AIQ Platform Application Rest API Specification

Document Version	Date	Updated by	Comments
1.0.0	1/12/2018	Khang Toh	Initial revision

Table of Contents

AIQ Platform Application API constitute resources available to an AIQ application. To utilise the Application API, you will need the App ID and App Secret for an AIQ application. You can create an AIQ application through the AIQ dashboard at https://dashboard.aiq.tech

Like most Web-based API, AIQ Application API uses Request Header for authentication and also to allow passing in additional request data. There

Required Request Headers

Application API requests must contain the following AIQ-specific headers:

AIQ-User-ID	a value that can be used to uniquely identify an application user. Maximum length 100 characters; Excess will be truncated automatically.

This unique identifier is used to differentiate individual app users for the purpose of statistical analytics and for app to track its users.

Optional Request Headers

Additionally Application API also accepts the following optional AIQ-specific headers. Value of the headers determine the verbosity or analytical data that customer can obtain via Management API.

AIQ-User-Gender	gender of application user [male|female|other]
AIQ-User-Age	user age
AIQ-User-Lat	latitude of application user
AIQ-User-Lng	longitude of application user
AIQ-Device-Type	device type running application [android|ios|web|other]
AIQ-Device-Model	device model running application
AIQ-Device-Version	device/os version running application
AIQ-App-Version	application version

Rate Limit

Rate limit is applied to all Application API request with a rate of 1 request per second. Rate violation will produce HTTP 429 with Retry-After header set to 1 minute.

Authentication

Prior to consuming Application API, all applications must authenticate themselves and present the authorization token in all requests. Authorization token must be Base64 encoded and submitted as part of the HTTP Request Header. For example:
Authorization: Bearer QWxhZGRpbjpPcGVuU2VzYW1l

Request authorization token
POST /app/v1/auth
Request Body Parameters

key *	string	application key
secret *	string	application secret

Response Body Parameters

token	string	authorization token for use in subsequent api calls
expiresIn	int	number of seconds until expiry
expiry	int	unix timestamp of expiry time

1. Collection

Get collection list
GET /app/v1/collection
Response Body Parameters

result	array	array of collections
id	string	collection unique identifier
name	string	collection name
imageUrl	string	collection image url
total	int	total number of records matching the query

2. Scan

Scan an image from a specific collection
POST /app/v1/scan/{collection_id}
This is multipart/form-data API; please set the Content-Type header accordingly.
Request Body Parameters

file *	file	image file to be matched

Response Body Parameters

id	string	unique identifier for this scan
triggerId	string	matched trigger identifier
triggerName	string	matched trigger name
triggerType	int	matched trigger type.Please refer to Appendix B for detail description for each value
triggerFps	int	matched video trigger frame rate
frame	int	matched frame number for video
thumbnailUrl	string	matched trigger thumbnail url
payloadUrl	string	matched trigger payload url
payloadData	string	matched trigger custom payload data
tags	string	matched trigger tags
score	float	matching score
ocr	array	array of string; each element is a line of detected text
timestamp	int	timestamp of the scan request

Scan an image from all collection
POST /app/v1/scan
This is multipart/form-data API; please set the Content-Type header accordingly. For performance reason it is recommended to use collection specific scan API endpoint, instead
Request Body Parameters

file *	file	image file to be matched

Response Body Parameters

id	string	unique identifier for this scan
triggerId	string	matched trigger identifier
triggerName	string	matched trigger name
triggerType	int	matched trigger type.Please refer to Appendix B for detail description for each value

triggerFps	int	matched video trigger frame rate
frame	int	matched frame number for video
thumbnailUrl	string	matched trigger thumbnail url
payloadUrl	string	matched trigger payload url
payloadData	string	matched trigger custom payload data
tags	string	matched trigger tags
score	float	matching score
ocr	array	array of string; each element is a line of detected text
timestamp	int	timestamp of the scan request

Scan a specific collection from url
POST /app/v1/scan/{collection_id}
Request Body Parameters

fileUrl *	string	url to image file to be matched

Response Body Parameters

id	string	unique identifier for this scan
triggerId	string	matched trigger identifier
triggerName	string	matched trigger name
triggerType	int	matched trigger type.Please refer to Appendix B for detail description for each value
triggerFps	int	matched video trigger frame rate
frame	int	matched frame number for video
thumbnailUrl	string	matched trigger thumbnail url
payloadUrl	string	matched trigger payload url
payloadData	string	matched trigger custom payload data
tags	string	matched trigger tags
score	float	matching score
ocr	array	array of string; each element is a line of detected text
timestamp	int	timestamp of the scan request

Scan all collection from url
POST /app/v1/scan
For performance reason it is recommended to use collection specific scan API endpoint, instead
Request Body Parameters

fileUrl *	string	url to image file to be matched

Response Body Parameters

id	string	unique identifier for this scan
triggerId	string	matched trigger identifier
triggerName	string	matched trigger name
triggerType	int	matched trigger type.Please refer to Appendix B for detail description for each value
triggerFps	int	matched video trigger frame rate
frame	int	matched frame number for video
thumbnailUrl	string	matched trigger thumbnail url
payloadUrl	string	matched trigger payload url
payloadData	string	matched trigger custom payload data
tags	string	matched trigger tags
score	float	matching score
ocr	array	array of string; each element is a line of detected text
timestamp	int	timestamp of the scan request

3. History

A mean for app users to retrieve their scan history in the past 3 months.
Get scan history
GET /app/v1/history
Request Body Parameters

count	int	number of records to return; default 20
offset	int	number of records to skip; default 0

