ICUC Tool of NA API Documentation

Project Overview

ICUC will provide moderation services for uploaded video.

Processing Details

Video Upload

A Video is uploaded to the SocialPatrol moderation and curation system by means of an HTTPS interface. The calling system submits identifying information for the Video to be moderated, including a URL of where the Video is located.

The calling application or system will submit the data to the *delivery* edge of the moderation stream interface. The format of this message is defined in the ***Entry Submission*** section of the API specification, below.

Moderator Review

Once the Video link has been retrieved, they are presented to ICUC's content specialists for moderation.

A set of custom *Approval* and/or *Rejection* reasons can be configured and used for reporting. The actual reason for approval and/or rejection does not have an affect on the processing workflow.

Notification

Once the message has been rejected or approved, the SocialPatrol application can immediately notify the submitting system through the use of a HTTPS request message ***sent to a specified callback URL***. The format of this submission is outlined in the API specification.

Submission Status

The submitting system can optionally also retrieve the status of any individual entry, or for all entries created within a defined time range.

Reporting

The SocialPatrol system creates Daily and Weekly summary reports detailing the overall moderation process, the number of entries approved and rejected, and the reasons for such. The exact format of the reports can be customized. Reports are produced in PDF format and are stored for later download, and may optionally also be emailed on completion.

API Details

Authorization

Authorization to the SocialPatrol API calls is performed using standard OAuth2 protocols. For external systems that need to connect as a service account, they can request access using the OAuth2 Token provider endpoint.

```
POST /service/oauth2/token
Host: www.socialpatrol.net
Content-Type: application/x-www-form-urlencoded

grant_type=urn%3aietf%3aparams%3aoauth%3agrant-type%3ajwt-bearer&assertion={JWT Header}.{JWT Claim Set}.{JWT Signature}
```

The JWT items in the request assertion data are Base64 encoded JSON arrays with the following sub-fields.

Header Field Name	Description	Required
alg	Signing algorithm. Currently the only supported value is ***RS256***, which indicates an RSA SHA=256 algorithm.	Mandatory
typ	The format of the signed claim request. Currently only ***JWT*** is supported	Mandatory
Claim Set Field Name	**Description**	**Required**
iss	The email address of the services account	Mandatory
scope	A space delimited set of application permissions being requested. For the purposes of this specific project, the permission required is: socialpatrol.net:stream-entries.manage	Mandatory
exp	The requested token expiration time, which has a maximum value of 24-hours after the issued time. If omitted this will default to one hour after the issued time.	Optional
iat	The time the assertion was issued.	Mandatory

If the requested access is granted, then the response will have an HTTP Status code of **200 - Success**, and the response body will contain a JSON array with the following fields.

Field Name	Description	Required
access_token	The access token to be included in subsequent requests.	Mandatory
token_type	The type of access token being provided. Currently the only supported value is ***Bearer***.	Mandatory
expires_in	The expiry time, in seconds since the IAT from the original assertion	Mandatory

If access is refused, then the response will have an HTTP status code of **400 - Bad Request**, and the response body will contain a JSON array with the following two fields:

Field Name	Description	Required
message_type	The type of message, in this case the value will be “Error”.	Mandatory
message	The textual error message describing the cause of the request being rejected.	Mandatory

Possible reasons for rejection include:

- The specified service account is not configured for external system access.
- The requested scope is not valid or not approved.
- The signature could not be verified, or the requested algorithm is not supported.
- The IAT time is more than 10 minutes different from the current time at the SocialPatrol server.

Entry Submission

The SocialPatrol application has a standard REST based interfacesub which external systems applications can use to deliver items into the moderation streams.

```
POST /service/delivery/{StreamId}/entries
Host: www.socialpatrol.net
Authorization: Bearer {AUTH_TOKEN}
Content-Type: application/json

{ “entries” : [] }
```

The ***entries*** is a JSON object array of one or more ***entry*** objects, each of which have the following field structure:

Field Name	Description	Required
entryId	Character string used as a unique identifier for this entry. Must be unique within the stream.	Mandatory
parentId	Optional id of parent or other related entry	Optional

author	The author information is optional, although if it is present then additional moderation and/or analysis can be performed.	Optional
authorName	The textual name of the author	Optional
authorImage	An image object used as an avatar for the author	Optional
authorId	The unique ID for the author within the submitting system	Optional
video	The object attributes of the image to be moderated	Mandatory
videoUrl	The URL where the image is to be downloaded from	Mandatory
videoType	The MIME type for the referenced media (video/mp4)	Mandatory
entryText	Text associated with the image.	Optional
entryTime	The timestamp for the image was created/uploaded in the source system.	Mandatory
entryTypes	A list of entry types.	Optional
callBackUrl	SocialPatrol will send notification callback to the given url	Optional

On success, the application will return an HTTP status code of **203 - No Data**. This response will have an empty message body. If there is a problem with the upload, the application will return a status code of **400 - Bad Request** with and the response body will contain a JSON array with the following two fields:

Field Name	Description	Required
message_type	The type of message, in this case the value will be “Error”.	Mandatory
message	The textual error message describing the cause of the request being rejected.	Mandatory

Possible reasons for rejection include:

- The access token is invalid or expired.
- The *streamId* in the request is not associated with the current service account.
- The image was not found at the specified *imageUrl* location.
- The image format specified in the *imageType* parameter is not supported.

