AUDIT LOG - API

Oct 2020

Purpose of the API

The Audit Service API is designed to provide a method of programmatically fetching the audit logs that are visible within the Admin web pages.

Getting Started

In order to start using the Dynamic Forms API, there are several key requirements:

1. A greenID account must be established. This includes specifying the rules to be used, the data sources that will be available, and a range of other configuration options.
2. Get the Web Service credentials. These consist of an "account ID" and a Web Service password. These must be obtained from greenID and used in every Web Service call.
3. Consume the WSDL.

GreenID offers two separate environments: test and production. Customers start out in test, where they may carry out their development activities, performing as many test verifications as they like. Once the customer is satisfied with their integration, their account can be activated in the production environment.

How to Use the API

The API has only one method:

- `retrieveAuditLogs` - this method returns the audit logs for a particular verification attempt. All logs related to the specified validation attempt, including calls to external data sources, and any admin activity is returned.

See Web Service Endpoints for WSDL details.

API Parameters

A successful audit log request call looks like the following:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://services.registrations.edentiti.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:retrieveAuditLogs>
```

```
            <accountId>account_id</accountId>
            <password>password</password>
            <verificationId>pUz9rXAc</verificationId>
         </ser:retrieveAuditLogs>
      </soapenv:Body>
   </soapenv:Envelope>
```

The accountId and password are the same credentials that Edentiti has given for your account. The verification id is the same code listed under "REF NO" in the admin panel and is related to a specific customer.

Additionally, filters can be added to the audit log request, for example:

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://services.registrations.edentiti.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:retrieveAuditLogs>
         <accountId>account_id</accountId>
         <password>password</password>
         <verificationId>pUz9rXAc</verificationId>
      </ser:retrieveAuditLogs>
   </soapenv:Body>
</soapenv:Envelope>
```

This will limit the results returned to only verification attempts of a person. The list of filters includes:

- `datasourceattempt` - verification attempts by a person.
- `webservice` - webservice calls. This includes things such as a user being successfully registered or requesting session tokens.
- `statechange` - overall state changes of the user's verification attempt.
- `admin` - administration logs related to a single user.

- `thirdpartycheck` - logging of any third-party checks.

More than one filter may be applied. In the event of more than one filter, logs are returned if they match on any of the filters.

Interpreting Results

A single audit log from the Audit Service look like this:

```
<auditlog>

    <dateAudited>2014-07-11T13:40:58.335+10:00</dateAudited>

    <errorEvent>false</errorEvent>

    <eventDescription>A watchlist check against the U.S. Treasury - Office of Foreign Assets Control Specially Designated Nationals List
source resulted in no match</eventDescription>

    <eventType>system</eventType>

    <eventCode>datasourceattempt</eventCode>

    <eventStatus>PASSED</eventStatus>

    <eventSubCode>OFAC Watchlist</eventSubCode>

    <guid>2863bf62-5faf-4200-b214-a75810a71750</guid>

</auditlog>
```

Interpreting audit logs

All audit logs have a dateAudited, whether or not it is an error event, an eventDescription, eventType, and guid. The event type is either system, admin, or customer. System events are background events, admin events are those created by an administration task, and customer events are generated from customer actions.

Most audit logs also have an eventCode, which contain the same values that can be placed in the filter when requesting the logs.

- `datasourceattempt`
- `webservice`
- `statechange`
- `admin`
- `thirdpartycheck`

There is also a sub-code, which appears for datasourceattempt logs, as well as for some webservice and admin logs. This sub code specifies what data source was being queried, or more specifically what occurred.

Possible 'datasourceattempt' Sub-Codes

The possible datasourceattempt sub-codes describe the data source that was being queried. The sub-code corresponds to the name of a data source as per the names returned via other web services (e.g. NSWRegoDVS, NSWRego, Wp, etc). When the eventType is customer then the attempt was done by the customer. If the eventType is system then the source or watchlist attempt was done in the background without the customer causing the attempt.

Possible "webservice" Sub Codes

The sub-code in the logs contains the method name called in the web service. Examples are registerVerification and getVerificationResult.

Event Statuses

This field will tell you whether the logged event was successful or not. There are a number of different possible results.