

# How to use Threat Emulation API via the Cloud

## Purpose:

- Explain how to use the Cloud Threat Emulation API with the cloud services

## Contents

Basic Overview .....	2
Link to Documentation .....	2
TE API .....	2
Eval License .....	2
How to find the MD5, sha1 or sha256 of a file .....	4
Cloud TE API via Ubuntu using Curl—query .....	5
Cloud TE API via Ubuntu using Curl—upload .....	7
Cloud TE API via Ubuntu using Curl—download .....	9
Cloud TE API via Ubuntu using Curl—quota .....	10
Install Postman .....	11
Cloud TE API via Postman – query .....	11
Cloud TE API via Postman – upload .....	12
Cloud TE API via Postman – download .....	14
Cloud TE API via Postman – quota .....	15

## Basic Overview

- To evaluate files with the Cloud, you will need an Evaluation API Key

## Link to Documentation

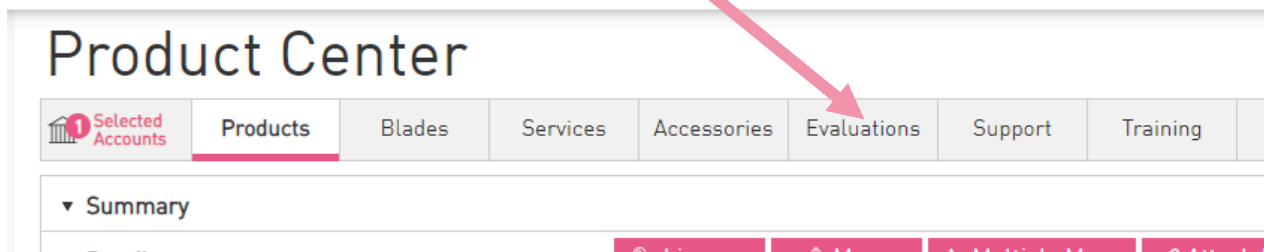
- [Threat API Guide](#)
- Threat Prevention API for Security Gateway [SK137032](#)
- Harmony Browse or Threat Prevention API – working with Security Gateway or SandBlast Threat Emulation appliance [SK113599](#)

## TE API

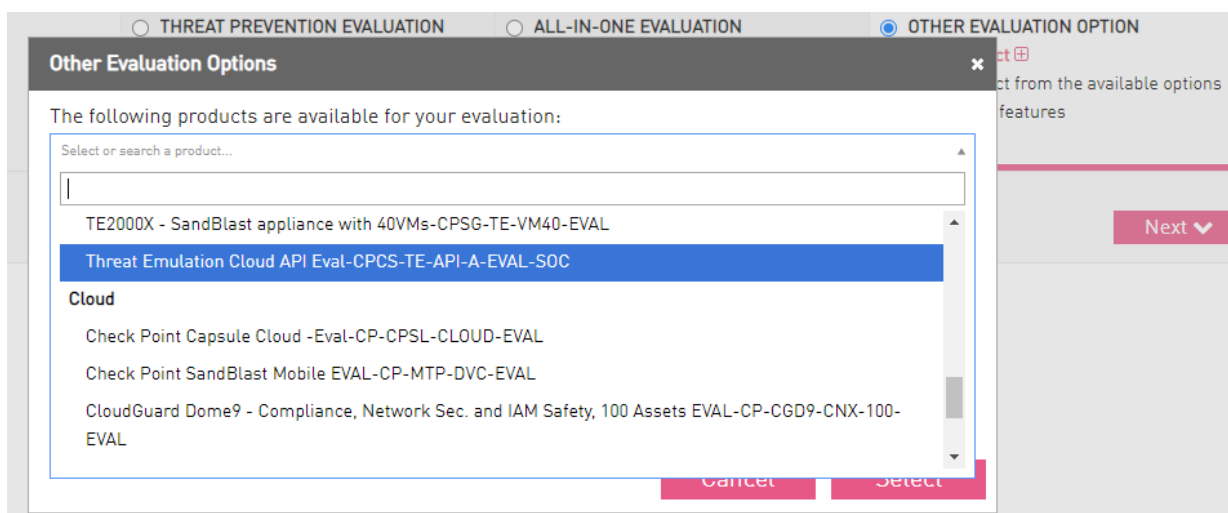
- Uses HTTP Post Method
- Body of requests and responses are in JSON format
- Access the API with the URL
  - [https://<Service\\_address>/tecloud/api/<version>/file/<API\\_name>](https://<Service_address>/tecloud/api/<version>/file/<API_name>)
- API Name, can be query, upload, download or quota (cloud only)

