HathiTrust Data API - Related Applications

Introduction

The HathiTrust Data API is referred to simply as API in this document.

This document describes the Key Generation Service (KGS) and the API web client (`htdc`).

It is a companion to the primary Data API documentation linked at http://www.hathitrust.org/data_api which details the available resources, their access restrictions and how to develop a program to access the API using keys obtained from KGS.

Key Generation Service (KGS)

The Key Generation Service is a web site where developers of API clients can request an OAuth key pair (`oauth_consumer_key` and `oauth_consumer_secret`) required to sign their API request URLs.   Following is a description of the secure mechanism used to communicate the key pair to the developer.

The KGS can be invoked at `http://babel.hathitrust.org/cgi/kgs/request`

To ensure that the recipient of the key pair is the same as the person who made the request, we require a valid email address. The email address serves several purposes:

- To email a signed, one-time URL link that presents a web page that displays the `oauth_consumer_key` and `oauth_consumer_secret`.
- To inform the developer that we may disable their `oauth_consumer_key` if abuse is detected.
    - [FUTURE] To optionally report daily usage to the developer by email, in order to:
    - alert the developer of illegitimate activity in the event their key pair is compromised
    - delete the key pair if the email address is no longer valid
- To update the developer of changes to the API.

KGS is accessed over https:// to secure the request's form data and the emailed KGS link that contains the one-time URL link that displays the key pair.

Registration

KGS provides the developer with a web form to register as follows.

1. Developer enters user name, institution name and email address and submits them over SSL to KGS.
2. KGS generates an `oauth_consumer_key` and a `oauth_consumer_secret` and stores them and the form data in the authentication database.
3. KGS generates a confirmation URL from the form data and `oauth_consumer_key` and emails it to the developer's email address. The URL consists of KGS host and path info and an `oauth_consumer_key` parameter in plain-text and a `oauth_signature` that is generated using the `oauth_consumer_secret` to encrypt the host, path info, `oauth_consumer_key` and form data:
    - **Example URL in email:** `https://babel.hathitrust.org/cgi/kgs/confirm?email=smith%40some.edu&name=Smith&oauth_consumer_key=fea256f552&oauth_nonce=adde9747a65ccaf073b0&oauth_signature=c%2F6KCYVM%2FysRy%2B5c2BR9QoF9syY%3D&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1331924673&oauth_version=1.0&org=Some%20University`
4. Developer receives email and clicks on URL
5. URL invokes the KGS over SSL. KGS retrieves `oauth_consumer_secret` and email address web form data by `oauth_consumer_key`. KGS performs identical encryption performed by client to sign the request and tests that signature matches and displays `oauth_consumer_key` and the shared `oauth_consumer_secret` on the KGS confirmation web page.

The key pair request is timestamped and deleted if the developer does not follow the email link within 24 hours to activate the key pair. The page displaying the keys cannot be reloaded or re-visited if the browser is closed.

Web Client

The web client provides an interface to the API that allows users to invoke the API directly from a web browser. This offers casual users a way to access the API without the hurdle of writing a program to generate OAuth signed API requests.

To use the web client, the user can log in to the portal at

`http://babel.hathitrust.org/cgi/kgs/portal`

or directly at

`http://babel.hathitrust.org/cgi/htdc`

"Friend" accounts are available to support login for users who are not affiliated with a HathiTrust institution. Instructions for setting up a friend account are available on the HathiTrust login page: http://babel.hathitrust.org/cgi/wayf.

Upon initial login a user is automatically added to the authorization database and may to use the web client to request unrestricted resources. See the main API documentation for information about the differences between unrestricted and restricted resources. The user's authentication credentials are a substitute for the identifying email address supplied by developers who request keys through the KGS.