Nama : Ahmad Faiz Adnan Dhani
NIM : G211190125
Matkul : Rekayasa web
Kelas : TIB(Pagi)
Sumber : https://developers.google.com/identity/protocols/oauth2

Menggunakan OAuth 2.0 untuk Mengakses Google API

API Google menggunakan OAuth protokol 2.0 untuk otentikasi dan otorisasi. Google mendukung skenario OAuth 2.0 umum seperti untuk server web, sisi klien, terinstal, dan aplikasi perangkat dengan input terbatas.
Untuk memulai, mendapatkan OAuth kredensial 2.0 klien dari Google API Console . Kemudian aplikasi klien Anda meminta token akses dari Server Otorisasi Google, mengekstrak token dari respons, dan mengirimkan token ke Google API yang ingin Anda akses.

Langkah dasar

Semua aplikasi mengikuti pola dasar saat mengakses Google API menggunakan OAuth 2.0. Pada tingkat tinggi, Anda mengikuti lima langkah:

1. Mendapatkan OAuth 2.0 mandat dari Google API Console.

   Kunjungi Google API Console untuk mendapatkan OAuth 2.0 mandat seperti klien ID dan klien rahasia yang diketahui Google dan aplikasi Anda. Kumpulan nilai bervariasi berdasarkan jenis aplikasi yang Anda buat. Misalnya, aplikasi JavaScript tidak memerlukan rahasia, tetapi aplikasi server web melakukannya.
2. Dapatkan token akses dari Server Otorisasi Google.

   Sebelum aplikasi Anda dapat mengakses data pribadi menggunakan Google API, aplikasi harus mendapatkan token akses yang memberikan akses ke API tersebut. API. Parameter variabel yang disebut `scope` kontrol set sumber daya dan operasi yang akses izin tanda.
3. Periksa cakupan akses yang diberikan oleh pengguna.

   Bandingkan cakupan yang disertakan dalam respons token akses dengan cakupan yang diperlukan untuk mengakses fitur dan fungsionalitas aplikasi Anda yang

bergantung pada akses ke Google API terkait. Nonaktifkan semua fitur aplikasi Anda yang tidak dapat berfungsi tanpa akses ke API terkait.

4. Kirim token akses ke API.

   Setelah aplikasi memperoleh token akses, ia akan mengirimkan token ke API Google dalam header permintaan HTTP Otorisasi . Dimungkinkan untuk mengirim token sebagai parameter string kueri URI, tetapi kami tidak menyarankannya, karena parameter URI dapat berakhir di file log yang tidak sepenuhnya aman. Selain itu, praktik REST yang baik untuk menghindari pembuatan nama parameter URI yang tidak perlu.

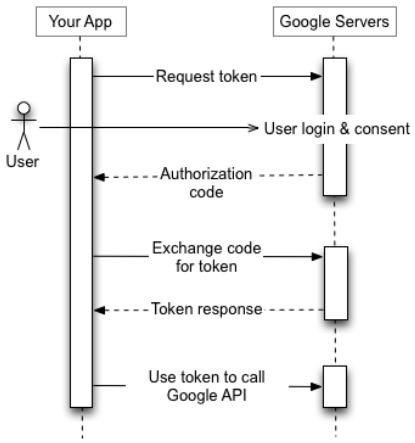
5. Refresh token akses, jika perlu.

   Token akses memiliki masa pakai yang terbatas. Jika aplikasi Anda membutuhkan akses ke Google API di luar masa pakai token akses tunggal, itu bisa mendapatkan token penyegaran. Token penyegaran memungkinkan aplikasi Anda mendapatkan token akses baru.