## Eval License

- Logon to User Center (UC)
- Open “My Products”
- Click on Evaluations Tab



- Select Product Evaluations
- Select Other Evaluation Option
- Select “Threat Emulation Cloud API Eval-CPCS-TE-API-A-EVAL-SOC



- Finish the steps to put the eval in your UC or the customer's UC
- Once it is done, go back to "Product Center", or use the "click here" Link

To license your evaluation product and enjoy a 30-day free trial:

1. Sign in to Check Point User Center with the User Center credentials.
2. Go to *Product Center* and select the following account:  
Name: JPAUL-SE  
Account ID: 0006798454
3. Select the following product in the *Evaluations* tab:  
Name: Threat Emulation Cloud API Eval  
SKU: CPCS-TE-API-A-EVAL-SOC  
Certificate Key: 6D14339EAE82
4. Select the "license" option and follow the licensing instructions.

To view this evaluation directly in Product Center, [click here](#).



- The eval will be listed under the Product Tab in the UC
- Once you are ready, click the Generate New Key, this will be the API authorization key you will use during your demo or POC, copy this key down to use later

## Threat Emulation Cloud Service API

### Overview

Threat Emulation Cloud API Eval | SKU: CPCS-TE-API-A-EVAL-SOC  
 Capacity: 10000 | Scanned: 0 | Emulated: 0  
 Valid from 24-Jun-2021 to: 23-Aug-2021 | Purchased on: 24-Jun-2021

### 1 Keys Generated - Details ▲

Filter as you type	GENERATE NEW KEY
Key	Issue Date
  TE_API_KEY_nYkztwqdrw1iUmjMLgg0wf [REDACTED]	24-Jun-2021
Showing 1 to 1 of 1 entries	

### Activity Summary

#### No Activity Information

Please verify that Threat Emulation is activated on one of the selected gateways.  
 For further assistance please contact Check Point [Account Services](#).

- Note the upper right area, shows you how many days left to use and how much quota has been used.

**59 Days Left** ●  
 Until renewal date: 23-Aug-2021

**0% Quota Used** ●  
 In Jun 2021

Status
Active

## How to find the MD5, sha1 or sha256 of a file

- On a windows machine, in file explorer right click on the file an select CRCSHA
  - If you select the \* option it will come back with all the SHA1 & SHA256

**Checksum information**

Name	InstructorInformation-2021.pdf
Size	437773 bytes (427 KiB)
CRC32	4B41C47F
CRC64	946C39EBA4EDCFFD
SHA256	9D93AC289B124BD071E961CC576D7C1A5FB7C38E1F031A72CCC69FF018AAAA63
SHA1	91D05DDB35F1B9080856A1E8E36F8C08F620874B
BLAKE2sp	A88750A98685209BD54F59F7BC6E7721280B6524815238036B9D344BA748F65D

- To get the MD5 at a command line in windows type `c:\CertUtil -hashfile filename.exe MD5`

```
C:\api>CertUtil -hashfile Movie.xlsx MD5
MD5 hash of Movie.xlsx:
59d33e1334d1fd6e9801be7dcb0e4c85
CertUtil: -hashfile command completed successfully.
```

