

API for image verification

Default

verifyImagesZipFilePost

Send a ZIP file containing images for processing

POST

/verifyImages/zipFile

Usage and SDK Samples

Python

```
import time
import requests

def verifyImages(url, file_path, x_api_key, case_number, case_date=None):
    # Open the zip file in binary mode
    with open(file_path, 'rb') as file:
        # Prepare the file data for posting
        files = {'file': ('images.zip', file, 'application/zip')}

        # Prepare headers
        headers = {'x-api-key': x_api_key}

        # Prepare payload
        data = {'caseNumber': case_number}
        if case_date:
            data['caseDate'] = case_date

        # Send the request
        try:
            response = requests.post(url, files=files, headers=headers, data=data)
            response.raise_for_status() # Raise an exception for HTTP errors
        except requests.exceptions.RequestException as e:
            print("Error:", e)
            return

        # Check response status
        if response.status_code == 200:
            print("Server response:")
            print(response.json())
        else:
            print("Failed to upload file.")
            print("Status Code:", response.status_code)
            print("Server response:", response.json())
```

Parameters

Header parameters

Name	Description
x-api-key*	String API key for authentication Required

Body parameters

Name	Description
body *	<div>▼ {</div> <div>Required: caseNumber,x-api-key,zipFile</div> <div>zipFile: ▼ string (binary) ZIP file containing images</div> <div>caseNumber: ▼ string Case number for tracking</div> <div>caseDate: ▼ string (date-time) Date of the case</div> <div>}</div>

Responses

Status: 200 - ZIP file successfully processed

Schema

▼ {

caseNumber: ▼ [

▼ {

imageId1: ▼ {

imageQuality: boolean

generatedDetection: ▼ {

confidence: number

predictedClassName: string

}

tamperedDetection: ▼ {

confidence: number

predictedClassName: string

}

doubletCheck: ▼ {

result: boolean

intern: boolean

similarityPercentage: number

}

reverseSearch: ▼ {

result: boolean

matches: ▼ [

▼ [

string (uri)

]

]

}

metadata: ▼ {

exifData: ▼ {

}

FileType: string

Mode: string

Width: integer

Height: integer

error: string

}

}

}

]

}

Status: 400 - Invalid request
