tru.ly API

8/8/11 REV2.0

BACKGROUND

tru.ly maximizes personal privacy by providing users with a single, verified identity on the internet. tru.ly verifies users in conjunction with several verification partners by matching user submitted data against a database of both public and government data. Additionally, tru.ly allows verified users to link accounts from other websites to establish their ownership of those accounts. Currently, the linking of Facebook, Twitter, and LinkedIn are supported with plans to add additional platforms.

SAMPLE USE CASES

The following is a list of potential integrations that would leverage the tru.ly API to enhance the various partner sites. This list is simply a guide and not intended to be a definite list of possible API integrations. tru.ly is completely open to exploring all integration opportunities.

E-COMMERCE SITES AND CREDIT CARD COMPANIES

E-Commerce sites could use the tru.ly API to verify users before charging their credit cards. By verifying users, sites will be able to reduce credit card fraud, chargebacks, and un-authorized transactions.

DATING SITES AND SOCIAL NETWORKS

The tru.ly API could be leveraged to allow users to verify their profiles. Partner sites could surface this by using tru.ly as a profile enhancer, badge, or a level of segmentation. Partner sites could implement tru.ly in order to allow their members to remain anonymous while still displaying a tru.ly verified badge. Additionally, partner sites could use the experimental “request verification” endpoint to allow tru.ly verified users to request verifications from other members on-site.

COMMENT ENGINES

Comment engines could utilize the tru.ly API to cut down on spam, ham, and other inflammatory comments. tru.ly could replace captcha implementations and would naturally compliment services such as Akismet and Mollom.

AGE VERIFICATION

Sites that require age verification such as liquor, tobacco, or gambling websites could leverage the tru.ly API to securely validate the ages of their users. The tru.ly API would impose a stronger level of restriction on age restricted content versus simply asking for a birthday.

TECHNICAL OVERVIEW

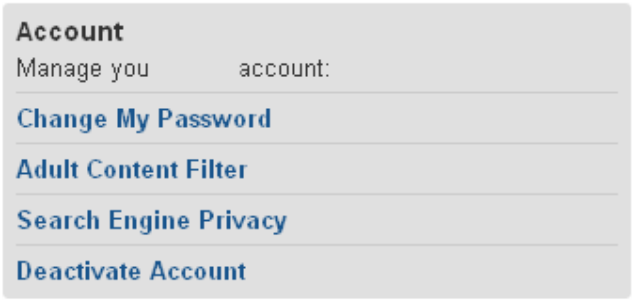
The tru.ly API is mainly focused around verifying users and allowing tru.ly verified users to share their verification with partner sites. With this goal in mind, the authentication and verification API is built on top of the OpenID 2.0 standard. OpenID is a widely deployed standard with dozens of client libraries available in various languages.

An extensive list of officially endorsed libraries can be found at http://openid.net/developers/libraries/ Additionally, the tru.ly API requires an OpenID library that supports the Attribute Exchange extension if the partner site intends to receive verified profile data from tru.ly

EXAMPLE IMPLEMENTATION

The following is a walkthrough of an example implementation of the tru.ly API. It should serve as a guide for how partner sites can integrate tru.ly into their platform.

Note: tru.ly currently only supports OpenID transactions using a popup window.