- On a Linux machine type in `sha1sum`, or `sha256sum` or `md5sum` and the file name

```
julie@odo-1:~$ sha256sum Movie.xlsx
67a9a93dfddca6084c85b48b8e14cbaeb0a8488b719b97513defd26f643381b0  Movie.xlsx
julie@odo-1:~$ shasum Movie.xlsx
98e7dcca5669f38da478d9020ba325e0e58d9ade  Movie.xlsx
julie@odo-1:~$ md5sum Movie.xlsx
59d33e1334d1fd6e9801be7dcb0e4c85  Movie.xlsx
julie@odo-1:~$
```

## Cloud TE API via Ubuntu using Curl—query

Query can be used to see if the file is already known by the threat cloud. Many times this is done first before uploading the document. The result will come back with a verdict and if it is not known it will send a message saying “Could not find the requested file. Please upload it.”

- Queries require the following elements:
  - HTTP **POST**: **https://<service\_address>/tecloud/api/<version>/file/query**
  - Header with the Authorization: **API key**
  - Header with the Content-Type of **application/json**
  - The raw data of the json request, which at a minimum should contain the **sha1, sha256 or md5** of the file.
  - Optional fields for the query are:
    - File\_type
    - File\_name
    - Features
    - Quota
    - NOTE, te & extraction have other optional parameters, see the online guide for more
- Simple query example below all in one line using curl on a linux box. Note the color coding to match the required elements noted above.

```
curl --location --request POST 'https://te.checkpoint.com/tecloud/api/v1/file/query' --header
'Authorization: TE_API_KEY_tlj190qwizG5neHMhpFCoeJvOJLQeXNY0*****' --header 'Content-Type:
application/json' --data-raw '{"request": [{"sha1":
"98E7DCCA5669F38DA478D9020BA325E0E58D9ADE"]}'
```

- The TE API key is obtained from your Eval license and the sha1, sha256 or md5 are of the file being queried.
- Below is the same example with the same json data stretched out to make it more readable.

```
curl --location --request POST 'https://te.checkpoint.com/tecloud/api/v1/file/query' --header
'Authorization: TE_API_KEY_tlj190qwizG5neHMhpFCoeJvOJLQeXNY0*****' --header 'Content-Type:
application/json' \
--data-raw '{
  "request": [
    {
      "sha1": "98E7DCCA5669F38DA478D9020BA325E0E58D9ADE"
    }
  ]
}'
```

- Below is the same example with some of the extra items included, note you do not need all of these to do a query.

```
curl --location --request POST 'https://te.checkpoint.com/tecloud/api/v1/file/query' --header
'Authorization: TE_API_KEY_tlj190qwizG5neHMhpFCoeJvOJLQeXNY0*****' --header 'Content-Type:
application/json' \
--data-raw '{
"request": [
{
"sha1": "98E7DCCA5669F38DA478D9020BA325E0E58D9ADE",
"features": [
"te",
"av",
"extraction"
],
"file_name": "Moviel.xlsx",
"te": {
"reports": [
"xml",
"summary"
],
"extraction": {
"method": "pdf"
}
}
}
]
}'
```

- Example of the query response below for a benign file:

```
{
"response": [
{
"status": {
"code": 1001,
"label": "FOUND",
"message": "The request has been fully answered."
},
"sha1": "e2bf99a60456521515e43a8985b092d71cee7319",
"file_type": "pdf",
"file_name": "",
"features": [
"te"
],
"te": {
"trust": 0,
"images": [
{
"report": {
"verdict": "benign"
},
"status": "found",
"id": "e50e99f3-5963-4573-af9e-e3f4750b55e2",
"revision": 1
},
{
"report": {
"verdict": "benign"
},
"status": "found",
"id": "5e5de275-a103-4f67-b55b-47532918fa59",
"revision": 1
}
]
}
}
]
```



Upload can be used to send a file to be emulated and extracted to the cloud.

