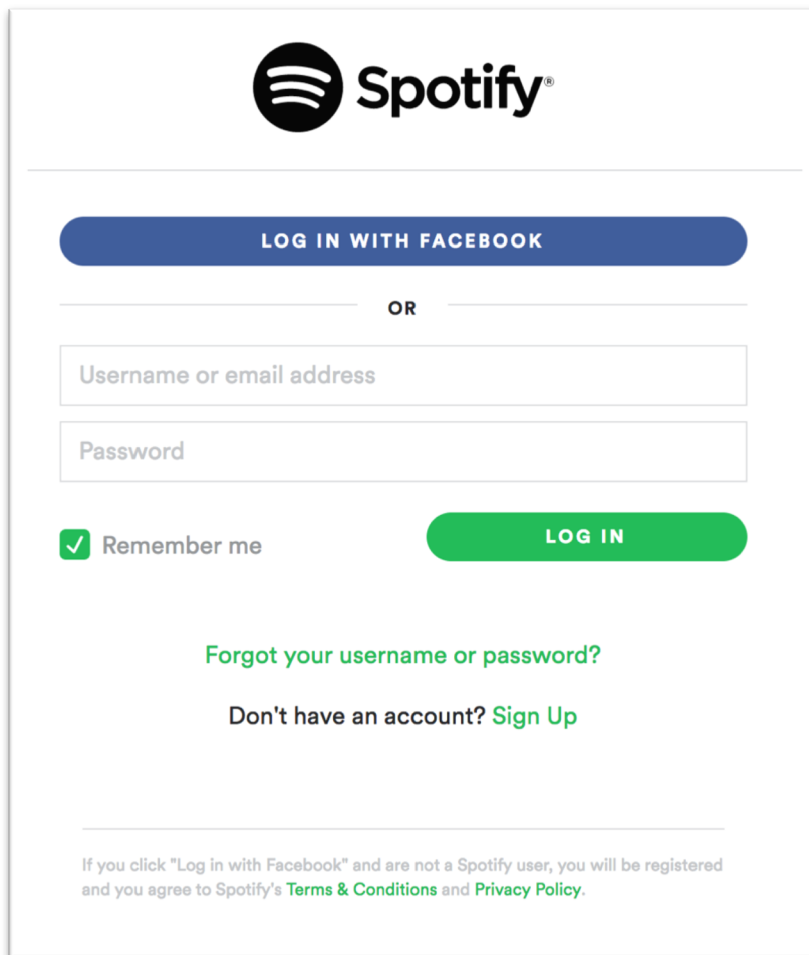





OAuth & OpenID Connect



OAuth for Sign-In



The Spotify login form features the Spotify logo at the top. Below it is a blue button labeled "LOG IN WITH FACEBOOK". Underneath this button is the word "OR" flanked by horizontal lines. There are two input fields: "Username or email address" and "Password". Below the password field is a checkbox labeled "Remember me" which is checked. To the right of the checkbox is a green "LOG IN" button. Below the login options are two links: "Forgot your username or password?" and "Don't have an account? Sign Up". At the bottom, there is a small disclaimer: "If you click 'Log in with Facebook' and are not a Spotify user, you will be registered and you agree to Spotify's Terms & Conditions and Privacy Policy."