Moderation Notification

Once an individual entry has been approved or rejected, SocialPatrol can notify the external system in real time. This notification is performed by mean of a PUT request to the desired callback url.

```
PUT /some/callback/{StreamId}/{entryId}
Host: www.some-system.com
```

```
Authorization: Bearer {AUTH_TOKEN}
Content-Type: application/json

{“status” : “APPROVED”}

Or

{"status" : "REJECTED", "category":[{"class":"REJECTED","code":"COPY"}]}

Class name and code can be different, it is just an example
```

The external system is expected to return a **200 - Success** or a **203 - No Data** in response. Note that it is possible to send multiple status updates, for example when a previously approved entry is subsequently rejected.

Item Status

The status of an individual entry can be retrieved from SocialPatrol by calling the external API.

```
GET /api/external/{StreamId}/entries/{entryId}Host: www.socialpatrol.net
Authorization: Bearer {AUTH_TOKEN}
Content-Type: application/json
```

You can also request the status of multiple entries whose creation time falls within a specified time period.

```
GET /api/external/{StreamId}/entries?from={datetime}&to={datetime}
Host: www.socialpatrol.net
Authorization: Bearer {AUTH_TOKEN}
Content-Type: application/json
```

If no time period is specified, then the application will return the most recent 200 entries.

The returned response will have an HTTP status code of **200 - Success**. The response body will contain an array of ***entries*** data, which is a JSON object array of zero or more ***entry*** objects, each of which have the following field structure:

Field Name	Description	Required
entryId	Character string used as a unique identifier for this entry. Must be unique within the stream.	Mandatory
parentId	Optional id of parent or other related entry	Optional
author	The author information is optional, although if it is present then additional moderation and/or analysis can be performed.	Recommended

authorName	The textual name of the author	Optional
authorImage	An image object used as an avatar for the author	Optional
authorId	The unique ID for the author within the submitting system	Optional
video	The object attributes of the image to be moderated	Mandatory
videoUrl	The URL where the image is to be downloaded from	Mandatory
videoType	The MIME type for the referenced media (video/mp4)	Mandatory
category	An array of zero or more category objects that indicate how the entry was moderated. Every category object must contain both a class and a code.	Optional
class	The category classification. Possible values include ***APPROVED***, ***REJECTED***, and ***EXTERNAL***.	Mandatory
code	Codes indicating the approval and or rejection reason.	Mandatory
entryText	Text associated with the image.	Optional
entryTime	The time the image was created/uploaded in the source system.	Mandatory
entryTypes	A list of entry types.	Optional
status	The status of the entry in processing. The following values are expected: **PREMOD** - The entry has been created but the associated image has not been retrieved yet. **NEW** - The entry has not been moderated. **PENDING** - The entry has been moderated and there is an outstanding notification to the source system. **APPROVED** - The entry has been moderated and is approved. **DELETED** - The entry has been moderated and is rejected.	Mandatory

If access is refused, then the response will have an HTTP status code of **400 - Bad Request**, and the response body will contain a JSON array with the following two fields:

Field Name	Description	Required
message_type	The types of message, in this case the value will be "Error".	Mandatory
message	The textual error message describing the cause of the request being rejected.	Mandatory

Possible reasons for rejection include:

- The access token is invalid or expired.
- The *streamId* in the request is not associated with the current service account.

- The specified *entryId* was not found in the existing dataset, in the case where a single status is being requested.

Item Update

The external system can update the status of an individual entry in the Social Patrol system by calling the external item specific API endpoint. Classes and categories may change depending on the client needs.

```
PUT /api/streams/{StreamId}/{entryId}
Host: www.socialpatrol.net
Content-Type: application/json

{ “action”: “ADD | REMOVE”,
  “class” : “APPROVED”,
  “category” : “KEPP” }
Or
{ “action”: “ADD | REMOVE”,
  “class” : “REJECTED”,
  “category” : “SPAM | PROF | INAP | PI” }
```

The fields in the JSON data are as follows:

Field Name	Description	Required
action	Indicates whether the supplied category tag should be added or removed	Mandatory
class	The category class	Mandatory
category	The category tag to add or remove from the item.	Mandatory

On success, the application will respond with an HTTP status code of **200 - Success**.
In the case of an error, then the response will have an HTTP status code of **400 - Bad Request**, and the response body will contain a JSON array with the following two fields:

Field Name	Description	Required
message_type	The type of message, in this case the value will be “Error”.	Mandatory
message	The textual error message describing the cause of the request being rejected.	Mandatory

Possible reasons for rejection include:

- The access token is invalid or expired.
- The *streamId* in the request is not associated with the current service account.
- The specified *entryId* was not found in the existing dataset, in the case where a single status is being requested.