Skenario

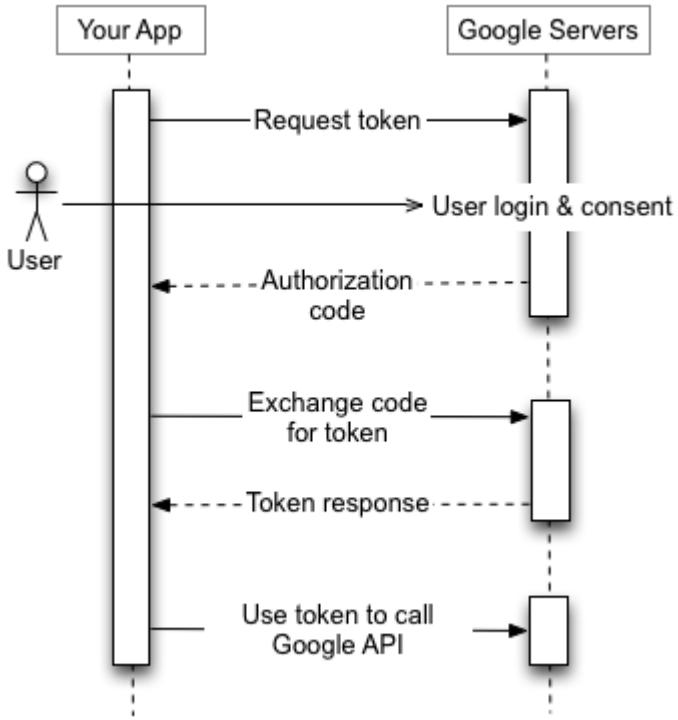
Aplikasi server web

Titik akhir Google OAuth 2.0 mendukung aplikasi server web yang menggunakan bahasa dan kerangka kerja seperti PHP, Java, Python, Ruby, dan ASP.NET. Aplikasi harus menyimpan token penyegaran untuk penggunaan di masa mendatang dan menggunakan token akses untuk mengakses Google API. Setelah token akses kedaluwarsa, aplikasi menggunakan token penyegaran untuk mendapatkan yang baru.



Aplikasi yang diinstal

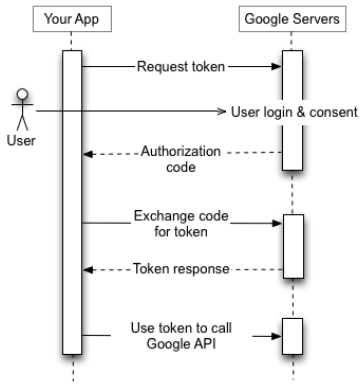
Aplikasi harus menyimpan token penyegaran untuk penggunaan di masa mendatang dan menggunakan token akses untuk mengakses Google API. Setelah token akses kedaluwarsa, aplikasi menggunakan token penyegaran untuk mendapatkan yang baru.



Aplikasi sisi klien (JavaScript)

Urutan otorisasi dimulai saat aplikasi Anda mengalihkan browser ke URL Google; URL menyertakan parameter kueri yang menunjukkan jenis akses yang diminta. Google menangani otentikasi pengguna, pemilihan sesi, dan persetujuan pengguna.

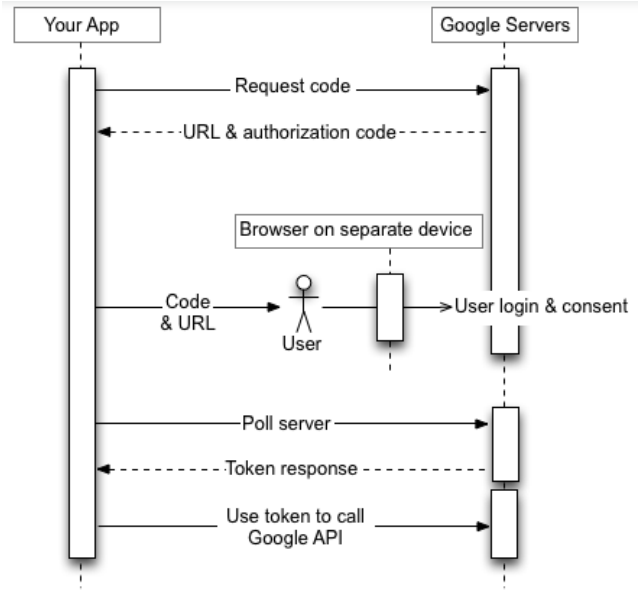
Hasilnya adalah token akses, yang harus divalidasi oleh klien sebelum memasukkannya ke dalam permintaan Google API. Ketika token kedaluwarsa, aplikasi mengulangi prosesnya.



