OAuth 2.0
Incremental Auth

IETF 102 Montreal, July 2018

William Denniss
Recap: Incremental Auth
Problem Statement

Asking for the kitchen sink of scopes up-front is a bad thing.

Users should have the context of the authorization request.

E.g. Granting a calendar scope only makes sense in the context of interacting with a calendar-related feature.
Hi Bill
billd1600@gmail.com

Google OAuth 2.0 Playground wants to

- View and manage the files in your Google Drive
- Manage your Blogger account
- Send email on your behalf
- Manage your contacts
- Manage your calendars
- Manage your YouTube account

Allow Google OAuth 2.0 Playground to do this?

By clicking Allow, you allow this app to use your information in accordance to their terms of service and privacy policies. You can remove this or any other app connected to your account in My Account.
Incremental Auth Definition

The ability to request additional scopes in subsequent requests adding to a single authorization grant representing all scopes granted so far.

Implies that the access and refresh token carry the union of all granted scopes.
Confidential Client Protocol

Auth 2.0 doesn’t stop you returning an authorization grant with *more* scope, so many people have implemented this already for confidential clients. This spec documents best practices, security considerations, and new authorization endpoint parameter \texttt{include\_granted\_scopes}.
Public Client Protocol

New token endpoint param: `existing_grant`.

When exchanging the authorization code from subsequent (i.e. incremental) requests, pass the previous refresh token in `existing_grant`.

Resulting access and refresh tokens will contain a union of the scope.
Incremental Auth for Native Apps

App -> Authorization Endpoint

Initial Authorization Request
Authorization Request (Scope A)
Response: authorization_code

Token Request: authorization_code
Response: access_token, refresh_token

Incremental Authorization Request
Authorization Request (Scope B)
Response: authorization_code (Scope B)

Token Request: authorization_code (Scope B), existing_grant (Scope A)
Response: access_token, refresh_token (Scope A, B)
Combined Grant $A \cup B$
Updates since IETF 101

Adoption as a working group document.

Pending work items:

Documented “scope” response param behavior for incremental auth.

Provide an RFC8414 metadata field & register with IANA registry.

Interop! Call to action: are you interested to implement incremental auth in your server or client? Get in touch!
New AppAuth Incremental Auth APIs

// builds authentication request
NSArray* scope = @[ @"https://www.googleapis.com/auth/calendar" ];
OIDAuthorizationRequest *incrementalAuthorizationRequest =
    [_authState incrementalAuthorizationRequestWithScopes:scope];

// performs authentication request
appDelegate.currentAuthorizationFlow =
    [_authState presentIncrementalAuthorizationRequest:incrementalAuthorizationRequest
        presentingViewController:self
        callback:^(
            BOOL success, NSError * _Nullable error) {
            // your code here
        }];
OAuth 2.0 Incremental Auth
Running Code

Google’s OAuth server already supports this spec!

Example application:
https://github.com/WilliamDenniss/AppAuth-iOS_IncrementalAuthDemo
OAuth 2.0 Incremental Auth

Discuss