- Create a directory with files on your Server
- Place files to demo or poc in the folder
- Gather the MD5 or SHA1 of the file to be uploaded
- Uploads require the following elements:
  - HTTP POST: `https://<service_address>/tecloud/api/<version>/file/upload`
  - Header with the Authorization: `API key`
  - Header Form with the `file`
  - Optional fields for the query are:
    - File\_type
    - File\_name
    - Md5, sha1 or sha256
    - Features
- Example of the upload with just the minimum's required:

```
curl --location --request POST 'https://te.checkpoint.com/tecloud/api/v1/file/upload' --header 'Authorization: TE_API_KEY_nYkztwqdrw1iUmjMLgg0wFJ1kky1jn*****' --form 'file=@/home/julie/Hibernate.pdf'
```

- Example of the response when the file is unknown.

```
{
  "response": {
    "status": {
      "code": 1002,
      "label": "UPLOAD_SUCCESS",
      "message": "The file was uploaded successfully."
    },
    "sha1": "e2bf99a60456521515e43a8985b092d71cee7319",
    "md5": "b774cfa2cb5cfe49bcea682cc9ec54d2",
    "sha256": "a7e5a2c481a3d7ec7bab510229a699dcb8c084882c63f9e005fbab465689359e",
    "file_type": "",
    "file_name": "Hibernate.pdf",
    "features": [
      "te"
    ],
    "te": {
      "trust": 0,
      "images": [
        {
          "report": {
            "verdict": "unknown"
          },
          "status": "not_found",
          "id": "e50e99f3-5963-4573-af9e-e3f4750b55e2",
          "revision": 1
        }
      ],
      {
        "report": {
          "verdict": "unknown"
        },
        "status": "not_found",
        "id": "5e5de275-a103-4f67-b55b-47532918fa59",
        "revision": 1
      }
    ],
    "score": -2147483648,
    "status": {
```



```

    "code": 1002,
    "label": "UPLOAD_SUCCESS",
    "message": "The file was uploaded successfully."
  }
}
}

```

- Example of the file uploaded and is known.

```

{
  "response": {
    "status": {
      "code": 1001,
      "label": "FOUND",
      "message": "The request has been fully answered."
    },
    "sha1": "e2bf99a60456521515e43a8985b092d71cee7319",
    "md5": "b774cfa2cb5cfe49bcea682cc9ec54d2",
    "sha256": "a7e5a2c481a3d7ec7bab510229a699dcb8c084882c63f9e005fbab465689359e",
    "file_type": "pdf",
    "file_name": "Hibernate.pdf",
    "features": [
      "te"
    ],
    "te": {
      "trust": 0,
      "images": [
        {
          "report": {
            "verdict": "benign"
          },
          "status": "found",
          "id": "e50e99f3-5963-4573-af9e-e3f4750b55e2",
          "revision": 1
        },
        {
          "report": {
            "verdict": "benign"
          },
          "status": "found",
          "id": "5e5de275-a103-4f67-b55b-47532918fa59",
          "revision": 1
        }
      ],
      "score": -2147483648,
      "combined_verdict": "benign",
      "status": {
        "code": 1001,
        "label": "FOUND",
        "message": "The request has been fully answered."
      }
    }
  }
}

```

## Cloud TE API via Ubuntu using Curl--download

Download can be used to retrieve the summary report for Threat Emulation if it is Malware or the Extracted file for Threat Extraction.