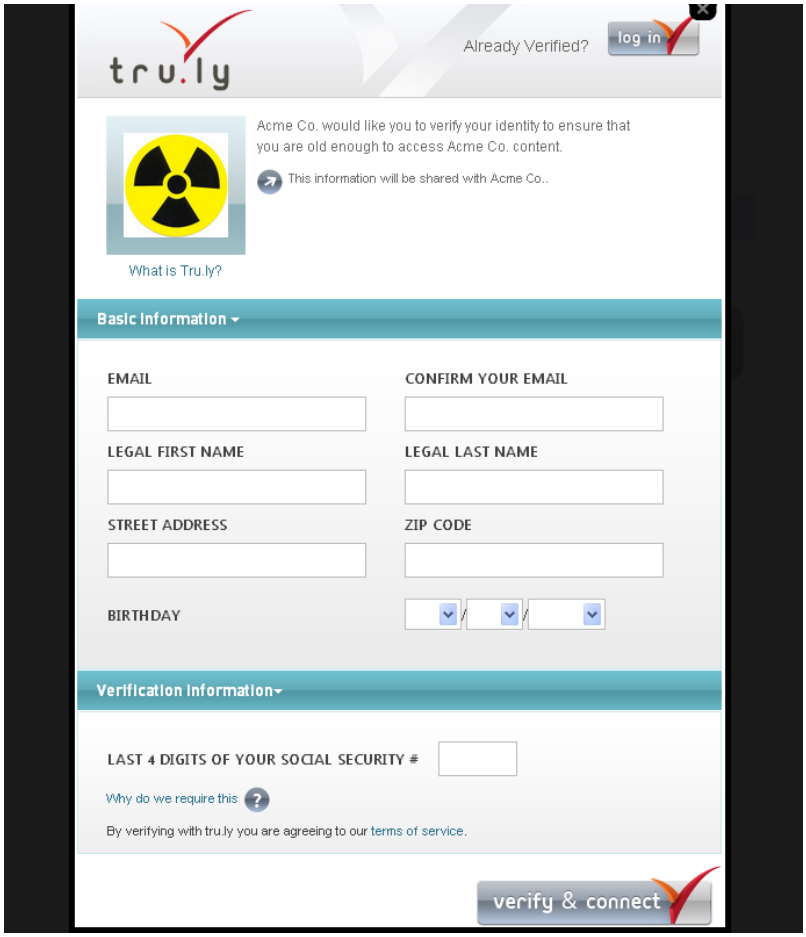
Step 1: Initiate tru.ly verification

- The initial step prompts the user to become verified via tru.ly
- The partner site should describe the benefits of becoming tru.ly verified. (profile badge, increased permissions, etc)

Step 1a: User initiates tru.ly verification

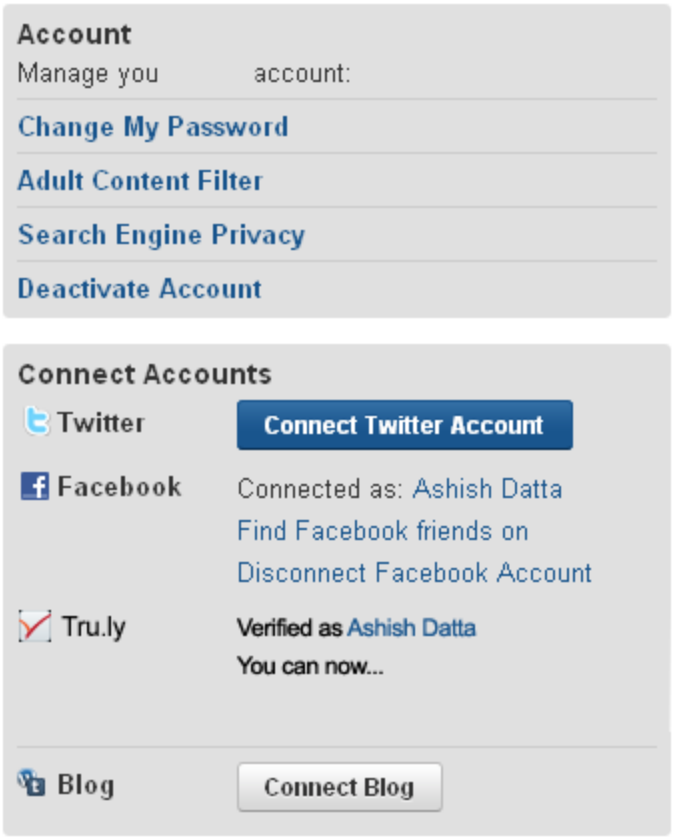
- When the user initiates a verification via tru.ly, the partner site should initiate an OpenID authentication request to the tru.ly OpenID endpoint.
- At this point, the partner site's OpenID client library should redirect the end user to tru.ly to begin the verification process.
- Additionally, at this point the partner site can utilize the OpenID Attribute Exchange extension to request that tru.ly pass return the user with additional information.

- **Note:** Currently, any attributes requested via Attribute Exchange will be required.



Step 2: User is prompted to verify

- Once the user arrives at the OpenID endpoint they will be prompted to become verified for whatever purpose the partner site specifies (profile badge, increased permissions, etc.)
- Additionally, users that all ready have tru.ly accounts will be able to login at this stage to avoid having to verify twice.
- **Note:** If the user cancels at this step they will be returned to the partner site with a canceled OpenID request.



Step 3: Complete OpenID Request

- Assuming the end user validates, they will be returned to the partner site with a completed OpenID request.
- At this point, the partner site must use an OpenID client library to retrieve the end user's URI
- Additionally, is the partner site requested profile fields with Attribute Exchange, they will be available inside the completed request.

.

GETTING STARTED

In order to jumpstart the integration process, tru.ly is asking prospective partners to provide answers to the following survey questions. tru.ly hopes to use these response to spark initial conversations and ultimately streamline integration.

- What language is your application built with (PHP, Ruby, Python, etc)? Any framework?
- Do you currently support OpenID or OAuth in any form?
- Roughly, how large is your user base (thousands of users, tens of thousands, etc)?
- How frequently does your team publish releases (continually, daily, weekly, etc)?
- Where is the development team that will be doing the integration located?
- When is the earliest you could begin integrating tru.ly?
- What kind of support does your organization provide (phone, email, forum, mixed)?
- Do you plan to use the Attribute Exchange extension to receive verified profile data from tru.ly?
- Please briefly describe how you envision tru.ly integrating into your platform.
- Do you have any immediate questions or concerns regarding the tru.ly integration?

AVAILABILITY

The tru.ly API is still currently under active development. tru.ly is actively seeking companies that would be interested in receiving exclusive early access to the beta version of the API.