarda los tokens de ClickUp de forma segura (cifrados o con acceso restringido).
 - ✓ **Renovación de token:** Usa el `refresh_token` para obtener un nuevo `access_token` cuando expire.
 - ✓ **Cumple con la política de privacidad:** Informa al usuario que almacenarás el token de ClickUp para poder actuar en su nombre.
-

● Ejemplo: cómo guardarlo en tu backend (pseudocódigo)

js

```
app.get('/oauth/callback', async (req, res) => {
  const code = req.query.code;

  const response = await
  fetch('https://api.clickup.com/api/v2/oauth/token', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({
      client_id: CLIENT_ID,
```

```
        client_secret: CLIENT_SECRET,
        code,
        redirect_uri: REDIRECT_URI
    })
});

const tokens = await response.json();

// Actualiza el usuario actual de tu app con los tokens
await User.updateOne(
  { _id: req.session.userId }, // tu sistema de sesión
  {
    $set: {
      'clickup.access_token': tokens.access_token,
      'clickup.refresh_token': tokens.refresh_token,
      'clickup.expires_at': Date.now() + tokens.expires_in * 1000
    }
  }
);

res.redirect('/dashboard');
});
```

✅ Resumen final

- ✓ **Sí se puede y es buena práctica.**
- ✓ Usas **OAuth2** para obtener el **access_token**.
- ✓ Guardas ese token en tu base de datos, no la contraseña de ClickUp.
- ✓ Puedes entonces hacer llamadas a la API de ClickUp en nombre del usuario.