- Downloads require the following elements:
  - HTTP POST: [https://<service\\_address>/tecloud/api/<version>/file/download?id=<id>](https://<service_address>/tecloud/api/<version>/file/download?id=<id>)

- Header with the Authorization: API key
- ID information
- Example below:

```
curl --location --request POST 'https://te.checkpoint.com/tecloud/api/v1/file/download?id=5e5de275-a103-4f67-b55b-47532918fa59' --header 'Authorization: TE_API_KEY_nYkztwqdrw1iUmjMLgg0wFJ1kky1jn*****'
```

- The id is pulled from the report information for Threat Emulation below—note you have a Summary Report ID below:

```
te:
  combined_verdict: "malicious"
  severity: 4
  confidence: 3
  summary_report: "8cf485d8-d095-4487-ba36-a887d20fc83a"
  images:
    - report:
        verdict: "malicious"
        xml_report: "ef5f38d8-c35e-42fa-b3f1-388e681e18b9"
        status: "found"
        id: "5e5de275-a103-4f67-b55b-47532918fa59"
        revision: 1
    - report:
        verdict: "malicious"
        xml_report: "c7486ce7-9cde-484d-9ba4-bfc51fd88f99"
        status: "found"
        id: "7e6fe36e-889e-4c25-8704-56378f0830df"
        revision: 1
  status:
    code: 1001
    label: "FOUND"
    message: "The requested data has been found"
  av:
```

- The Extracted file download ID is noted below as an example:

```
extraction:
  method: "pdf"
  extract_result: "CP_EXTRACT_RESULT_SUCCESS"
  extracted_file_download_id: "82f67772-2116-4d29-a5be-245a434af2ae"
  output_file_name: "MyFile.docx.pdf"
  time: "1.374"
  extract_content: "Database Queries"
  extraction_data:
    input_extension: "docx"
    input_real_extension: "xls"
    message: "OK"
    orig_file_url: ""
    output_file_name: "MyFile.cleaned.xls.pdf"
    protection_name: "Potential malicious content extracted"
    protection_type: "Conversion to PDF"
    risk: 2
    scrub_activity: "Active content was found - XLS file was converted to PDF"
    scrub_method: "Convert to PDF"
    scrub_result: 0
    scrub_time: '1.374'
    scrubbed_content: "Database Queries"
  tex_product: false
```

## Cloud TE API via Ubuntu using Curl—quota

Quota is used to retrieve the current license and quota stats of your API key.

- Downloads require the following elements:
  - HTTP POST: `https://<service_address>/tecloud/api/<version>/file/quota`
  - Header with the Authorization: `API key`
- Example below:

```
curl --location --request POST 'https://te.checkpoint.com/tecloud/api/v1/file/quota' --  
header 'Authorization: TE_API_KEY_nYkztwqdrw1iUmjMLgg0wFJ1kky1jn*****'
```

- Example Response:

```
{  
  "response": [  
    {  
      "remain_quota_hour": 499,  
      "remain_quota_month": 9994,  
      "assigned_quota_hour": 500,  
      "assigned_quota_month": 10000,  
      "hourly_quota_next_reset": "1625083200",  
      "monthly_quota_next_reset": "1625097600",  
      "quota_id": "54T1341",  
      "cloud_monthly_quota_period_start": "1622505600",  
      "cloud_monthly_quota_usage_for_this_gw": 6,  
      "cloud_hourly_quota_usage_for_this_gw": 1,  
      "cloud_monthly_quota_usage_for_quota_id": 6,  
      "cloud_hourly_quota_usage_for_quota_id": 1,  
      "monthly_exceeded_quota": 0,  
      "hourly_exceeded_quota": 0,  
      "cloud_quota_max_allow_to_exceed_percentage": 1000,  
      "pod_time_gmt": "1625082228",  
      "quota_expiration": "1629676800",  
      "action": "ALLOW"  
    }  
  ]  
}
```

## Install Postman

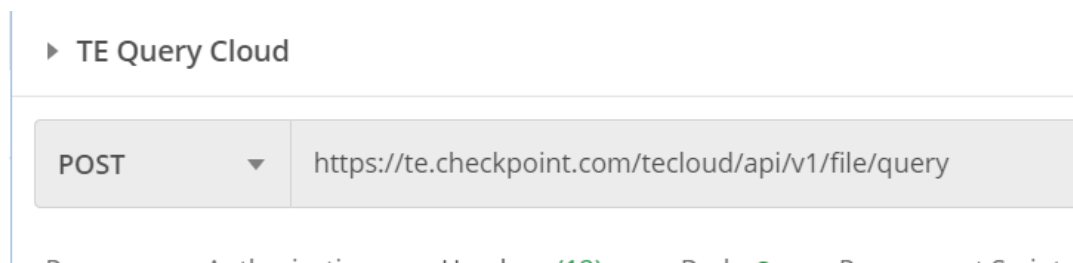
Postman is used by many as a testing platform for developers before they move it into production.

