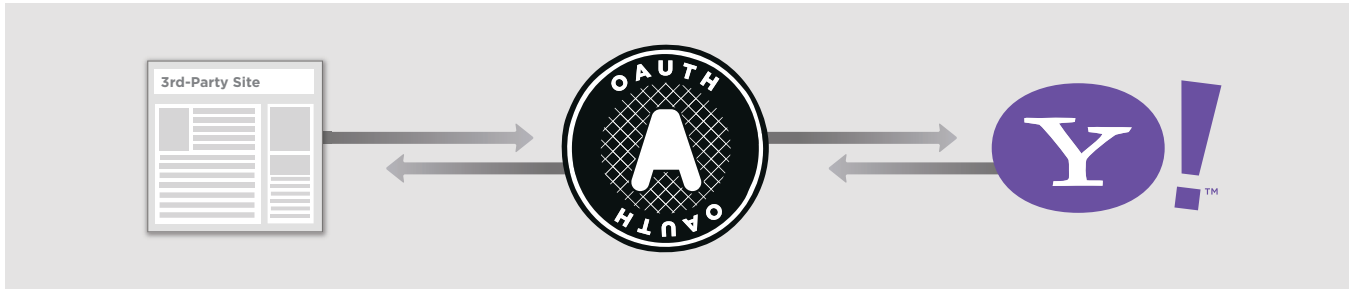




OAuth

An open protocol to secure API authorization

developer.yahoo.com/oauth



About OAuth

OAuth is a simple, secure, and quick way to publish and access protected data such as photos, videos, and contact lists. It's an open authorization model based primarily on existing standards that ensures secure credentials can be provisioned and verified by different software platforms.

User information is securely transferred. For example, if you want to run a photo-sharing app on your profile, you would have to allow your social networking site to share personal data with that app. OAuth allows you to anonymously grant the app access to your profile information.

Why OAuth

Many proprietary Web authorization protocols have emerged over the years: Google's AuthSub, AOL's OpenAuth, and BBAuth. OAuth integrates the commonalities and adopts the best practices of these protocols into a single open standard.

For developers, OAuth establishes a single, trustworthy, and open standard for authorizing access to user data.

For users, OAuth means never having to enter your Yahoo! ID and password into a third-party site. This increases security for users.

Some other reasons for using OAuth authorization:

- Compatible with existing authorization methods
- Flexibility to adjust to security needs of different sites
- Extensible through different signing algorithms
- Designed to work with mobile devices and desktop applications

Two Legged vs. Three Legged OAuth

OAuth can be used in two ways:

- To authorize third-party access to private Yahoo! user data.
- To secure communication between two services (Yahoo! and a third-party).

Apps that access private user data first require user authorization. The method is called "three-legged" in OAuth terminology because authorization occurs between three parties: the end user (User), the application (the Consumer), and the private data source (the Service Provider).

When creating apps that only use public data, you do not need authorization from the end User. These types of applications are called "two-legged" in OAuth terminology because the authorization occurs between two parties: an application and the public data source.

OAuth Security Updates

Yahoo! uses an updated version of the OAuth protocol, known as OAuth Core 1.0 Revision A, which fixes security issues discovered in a previous version of the OAuth protocol. You can read more about how Yahoo! has addressed this security issue here:

developer.yahoo.com/oauth/faq

OAuth Using Yahoo! Social SDKs

Yahoo! provides several SDKs that help ease the use of OAuth including PHP, Flash, and Objective-C for (iPhone and Macs). You can learn more about these SDKs here:

developer.yahoo.com/social/sdk

Getting Started

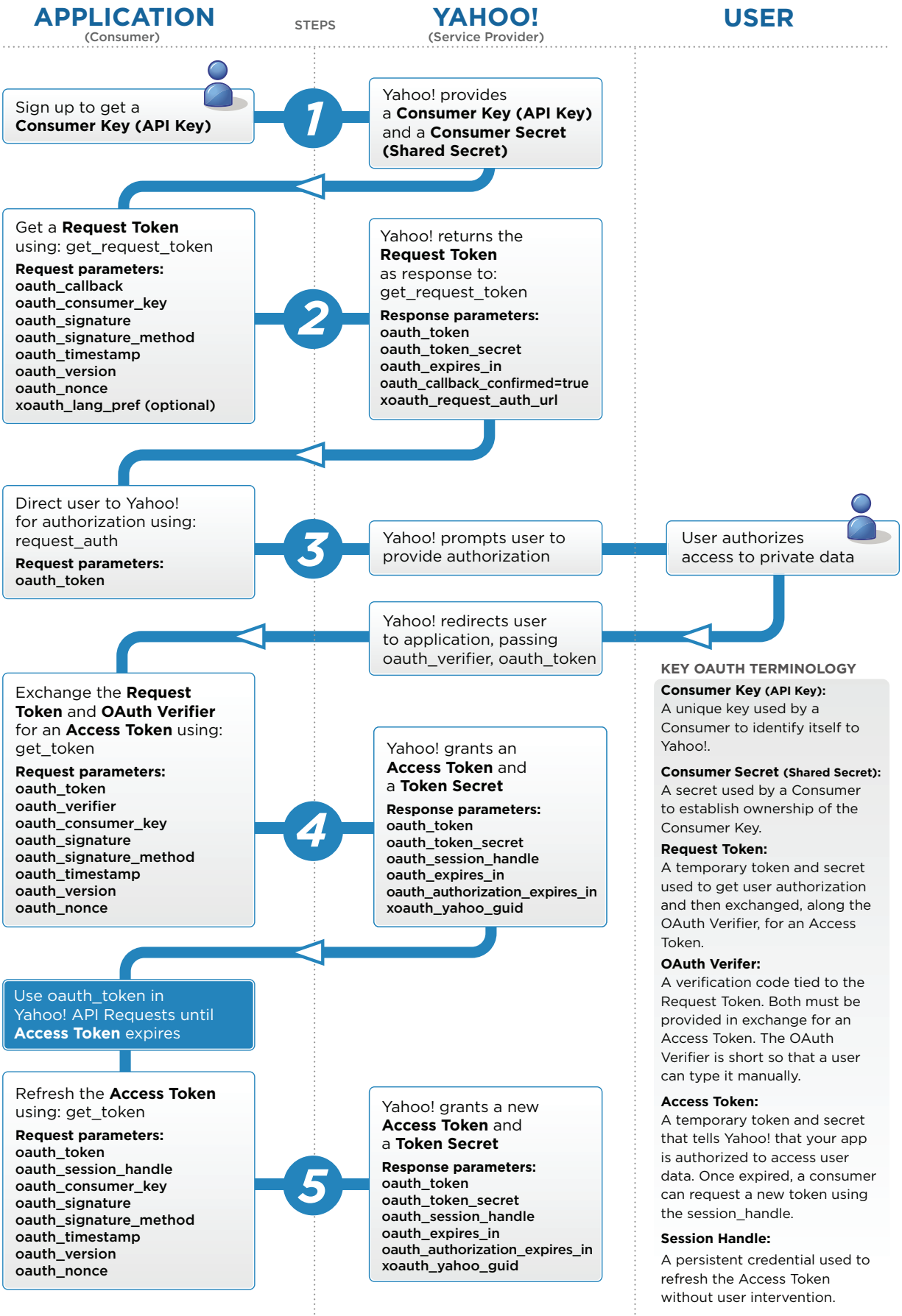
- 1 Get an API Key (Consumer Key) at developer.yahoo.com/dashboard/createKey.html
- 2 Read the OAuth Quick Start Guide at developer.yahoo.com/oauth/guide
- 3 Use the Yahoo! Social API SDKs (optional) developer.yahoo.com/social/sdk

Resources

YAHOO! OAUTH
developer.yahoo.com/oauth

TUTORIAL: CREATE AN OAUTH APP
developer.yahoo.com/oauth/guide/creating_oauth_app.html

OAUTH INFO & ADDITIONAL LIBRARIES
oauth.net



KEY OAUTH TERMINOLOGY

Consumer Key (API Key):

A unique key used by a Consumer to identify itself to Yahoo!.

Consumer Secret (Shared Secret):

A secret used by a Consumer to establish ownership of the Consumer Key.

Request Token:

A temporary token and secret used to get user authorization and then exchanged, along with the OAuth Verifier, for an Access Token.

OAuth Verifier:

A verification code tied to the Request Token. Both must be provided in exchange for an Access Token. The OAuth Verifier is short so that a user can type it manually.

Access Token:

A temporary token and secret that tells Yahoo! that your app is authorized to access user data. Once expired, a consumer can request a new token using the session_handle.

Session Handle:

A persistent credential used to refresh the Access Token without user intervention.



Step



Yahoo! API step



OAuth flow step



Step requiring human entry