API Integration Best Practices

Table of Contents

Workflow

The recommended user experience is as follows:

1. Promote exclusive benefits
2. Present the user with the ability to verify their eligibility.
   - If successful, move directly to the final step.
   - If unable to instantly verify, present the user with an opportunity to correct their information and resubmit.
3. If still unable to verify, present the user with the option to upload documentation proving their eligibility.
4. Upon successful receipt of documents, notify the user that they will receive an update when verification is complete.
5. If eligibility cannot be confirmed with document review, notify the user with a reason why and invite them to upload a valid document. This should be done with an email notifier.
6. Upon successful document review, notify the user and provide a link to proceed to the final step.
7. The Final Step: present the user with a message indicating they have been verified and proceed to the next steps which are exclusive to eligible users.

If the user fails document review or else continues to have difficulties verifying, consider having some plan for instructing the user what to do next. For example, you can direct them to a Contact Us or FAQ page. This step can be further advised during Project Planning Readiness Meeting with SheerID representatives.

Note that the above workflow may be asynchronous - a manual review of documents will result in a short delay (generally about 5 minutes, could take up to 20 minutes). It's important to consider this potential when deciding where in the process to place the eligibility verification. For best results, eligibility verification should occur at an early enough time in the conversion funnel to allow for

some time to elapse between the initial verification attempt and a final document review response if necessary. Our customers have found success performing this check on a targeted landing page, at the time of account signup or when adding an item or a promo code to a shopping cart. It should not be performed during order finalization (like a credit card transaction), since you won't want to hold up the processing of the order.

Access Token Security

The SheerID API access token(s) used to authenticate requests should be protected like a sensitive password. It should go without saying that this token should never be exposed on the client side or in any page's HTML source. Provisions should be made to store this value external to the source code so that it can be rotated periodically.

Unified Verification Response Handling

The VerificationResponse object (JSON) returned by the Verify method is identical to that returned by the Inquire method. It's suggested that you use the same subroutine to process the VerificationResponse (attributing a customer record, granting a discount) whether it is the direct, synchronous result of a call to Verify, or if it's the result of performing an Inquire after receiving a HTTP notification that the response is complete.

Design considerations

Persistence

It's preferable to maintain a relationship between the SheerID verification request performed and some object in your system, whether it be a registered user, a quote (shopping cart), or an order. This can be done in one of two ways:

1. Storing SheerID Attributes

Add additional fields in your database to store the following data points that are returned with a VerificationResponse:

- `requestId` - the unique identifier for the SheerID verification request
- `personId` - the unique identifier for the individual that was verified, can be used to track a individual's verification activity
- `timestamp` - when the verification occurred. For some affiliation types (students, employees) it is advantageous to periodically re-verify
- `affiliations` - if your integration is handling multiple types of affiliations, this is helpful in

presenting verified users with the appropriate benefits

2. Using Request Metadata

Additional attributes (such as your site's customer ID, order ID, etc.) can be included with the verification request that is submitted to SheerID. This information is stored by SheerID with the verification request and can be obtained at any time by using the Inquire API method to fetch the verification response. These request metadata can also be included in bulk reports of verification activity. Even when using this technique to link SheerID activity with your system, it's recommended that your system store the Verification Request ID so that the additional attributes can be retrieved from SheerID.

User Experience

When designing mockups of the user experience, be sure to visually display which information fields are required by using an *asterisk, or else mention that all fields are required. Consult the Required Verification Fields for a list of required and optional data for each type of verification you wish to perform.

If the type of verification you use requires an organizationId, we recommend you use an Organization Combobox JSAPI component or else obtain the list of SheerID organizations from the REST API so that you can help users select the ID that matches their organization. We recommend that you implement a closed organization list to ensure users can only submit an organization included within SheerID's organization lists. This guarantees the verification request will contain a valid organizationId and be less likely to fail instant verification and get passed to document review. If you wish to include organizations outside SheerID's organization lists, we can work with you to add custom organizations for your implementation.

If using the SheerID combobox for international student verification, including an extra verification form field to collect country is recommended. The country can be used with the combobox to change and limit the scope of organizations populated. If no country is provided, only US organizations will be populated.

Coding Considerations

Instant Verification

An instant verification request returns one of three result responses, `true`, `false`, or `null`. Your implementation's workflow should handle `false` and `null` responses the same way.

Handling Bad Inputs

As stated before, we recommend that your implementation handles malformed or incorrect inputs and displays the error message so that the user can correct their mistakes. See REST API Error Codes and REST HTTP Errors for more details on what types of errors might occur.

Some of these errors are easily avoidable, for example, an invalid date format. We recommend you

design date input forms to be calendar inputs or separate day-month-year fields, and then reformat it to SheerID's accepted model before issuing a verification request to avoid this issue.

Prevent Fraud and Abuse

Same Person Limit

The samePersonLimit request configuration can be supplied to restrict the number of successful verifications performed by a single end-user. For example, by specifying the following additional request parameters when invoking the Verify method, a user would only be able to verify successfully once per 90 days:

```
_samePersonLimit=1&_samePersonLimitExpiration=90
```

If the same person limit policy has been exceeded, the request will not be completed and the Verify REST API resource method will return a 409 Conflict HTTP status code rather than a VerificationResponse JSON object.