- Download and install Postman
  - <https://www.postman.com/downloads/>
- Once Installed, Click New Collection, call it TE API

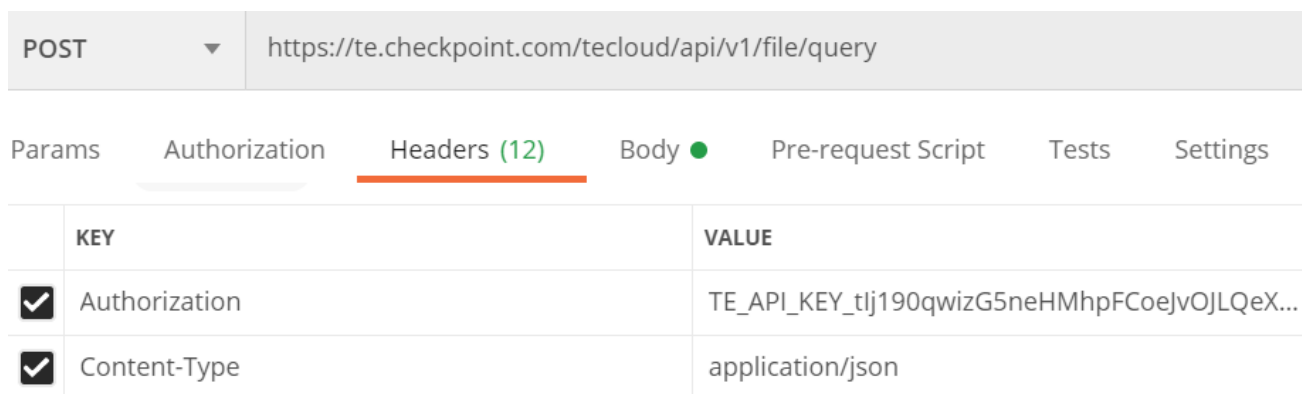
## Cloud TE API via Postman – query

- In the collection select the + to create a new item—save as TE Cloud Query
- Change the Get to Post
- Ensure you use the query URL
  - <https://te.checkpoint.com/tecloud/api/v1/file/query>





- Headers for a query are:
  - Authorization with the value of your TE Key
  - Content-Type is now application/json