Response Body Parameters

result	array	array of scan logs
scanId	string	scan unique identifier
timestamp	int	unix timestamp of scan
matched	boolean	scan matches a trigger
triggerId	string	matched trigger unique identifier
campaignId	string	matched campaign unique identifier
thumbnailUrl	string	matched trigger thumbnail url
imageUrl	string	scanned image url
payloadUrl	string	matched trigger payload url
lat	double	user latitude
lng	double	user longitude
total	int	total number of records matching the query

4. Reporting

Report a false positive
POST /app/v1/report
Request Body Parameters

scanId *	string	scan identifier
message *	string	feedback message

Response Body Parameters

id	string	report unique identifier
scanId	string	scan identifier
message	string	feedback message
status	int [0|1|2]	report status
reply	string	feedback reply message
timestamp	int	report timestamp

Get report status
GET /app/v1/report/{report_id}
Response Body Parameters

id	string	report unique identifier
scanId	string	scan unique identifier
message	string	feedback message
status	int [0|1|2]	report status
reply	string	feedback reply message
timestamp	int	report timestamp

5. Application User

Application Users are using Application as namespace. Application User resource applies to the currently active user.

Register app user

POST /app/v1/user

Request Body Parameters

email *	string	user email address
password *	string	password
name *	string	user full name
gender	string [male|female|other]	user gender
birthdate	string	user birthdate in iso 8601 date format

Response Body Parameters

email	string	user email address
name	string	user full name
gender	string [male|female|other]	user gender
birthdate	string	user birthdate in iso 8601 date format
avatar	string	avatar url

Get app user profile

Requires request header AIQ-User-Token to be set with token obtained from user login API.

GET /app/v1/user

Response Body Parameters

email	string	user email address
name	string	user full name
gender	string [male|female|other]	user gender
birthdate	string	user birthdate in iso 8601 date format
avatar	string	avatar url

Update app user profile

Requires request header AIQ-User-Token to be set with token obtained from user login API.

PATCH /app/v1/user

Request Body Parameters

At least one of the followings must be provided

email	string	user email address

password	string	password
name	string	user full name
gender	string [male|female|other]	user gender
birthdate	string	user birthdate in iso 8601 date format

Response Body Parameters

email	string	user email address
name	string	user full name
gender	string [male|female|other]	user gender
birthdate	string	user birthdate in iso 8601 date format
avatar	string	avatar url

Update app user avatar
Requires request header AIQ-User-Token to be set with token obtained from user login API.
POST /app/v1/user/avatar
Request Body Parameters
This is multipart/form-data API; please set the Content-Type header accordingly.

file *	file	avatar image file

User login
POST /app/v1/user/auth
Request Body Parameters

email *	string	user email address
password *	string	user password

Response Body Parameters

token	string	user session token to be used in AIQ-User-Token request header
email	string	user email address
name	string	user full name
gender	string [male|female|other]	user gender
birthdate	string	user birthdate in iso 8601 date format
avatar	string	avatar url

User logout
Requires request header AIQ-User-Token to be set with token obtained from user login API.

DELETE /app/v1/user/auth

Appendix A - Campaign Status

This list all possible values of campaign status.

Value	Description
0	Campaign is active
1	Campaign is paused/inactive. Triggers in this campaign will not return until campaign is back in active status.
2	Campaign is archived. Triggers in this campaign are archived.

Appendix B - Trigger Status and Types

This list all possible values of trigger status. All other value should be treated as rejection by vision system.

Value	Description
0	Trigger without media file; Pending training. Training will not start until trigger has media file
1	Trigger with media file; Pending training
2	Trigger training in progress
3	Trigger is trained and active. Trigger is now searchable
4	Trigger is archived. Trigger is no longer searchable
5	Trigger is rejected by vision system. Trigger cannot be searched
6	Trigger is inactive/paused. Trigger will not be returned

This list all possible trigger type

Value	Description
0	Image trigger
1	Video trigger

Appendix C - HTTP Response Codes

This list all possible HTTP status code that can be returned.

HTTP Response Code	Description
200	Successful request.
201	Resource created successfully.
400	Bad request. This indicates request validation error. Consult returned error message for more details.
401	Unauthorized. Session or authorization token has expired and must re-authenticate
402	Payment required. This feature is not available to general public. Please contact Reseller to enable this feature.
403	Not allowed to access this resource due insufficient access permission.
404	Resource not found.
405	The requested resource does not support the requested request method.
406	The resource does not support the requested format indicated in Accept header.
409	The submitted request is in conflict with existing resource, ie: Duplicate unique identifier.
415	The submitted media type is not supported.
429	Too many requests. Please slow down.

Appendix D - Error Codes

Error Code	Description
Request	
30	Invalid URL
31	Invalid Content-Type header value
32	Improperly formatted JSON request body
33	Invalid Accept header value
34	Invalid request method
35	AIQ-USER-ID header not found
36	Invalid AIQ-Device-Type header
37	Too many requests
Authentication	
50	Invalid user id and/or password
51	User account has been deactivated
52	Invalid app key and/or secret
53	App has been disabled
Authorization	
61	Authorization token in not valid
62	Not allowed to access the requested resource due to insufficient access permission
63	Not allowed to access the requested resource due to subscription limitation
64	Missing authorization token
Subscription	
70	Subscription has expired
71	Quota limitation exceeded
72	Video length exceeds limit
Resource Specific	
100	Validation error; Refer to detail section in response body for more information
101	The requested resource does not exist
102	The submitted resource already exists
103	The submitted request body is empty
104	Resource is in use

105	Resource cannot be deleted
106	Invalid media mime type
220	Resolution too low
221	Similar media already exists
222	Trigger already has media attachment
999	Method is not implemented