2023

CAPTIS INTELLIGENCE

REST API Intro

REV 013

1 Table of Contents

2 Abstract

This document describes our APIs.

Generally:

1. They are exposed over HTTPS
2. They follow a REST style
3. They mostly marshal request/responses as JSON
4. They offer uploading of single files as an inline single-part body
5. They return standard status codes
6. They implement a life cycle that does include EOL

For each endpoint, we include a description of the function, request, response, and status codes.

If you find this documentation to be incorrect, or needing improvement, please let us know.

3 Security

Most APIs require authentication.

The default authentication method requires all calls to specify an Access Key.

The APIs accept the Access Key as a URL parameter named **accessKey**, also referenced in the specific API documentation for each endpoint.

The Access Key logically represents a User or an Organization, and can be obtained via:

1. SAC | Edit User
2. SAC | Edit Org
3. Captis staff

4 Dictionary

4.1 Entities

This section provides rough examples of the object representation for the entities required to interact with these services.

4.1.1 Enterprise

Body	Validation
{ "id": "b18b4f24-a737-4a7b-a249-b7030be826cc", "name": “ABC Company”, "address": "123 Main St, AnyTown, 12345, USA" }	

4.1.2 User

Body	Validation
{ "id": "262c4c27-b6e2-440e-8982-223bd56b3de6", "name": “John Smith”, "email": john.smith@abccompany.com, "type": “ADMIN” }	

4.1.3 Scan

Body	Validation
{ "id": "67b3b847-b524-4d47-a664-28a85bf54311", "status": "COMPLETED", "searchStatus": "COMPLETED", "photo": {photoBase64}, "photoName": "652cfebc928abfb79580608.gif",	

"matchCount": 11 }	

4.1.4 Match

Body	Validation

{
"match": {
"id": "aec4a5ed-3e14-470d-a987-78c2d3984c53",
"score": 98.099754,
"scoreLevel": "HIGH"
},
"subject": {
"id": "5d10bef567f6f41880a7c4b4",
"name": "GORGODIAN, PAUL",
"type": "CRIMINAL",
"photo": {photoBase64}
}	

5 Appendix

5.1 Enums

5.1.1 Processing Status

PENDING
ALLOCATED
COMPLETED
FAILED
INTERRUPTED
BAD

Entities use fields of type “Processing Status” to indicate at which point of the workflow processing has progressed.

For example, Scans are created with a status of PENDING, change to ALLOCATED when actively being processed, and change to COMPLETED when they are done.

In case of failure, they change to FAILED.

If interrupted for any reason, they change to INTERRUPTED.

If the input data was accepted but somehow found to be corrupted, they change to BAD.

Some responses include a “done” flag, to indicate that the case has reached a final state.

6 Response Codes

Response Code	Description	Examples

200	The request was accepted, processed inline, and a response was provided.	{
		"id" : "67b3b847-b524-4d47-a664-28a85bf54311",
		“provisionedUrl”: “https://www.solveacrime.com/”,
		}

202	The request was accepted and scheduled for asynchronous processing. This typically returns an ID that can be used to acquire the response later.	{
		"id" : "67b3b847-b524-4d47-a664-28a85bf54311",
		“eta”: “45”,
		}

307	Indicates that the request was not processed, and that the caller should try this request and all subsequent requests against another URL for the given amount of time.
	
	It is imperative that the caller respect this directive. Failure results in the offending account being suspended.
	
	You can specify the “simulateRedirect” URL parameter to have the server simulate a 307 response.
	
	Warning: many HTTP libraries offer the ability to follow redirects, but they do so only for one call. This 307 returns an “expires” directive indicating that this and all subsequent requests must be made against the new baseUrl.
		{
		"requestId": "209e8167",
		"baseUrl": "http s: //api-asi2.solveacrime.com:443",
		"expires": "2023-11-04T16:08:16-0700"
		}

400	Indicates that the URL path, URL parameter, and/or request body have insufficient or incorrect data to fulfill the request.	If the Access Key is not valid:
		{
		"dateTime": "2018-12-18T13:27:54Z",
		"message": "error.400",
		"description": "Access Key is not valid",
		"fieldErrors": null
		}
		
		If the scan name is not valid:
		{

		"dateTime": "2019-01-09T10:49:56Z",
		"message": "error.validation",
		"description": null,
		"fieldErrors": [
		{
		"objectName": "ResolveRequest",
		"field": "case.name",
		"violation": "Name",
		"message": "Name should be 0-64 characters long"
		}
		]
		}

401	The request was not processed because the accessKey is bad.	{
		"dateTime": "2018-10-22T05:57:14Z",
		"message": "error.unauthorized",
		"description": "Bad Access Key ",
		"fieldErrors": null
		}

403	The request was not processed because the caller credentials are good, but the caller lacks required privileges to execute this request.	{
		"dateTime": "2019-01-11T14:46:21Z",
		"message": "error.accessDenied",
		"description": "Access Key is disabled",
		"fieldErrors": null
		}

404	Indicates that the object being accessed does not exist.	{
		"dateTime": "2018-10-30T09:23:35Z",
		"message": "error.404",
		"description": "Scan 719729c3-a372-4d9d-85f1-1ba8166a2037 not found.",
		"fieldErrors": null
		}

405	Indicates that the given endpoint does not support the HTTP method used to make this request.	
409	Indicates that the request does not match the expected version.	

500	Indicates an internal server error. We are automatically notified about these errors, and either fix the underlying problem, or improve the error response if this is due to an issue with the request.	{
		"dateTime": "2019-01-11T15:04:24Z",
		"message": "error.internalServerError",
		"description": "There was a server error. Please try your request again.",
		"fieldErrors": null
		}

503	Service Unavailable: The server cannot handle the request (because it is overloaded or down for maintenance).	

	Generally, this is a temporary state.	