LOG IN WITH FACEBOOK

OR

Username or email address

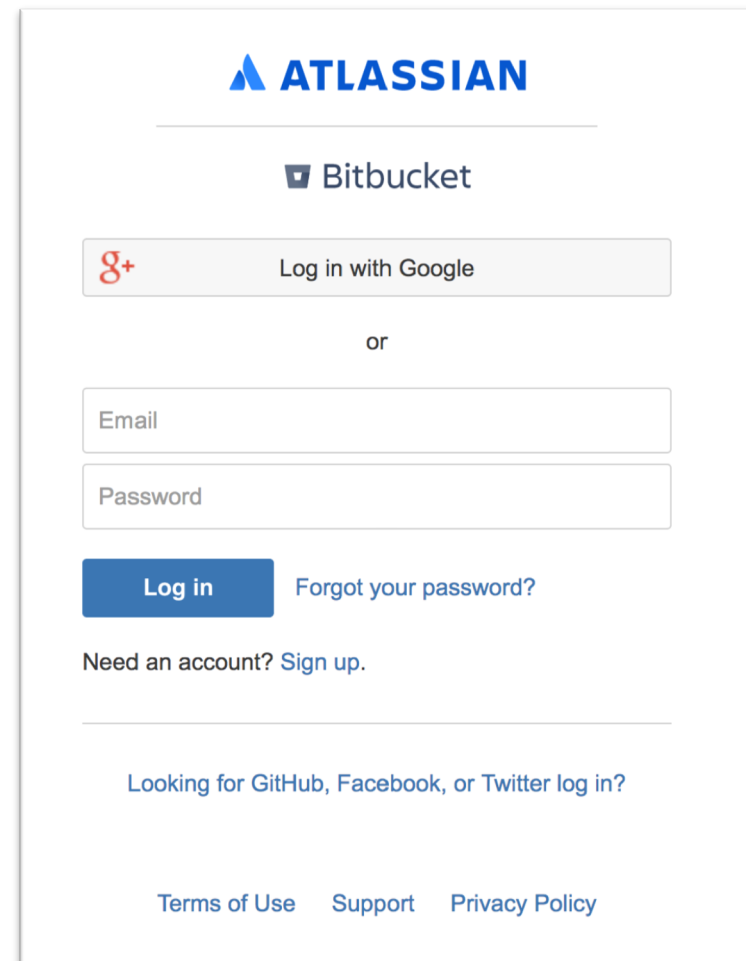
Password

Remember me **LOG IN**


[Forgot your username or password?](#)

Don't have an account? [Sign Up](#)


If you click "Log in with Facebook" and are not a Spotify user, you will be registered and you agree to Spotify's [Terms & Conditions](#) and [Privacy Policy](#).



The Atlassian Bitbucket login form features the Atlassian logo at the top. Below it is a dropdown menu showing "Bitbucket". There is a "Log in with Google" button with a Google+ icon. Below this is the word "or" flanked by horizontal lines. There are two input fields: "Email" and "Password". Below the password field is a blue "Log in" button and a link "Forgot your password?". Below the login options is a link "Need an account? Sign up.". At the bottom, there is a link "Looking for GitHub, Facebook, or Twitter log in?". At the very bottom are three links: "Terms of Use", "Support", and "Privacy Policy".



Bitbucket

 Log in with Google

or

Email

Password

Log in [Forgot your password?](#)

Need an account? [Sign up.](#)

[Looking for GitHub, Facebook, or Twitter log in?](#)

[Terms of Use](#) [Support](#) [Privacy Policy](#)

OAuth for Sign-In

Sign-In With ...