Aplikasi pada perangkat dengan input terbatas

Titik akhir Google OAuth 2.0 mendukung aplikasi yang berjalan pada perangkat dengan input terbatas seperti konsol game, kamera video, dan printer. Pengguna memperoleh URL dan kode dari perangkat, lalu beralih ke perangkat atau komputer terpisah dengan kemampuan input yang lebih kaya. Pengguna meluncurkan browser, menavigasi ke URL yang ditentukan, masuk, dan memasukkan kode.

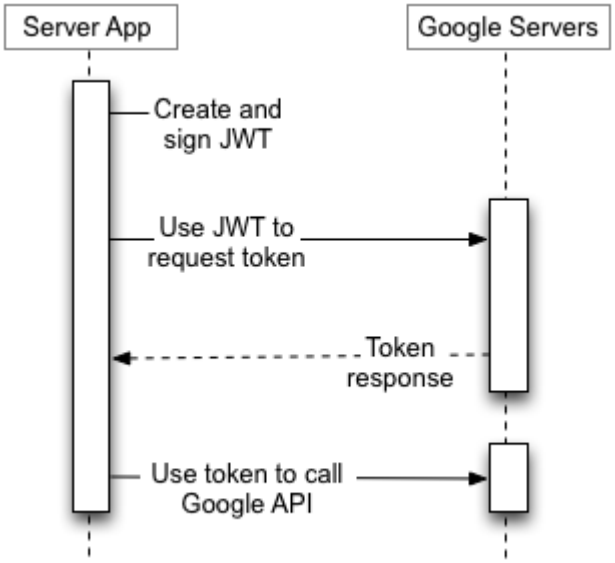
Sementara itu, aplikasi melakukan polling URL Google pada interval tertentu. Setelah pengguna menyetujui akses, respons dari server Google berisi token akses dan token penyegaran. Aplikasi harus menyimpan token penyegaran untuk penggunaan di masa mendatang dan menggunakan token akses untuk mengakses Google API. Setelah token akses kedaluwarsa, aplikasi menggunakan token penyegaran untuk mendapatkan yang baru.



Akun layanan

Google API seperti Prediction API dan Google Cloud Storage dapat bertindak atas nama aplikasi Anda tanpa mengakses informasi pengguna.

Untuk jenis server-ke-server interaksi Anda memerlukan **account layanan,** yang merupakan akun yang dimiliki aplikasi Anda, bukan untuk pengguna akhir individual. Aplikasi Anda memanggil Google API atas nama akun layanan, dan persetujuan pengguna tidak diperlukan. (Dalam skenario akun non-layanan, aplikasi Anda memanggil Google API atas nama pengguna akhir, dan persetujuan pengguna terkadang diperlukan). Aplikasi Anda kemudian mengirimkan permintaan token ke Server Otorisasi Google OAuth 2.0, yang mengembalikan token akses. Aplikasi menggunakan token untuk mengakses Google API. Ketika token kedaluwarsa, aplikasi mengulangi prosesnya.



Ukuran token

Token dapat bervariasi dalam ukuran, hingga batas berikut:

- Kode otorisasi: 256 byte
- Token akses: 2048 byte
- Segarkan token: 512 byte

Segarkan kedaluwarsa token

Anda harus menulis kode untuk mengantisipasi kemungkinan bahwa token penyegaran yang diberikan mungkin tidak lagi berfungsi. Token penyegaran mungkin berhenti berfungsi karena salah satu alasan berikut:

- Pengguna telah dicabut akses aplikasi Anda .
- Token penyegaran belum digunakan selama enam bulan.
- Pengguna mengubah sandi dan token penyegaran berisi cakupan Gmail.
- Akun pengguna telah melebihi jumlah maksimum token penyegaran (langsung) yang diberikan.
- Pengguna termasuk dalam organisasi Google Cloud Platform yang memiliki kebijakan kontrol sesi yang berlaku.