Annotation API Documentation – V1.0

1.1.1 Versions

- V 1.0 – **Released**. The first version of the Threat Annotation API supports uploading a video clip and downloading the annotated video.

1.1.2 Endpoints

The API has basically 3 endpoints

1. Annotate endpoint (/annotate)
    - The POST endpoint to which the video file that need to be annotated is sent to along with preferred annotation (emo, behav, threat)
    - Accepts - form-data
    - Returns-JSON
2. Poll endpoint (/poll)
    - This GET endpoint accepts polling for annotated video and returns with unique id that has been approved with download
    - Returns – JSON
3. Download endpoint
    - This GET endpoint that allows the used with a unique id to download the annotated file

1.1.3 Communication Flow

Here first the user or any other endpoints, send the video. Live stream to a web application/mobile application.

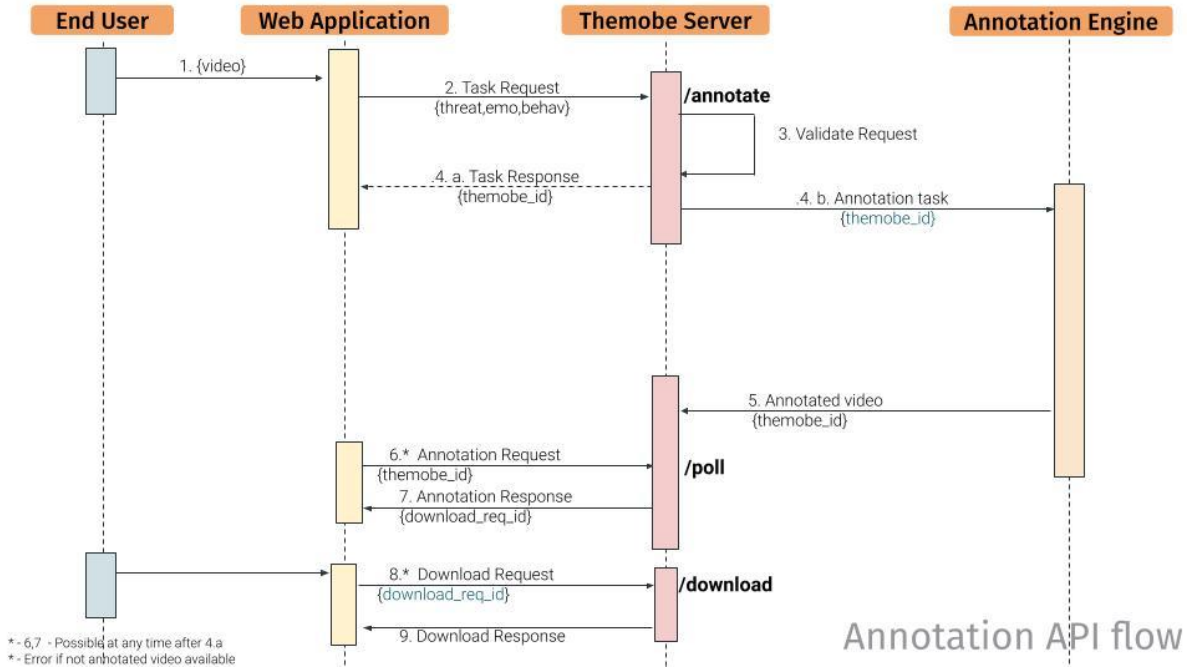

*Figure **Error! No text of specified style in document.**.1 Communication Flow*

- **Task Request:**
  Then the particular application creates a Task request. Task request should be a POST request with a video file in the body and necessary to mention what type of annotation is expected.
  - emo:
    - Boolean parameter that states whether emotional annotation is necessary.
    - Possible values: true, false
  - behav
    - Boolean parameter that states whether behavioral annotation is necessary.
    - Possible values: true, false
  - threat
    - Boolean parameter that states whether threat annotation is necessary.
    - Possible values: true or false
  - video
    - the actual video file that need to be annotated
    - **.mp4** is the accepted format. Returns error otherwise
  - expires_in -
    - Optional parameter which defines the durability of the task or the lifetime of theombe_id.
    - Maximum (default) value is 900 sec.
  - persistent_status
    - Parameter that states the user consent on allowing the server to allow saving the annotated file (only for research purposes)
    - Possible values: true or false

  Sample Task Request:

```
"name": "Task Request",
"request": {
        "method": "POST",
        "header": [],
        "body": {
                "mode": "formdata",
                "formdata": [
                        {
                                "key": "video",
                                "type": "file",
                                "src":
"/C:/Users/cvvin/Desktop/testvideo1..mp4"
                        },
```

```
                        {
                                "key": "emo",
                                "value": "False",
                                "type": "text"
                        },
                        {
                                "key": "behav",
                                "value": "False",
                                "type": "text"
                        },
                        {
                                "key": "threat",
                                "value": "True",
                                "type": "text"
                        },
                        {
                                "key": "expires_in",
                                "value": "10",
                                "type": "text",
                        }
                ]
        },
        "url": "http://127.0.0.1:5000/annotate"
}
```


- **Task Response**
  One the Task request is validated Themobe server responds with 200 OK as task response indicating the task request is accepted and put to work.
  - Themobe id:
    - A unique identifier that will be sent in Task Response for further transactions of polling to check whether the provided task (or video) is annotated and available to download

  - interval:
    - Parameter that defines the polling interval which is by default 2 sec.