facebook

Google



LinkedIn



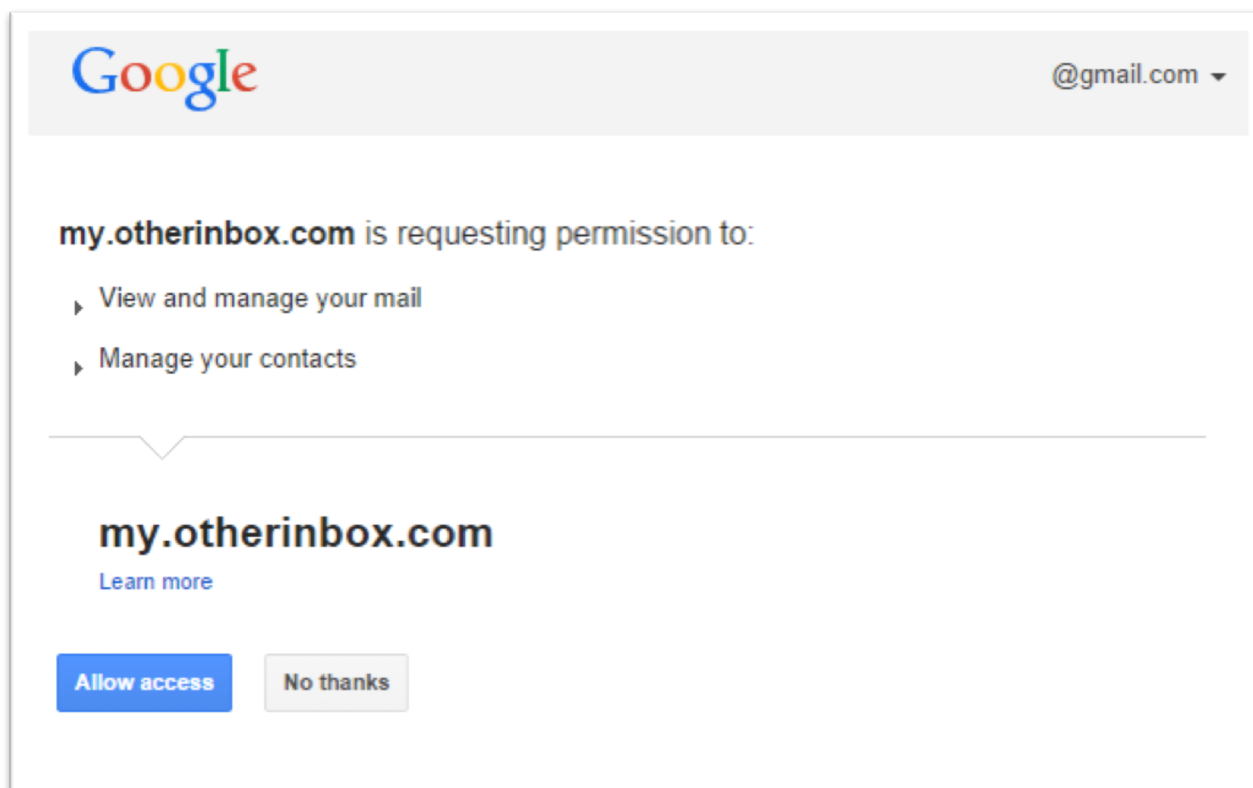
GitHub

amazon

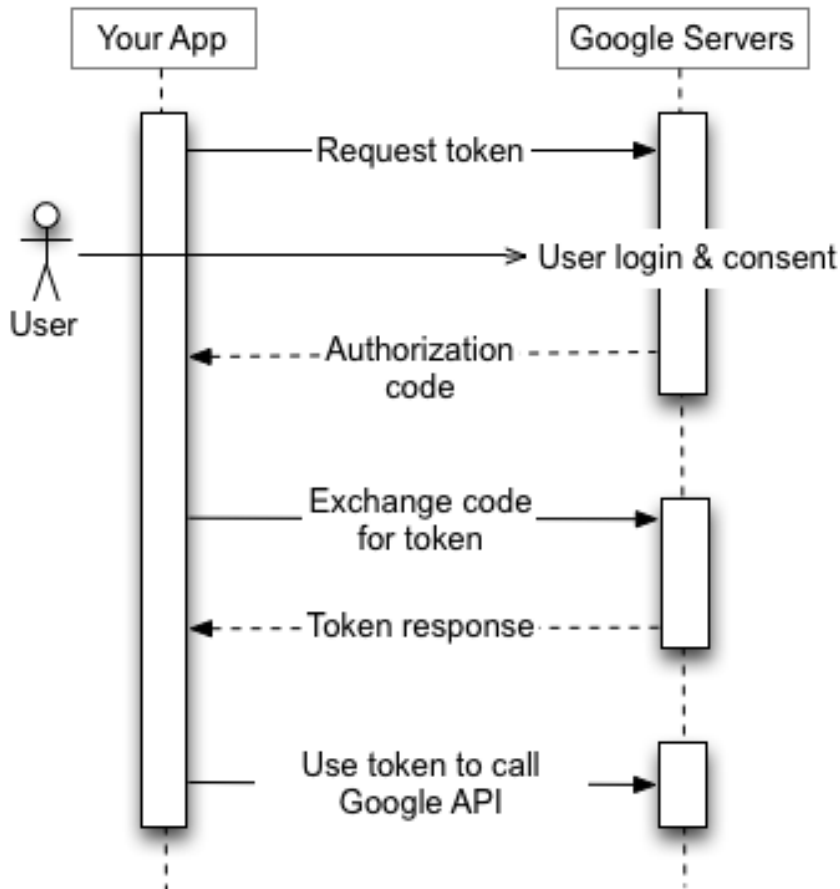
OAuth for Sign-In

- Assume: You're already logged in with Facebook (web browser has cookie)
- Facebook will give limited account information (Email, public profile, ...) to new service for account creation purposes
- Facebook password is not shared
- New service cannot post to your Facebook account

OAuth for Third Party Access



OAuth Workflow [Google]



- Anonymous user visits your website / app
- User wants to use Google Identity
- They click a “Log In” button on your site / app and are redirected to Google’s website, and are prompted to accept certain permissions
- If they accept these permissions, Google will redirect the user back to your website along with an *authorization code*.
- You can exchange this code for *access token* and *refresh token*
- You can then use this *access token* to actually retrieve the user’s information via API from Google. The *refresh* token is used when the access token expires.

<https://developers.google.com/identity/protocols/OAuth2>

OAuth Workflow [Google]

(1) User wants to login via Google

(2) Redirect to Google's Authorization Server:

```
https://accounts.google.com/o/oauth2/v2/auth?  
  scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive.metadata.readonly&  
  access_type=offline&  
  include_granted_scopes=true&  
  state=state_parameter_passthrough_value&  
  redirect_uri=http%3A%2F%2Foauth2.example.com%2Fcallback&  
  response_type=code&  
  client_id=client_id
```

redirect_uri is YOUR APP (where to go after authorization)

client_id is YOUR APP (need API key from Google)