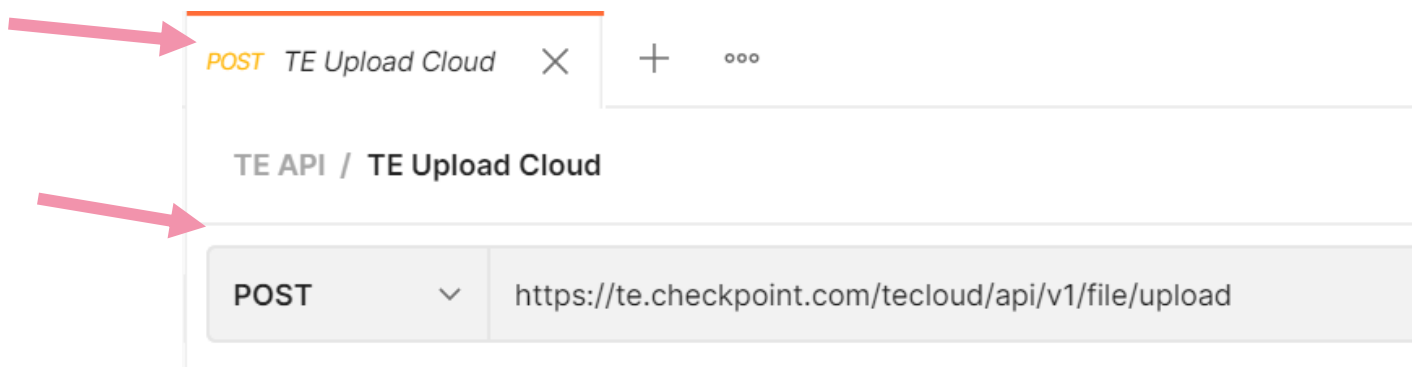


- Body, this time is raw with the following text
  - Remember to replace the sha1 or md5 to the file you want to know if known or to find out a response to a file that has been uploaded

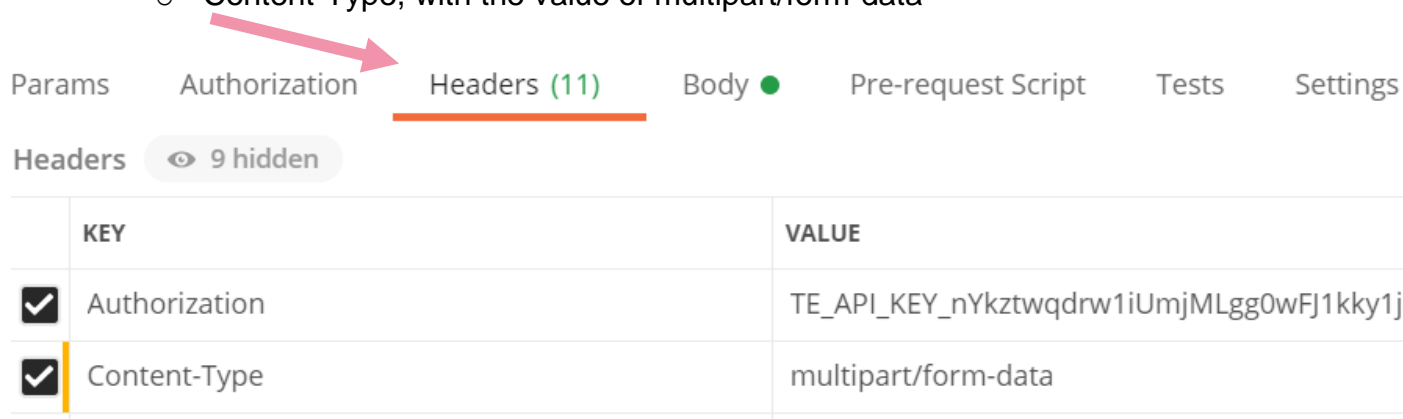
```
{
  "request": [
    {
      "features": ["te"],
      "sha1": "E2BF99A60456521515E43A8985B092D71CEE7319",
      "te": {
        "reports_version_number": 2,
        "reports": [
          "tar"
        ]
      }
    }
  ]
}
```

## Cloud TE API via Postman – upload

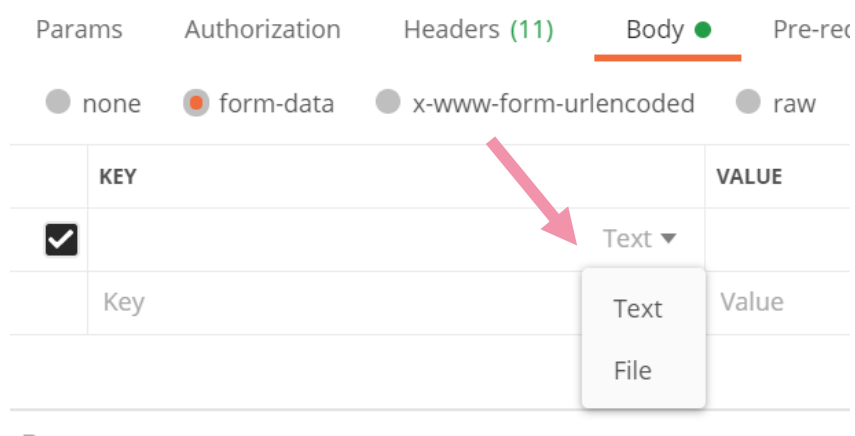
- In the collection select the + to create a new item
- Change the Get to Post
- Save as TE Cloud Upload
- Enter the URL to: <https://te.checkpoint.com/tecloud/api/v1/file/upload>



