

OAuth 2.0 YouTube API

Custom Brand Channels: Gadget Developer Guidelines

<https://support.google.com/youtube/bin/static.py?hl=en&page=guide.cs&guide=30071&topic=30085&answer=1727240>

(under Privacy & Security)

Authentication must be handled using OAuth, AuthSub, or other authentication method where credentials are not collected directly by the gadget. The gadget should not request more privileges than necessary for the user's intended action.

Designing and Running a Contest

<http://support.google.com/youtube/bin/static.py?hl=en&page=guide.cs&guide=30071&topic=30085&answer=1100988>

(under Permitted Contest Gadgets)

Must use either the AuthSub or OAuth APIs (and no others) for authentication. See the YouTube Developer's Guide for more information.

YouTube API 2.0

https://developers.google.com/youtube/2.0/developers_guide_protocol

We recommend that you use OAuth 2.0 authorization for your application, and the sample API requests throughout this documentation use the OAuth 2.0 syntax for their Authorization HTTP header values. The OAuth 2.0 protocol provides a standard way to access protected data on different websites and it is the recommended authorization mechanism for Google APIs. OAuth is an open protocol that may be implemented for many APIs, including Google APIs. All Google APIs, including the YouTube API, support the OAuth 2.0 authorization protocol. OAuth 2.0 relies on SSL for security instead of requiring your application to do cryptographic signing directly.

The OAuth 1.0, AuthSub, and ClientLogin authorization schemes also still work with the YouTube API. The APIs for those schemes have all been officially deprecated as of April 20, 2012. They will continue to work as per our deprecation policy, but we encourage you to migrate to OAuth 2.0 authorization as soon as possible. Similarly, if you are building a new application, you should use OAuth 2.0 authorization.

(under OAuth 2.0)

The YouTube Data API supports the OAuth 2.0 protocol for authorizing access to private user data. The list below explains some core OAuth 2.0 concepts:

- *When a user first attempts to use functionality in your application that requires the user to be logged in to a Google Account or YouTube account, your application initiates the OAuth2 authorization process.*
- *Your application directs the user to Google's authorization server. The link to that page specifies the scope of access that your application is requesting for the user's account. The scope specifies the resources that your application can retrieve, insert, update, or delete when acting as the authenticated user.*
- *If the user consents to authorize your application to access those resources, Google will return a token to your application. Depending on your application's type, it will either validate the token or exchange it for a different type of token.*