(3) Google prompts user for consent

(4) Google redirects back to your app (via **redirect_uri**) and provides **authorization code**

```
https://oauth2.example.com/auth?code=4/P7q7W91a-oMsCeLvIaQm6bTrgtp7
```

OAuth Workflow [Google]

(5) Exchange **authorization code** for **refresh and access tokens** via HTTP/REST API

POST /oauth2/v4/token HTTP/1.1
Host: www.googleapis.com
Content-Type: application/x-www-form-urlencoded

```
code=4/P7q7W91a-oMsCeLvIaQm6bTrgtp7&  
client_id=your_client_id&  
client_secret=your_client_secret&  
redirect_uri=https://oauth2.example.com/code&  
grant_type=authorization_code
```

(6) Server returns JSON object with **access token** (short lived) and **refresh token**

```
{  
  "access_token": "1/fFAGRNJru1FTz70BzhT3Zg",  
  "expires_in": 3920,  
  "token_type": "Bearer",  
  "refresh_token": "1/xEoDL4iW3cxII7yDbSRFYNG01kVKM2C-259HOF2aQbl"  
}
```

(7) Use **Access token** to call Google API for specific data

OAuth

- OAuth is a *framework*, not a protocol
 - Implementations vary by enterprise
 - ~~Interoperability~~ 😞
 - You can't switch from Google sign-in to Facebook sign-in just by replacing `google.com` with `facebook.com`
- No signatures or cryptography, just plain tokens that are protected by TLS (web)

OAuth and OpenID Connect

- Two web standards but with different goals
- OAuth is a framework that provides *authorization*
 - Authorize other websites to access your Google Drive
 - Fine grain permission – OK to read/write Drive files, but not access your GMail
 - Does not handle how you authenticate with Google in the first place – that's Google's problem
- OpenID Connect is a layer built on OAuth that provides *authentication*