- Click on Headers and add the two following keys:
  - Authorization, with the value of your API key
  - Content-Type, with the value of multipart/form-data



- Click on Body, change to form-data
  - Under the Key, change to File



- In the Value, select a file to upload—click Send



/api/v1/file/upload

Send

Pre-request Script Tests Settings

raw binary GraphQL

VALUE	DESCRIPTION
VPNChange.docx	

- Example response

file VPNChange.docx

Body Cookies (1) Headers (8) Test Results Status: 200 200 Time: 855 ms Size: 1.36 KB

Pretty Raw Preview Visualize JSON

```
1 {
2   "response": {
3     "status": {
4       "code": 1002,
5       "label": "UPLOAD_SUCCESS",
6       "message": "The file was uploaded successfully."
7     },
8     "sha1": "be1f7d6d57a4fce897ba0fd371520503b012dc49",
9     "md5": "53e83b82ffd6944c1ccc05dae3bc3478",
10    "sha256": "4bfe0af16df7637cefb42a39c777f0128e9f1d8b61f00b9a5487360bba7a99a7",
11    "file_name": ""
12  }
13 }
```

## Cloud TE API via Postman – download

- In the collection select the + to create a new item
- Change the Get to Post
- Save as TE Cloud download
- Enter the URL to: <https://te.checkpoint.com/tecloud/api/v1/file/download>

TE API / TE Download Cloud

POST https://te.checkpoint.com/tecloud/api/v1/file/download

- Headers keys are:
  - Authorization with the value of your TE Key

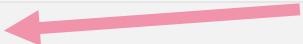
POST ▼ https://te.checkpoint.com/tecloud/api/v1/file/download

Params Authorization **Headers (10)** Body Pre-request Script Tests Settings

Headers 👁 8 hidden

	KEY	VALUE
<input checked="" type="checkbox"/>	Authorization	TE_API_KEY_nYkztwqdrw1iUmjMLgg0wFJ...
<input checked="" type="checkbox"/>		

- Params for the query are ID with the value found in the query response
  - Enter the value from your query response
  - NOTE: Once you add the Key of "id", it will change the url post, this is correct

POST ▼ https://te.checkpoint.com/tecloud/api/v1/file/download?id 

Params ● Authorization **Headers (10)** Body Pre-request Script Tests Settings

Query Params

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	id		
	Key	Value	Description

- In the example query response below, you would use the extracted\_file\_download\_id

```

extraction:
  method: "pdf"
  extract_result: "CP_EXTRACT_RESULT_SUCCESS"
  extracted_file_download_id: "82f67772-2116-4d29-a5be-245a434af2ae"
  output_file_name: "MyFile.docx.pdf"
  
```

## Cloud TE API via Postman – quota

- In the collection select the + to create a new item
- Change the Get to Post
- Save as TE Cloud download
- Enter the URL to: <https://te.checkpoint.com/tecloud/api/v1/file/quota>
- Headers Keys are:
  - Authorization with the value of your TE Key



POST



https://te.checkpoint.com/tecloud/api/v1/file/quota

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Headers

8 hidden

	KEY	VALUE
<input checked="" type="checkbox"/>	Authorization	TE_API_KEY_nYkztwqdrw1iUmjMLgg0wFJ1...