Vaarhaft API

API and SDK Documentation

Version: 3.0.0

API for image verification

Use the following base link for the API:
`https://api.vaarhaft.com`

Maximum upload size is 10 MB.

fraudScannerPost

Send a ZIP file containing images for processing

POST

/fraudScanner

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} # 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 *	{ Required: caseNumber,x-api-key,zipFile zipFile: string (binary) *ZIP file containing images* caseNumber: string *Case number for tracking* caseDate: string (date-time) *Date of the case* }

Responses

Status: 200 - ZIP file successfully processed

Schema

```
{
  caseNumber: [
    {
      image1: {
        imageQuality: {
          result: boolean
          error: boolean
          enabled: boolean
          description: This is an upstream check that checks whether the image meets certain minimum requirements. (resolution, sharpness ...)
        }
        generatedDetection: {
          confidence: number
          predictedClassName: string
          error: boolean
          enabled: boolean
          description: Images that have been fully generated by an artificial intelligence are recognized with this feature. The predictedClass can be either 'gen' (generated) or 'real' (Real). The confidence indicates how certain our model is.
        }
        tamperedDetection: {
          confidence: number
          predictedClassName: string
          error: boolean
          enabled: boolean
          description: Images that have been subsequently processed by an AI or other software are recognized with this feature. The predictedClass can assume either 'tp' (tampered) or 'real' (Real). The confidence indicates how certain our model is.
        }
        doubletCheck: {
          result: boolean
          intern: boolean
          similarityPercentage: number
          error: boolean
          enabled: boolean
          description: With the similarity comparison, the Fraud Scanner checks whether the analyzed image has already been submitted to you or another insurance company in our database.
        }
        reverseSearch: {
          result: boolean
          matches: [
            string (url)
          ]
          error: boolean
          enabled: boolean
          description: This feature checks whether the submitted image originates from the Internet and has already been uploaded there once. If the image was found on the Internet, we also return the links to the websites found.
        }
        metadata: {
          exifData: [
            {}
          ]
          FileType: string
          Mode: string
          Width: integer
          Height: integer
          error: boolean
          enabled: boolean
          description: The metadata for each image is extracted and displayed in an organized manner so that you can quickly obtain all important additional information about the image. This is a rudimentary check that checks existing metadata.
        }
      }
    }
  ]
}
```

Example Result


```
{
  "statusCode": 200,
  "body": {
    "TestCase": {
      "stable3_0111c29e-cd60-42e4-9c3a-31093a210603.png": {
        "imageQuality": {
          "result": true,
          "error": false
        },
        "doubletCheck": {
          "result": true,
          "caseNumber": "erv",
          "intern": false,
          "error": false,
          "enabled": true
        },
        "reverseSearch": {
          "result": false,
          "matches": [],
          "error": false,
          "enabled": false
        },
        "metadata": {
          "analysed": {
            "creationDate": {
              "cDate": "-",
              "cTime": "-",
              "isSus": false,
              "diffInDays": 0,
              "isTodayUsed": false,
              "refDate": "-"
            },
            "imgRanking": null,
            "fieldsMarkedSus": {}
          },
          "GPSInfo": {},
          "raw": {
            "Rating": {},
            "Dates": {},
            "GPS": {},
            "Other": [
              {},
              {
                "Image width": "1024 pixels",
                "Image height": "1024 pixels",
                "Bits/pixel": "24",
                "Pixel format": "RGB",
                "Compression rate": "1.3x",
                "Compression": "deflate",
                "MIME type": "image/png",
                "Endianness": "Big endian"
              }
            ]
          },
          "error": false,
          "enabled": true
        },
        "generatedDetection": {
          "predictedClassName": "gen",
          "confidence": 0.9999997615814209,
          "error": false,
          "enabled": true
        },
        "tamperedDetection": {
          "predictedClassName": "real",
          "confidence": 0.9994547347868388,
          "error": false,
          "enabled": true
        }
      }
    }
  },
  "headers": {
    "Access-Control-Allow-Headers": "Content-Type,X-Amz-Date,Authorization,X-Api-Key,X-Amz-Security-Token",
    "Access-Control-Allow-Methods": "OPTIONS,POST",
    "Access-Control-Allow-Origin": "*"
  },
  "sessionId": "065a9dbc-aeeb-4a0f-890a-574fe03b5f08"
}
```