  - expires_in:
    - Defines the durability of the task or the lifetime of theombe_id.
    - Maximum (default) value is 900 sec is returned if not a custom value is requested in Task Request.

  - task status:
    - Give the information about the task that was indicated by themobe_id
    - Possible values
      - “Processing” - delivers that the task is successfully accepted and is added to the task queue for annotation.
      - “Annotated” - delivers that the task is successfully annotated.

- Downloaded - delivers that the task is already downloaded.
- Expired - delivers that the task request expires.

```
Content-Type: application/json
Cache-Control: no-store
{
“themobe_id”:  "702552be-f0ef-4ca0–9f66-c766e9525e63",
“interval”:2,
“expires_in”:3600,
“task_status”: "Processing"
}
```

- **Annotation:**
  This is the internal process of the server. Scheduling tasks, process of annotating, storing annotated video and updating relevant databases is handled here. It is noteworthy to mention that the process of annotation includes our model or the approach of our project. Convert video into frames, preprocess the frames for behavior and emotion feature extraction, feature extraction using DTL-ResNet, use MAL-GSOM(especially specialized DAL-GSOM and then perform classification for relevant threats, emotions and behaviors and annotate as requested by the user/application.

  - Annotation request:
    The request that is made be application to poll for the update of the task.

```
"name": "AnnotationRequest",
"request": {
        "method": "GET",
        "header": [],
        "url": {
                "raw":
"http://127.0.0.1:5000/poll?themobe_id=11ef8766-f7e4-11ea-
9a22-a6d1fe3f2478",
                "protocol": "http",
                "host": [
                        "127",
                        "0",
                        "0",
                        "1"
                ],
                "port": "5000",
                "path": [
                        "poll"
                ],
```

```
                "query": [
                        {
                                "key": "themobe_id",
                                "value": "11ef8766-f7e4-11ea-
9a22-a6d1fe3f2478"
                        }
                ]
        }
}
```

- AnnotationResponse:

  It is a response for AnnotationRequest. Will respond with ***download_req_id*** if the annotation is completed and available to download and return an error message if not the task is done.

  - Download_req_id: Another unique identifier that is returned in the Annotation response confirming the task (or video) sent in for is completed.

```
Content-Type: application/json
Cache-Control: no-store
{
"download_req_id": download_req_id,
"task_status": "Annotated"
}
```

- Download Request

  Upon receiving successful annotation response (with ***download_req_id)*** can send download request requesting to download the video.

```
"name": "Download Request",
"request": {
        "method": "GET",
        "header": [],
        "url": {
                "raw":
"http://127.0.0.1:5000/download?download_req_id=6b694c66-40d9-
469a-9e06-d4bc20b24469",
                "protocol": "http",
                "host": [
                        "127",
                        "0",
                        "0",
                        "1"
                ],
                "port": "5000",
```

```
                "path": [
                        "poll"
                ],
                "query": [
                        {
                                "key": "download_req_id",
                                "value": "6b694c66-40d9-469a-9e06-
d4bc20b24469"
                        }
                ]
        }
}
```

- DownloadResponse
  This will directly download the annotated video

1.1.4 Error Codes

Error Codes	Description
200 - OK	Everything worked as expected.
400 - Bad Request	The request was unacceptable, often due to missing a required parameter.
401 Unauthorized	lacks valid authentication credentials for the target resource.
404 - Not Found	The requested resource doesn't exist
500- Internal Server Errors	Something went wrong on Servers end

1.1.5 Error Types

1. TaskRequest Error Response

   HTTP 400 Bad Request

   - invalid_request
     The request is missing a required parameter, includes an invalid parameter value, includes a parameter more than once, contains more than one of the hints, or is otherwise malformed

2. Annotation Request Error Response
   HTTP 400 Bad Request

   - task_not_completed
     The task request is still being processed as annotation is still progressing
   - slow_down

task request is still pending, and polling should continue, but the interval MUST be increased by at least 5 seconds for this and all subsequent requests.

- expired_id
  The themobe_id has expired. The Client will need to make a new Task Request.
- invalid_id
  Invalid themobe_id or ID which was not issued by the HABTP server.

3. DownloadRequest
   - excessive_downloads:
     Maximum number of downloads reached per download_req_id
   - expired_id
     The themobe_id has expired. The Client will need to make a new Task Request.
   - invalid_id
     Invalid download_req_id or ID which was not issued by the HABTP server.