Technical Language Considerations

Military Documents

Directly asking for military documents via upload is asking end-users to violate a federal regulation that prohibits members of the military from uploading reproduced images of their ID, or CAC card. This article sums this up well. Military IDs are still an acceptable form of documentation, but can't legally be requested.

Here are some examples of acceptable language when asking for military documents:

- **General Example**: Upload any military documentation that shows first, last name, military status, and valid date (if active duty).
- **Active Duty:** Any document that proves you are currently serving under Title 10 Active Duty Orders for 30 days or more.
- **Reservist:** Any document that proves you are currently serving under Title 32 Active Duty Orders for 30 days or more.
- **Military Retiree:** Any document that proves you are a Retiree from the US Armed Forces, Disabled Veteran with a rating of 30% or higher, or a registered Military Dependent.
- **Military Family:** Any document that proves you are a registered Military Dependent.
- **Veteran:** Any document that proves you met the qualifications of military service and were honorably discharged.
- **Disabled Veteran:** Any document that proves you are a Retiree from the US Armed Forces, Disabled Veteran with a rating of 30% or higher, or a registered Military Dependent.

Control Center

You should issue your verification requests against a web template even though you are not using SheerID's hosted implementation. You can create multiple templates for different groups of people you wish to verify, i.e. college students, teacher, military, etc. This is particularly important if you plan to offer benefits to multiple groups of people, so that, for example, students are verified against student-type affiliations and military personnel are verified against various military affiliations. Use the web template's `templateId` as a parameter when issuing a verification request so as to verify an end-user against that template.

Even though there are API endpoints for doing so, you should create Web Templates, Reward Pools, and Notifiers in Control Center as it has greater control over configurations and can link the workflow together quite easily. Make sure you have a `Template Admin` role, which is disabled by default.

HTTP Notifier

An HTTP Notifier is an event handler which sends a HTTP request (a "webhook") to a specified URL on your site for each state change that occurs for Verification Requests in your account. Your callback URL should be provisioned to accept the requestId parameter from the request (either GET or POST method depending on your choice), and then use the Inquire REST API method to obtain the corresponding VerificationResponse.

If using a HTTP Notifier configured for POST notifications, it is recommended to validate that the notification comes from SheerID for an added level of security. For more information on validating SheerID HTTP Notifications, see here.

You should use Control Center to create HTTP notifiers, but use the REST API to update it, specifying what filters (situations) in which to fire the notifier. We recommend filtering the notifier to not take any action upon specific events. By default, a notification will be fired as the verification request moves through the status transitions from NEW (created) to OPEN (submitted but not verified) to PENDING (docs uploaded, awaiting decision) to COMPLETE (finished). In most cases, only some of these verification states are actionable (i.e. COMPLETE).

Email Notifiers

An email notifier is an event handler that sends the end-user an email (with content customized by you) as a result of the following actions:

- Failed document review - conveys the reason the document was not approved and optionally includes a link back to your site so that they may upload an alternate document.

- Successful document review - notifies the end-user of a completed document review, with a direct, personalized link back to the final ("success") step in your verification workflow.
- Successful instant verification - welcomes the user and provides instructions on how to enjoy their exclusive benefits.

Configure the body of an email notifier with your personal branding so as not to mislead the user into believing it is spam or phishing. We use Apache Velocity to configure email content, which can be used in addition to HTML and CSS. See here for more information on email styling.

We recommend providing users with the option to upload additional documents in the event their document is rejected by document review by including a try-again link in the Failure Email. This link should redirect users to the document upload page you have configured in your verification flow and include an uploadToken (part of the query string parameter) to be used with a new upload request.

Caution should be exercised when emailing your end-user a link back to the success step. For the best user experience, it's important that the link takes the user back to a personalized destination which continues where they left off; however, it's important to take steps to prevent this user from being able to share that link and its associated restricted benefits (for example, a 25% discount) with other users.

The best way to handle this for applications that have user accounts is to store the in-progress verification request ID in the session or with the user account so that they can return to the verification workflow on a subsequent visit or session. If this strategy is used, the URL does not need to contain any state. If user accounts are not supported or desired in this workflow, a single-use token or key should be generated and stored using one of the techniques described in the "Persistence" section above. URLs may then use the single-use token to allow the user to recover the state of their workflow in progress. Make sure to invalidate or destroy this token once the verification workflow has been completed.

Document Upload

Document upload is a necessity in order to ensure a high-quality user experience for your integration. Submitting a document for review has three steps:

1. Perform a verification request to attempt instant verification.
2. We recommend displaying to the customer the information submitted with the verification request (name, email, birth date, organization, and other required fields) as an optional "Correct My Information" step before proceeding to document review.
3. Issue an Asset Upload Token associated with the unsatisfied request.
4. Upload one or more documents for review.

We recommend using our JSAPI Asset Upload component to allow your user to submit their documents directly to SheerID. Once you have obtained an asset token, this module can be used to

easily place and customize a SheerID document upload form on any HTML page. This simplifies the coding for you and improves the performance and security of the document transfer for your end-user.

We also recommend displaying the user's first name, last name, and organization they provided in the first step and remind them that the document must match that information. Potentially, users will need to upload multiple files to prove their eligibility so it is recommended to accept multiple files. For example, the back and front scans of a driver's license in combination will satisfy the information field requirements. The behavior to accept multiple documents is built into the JSAPI Asset Upload component.