Infosys

Responsible AI Office

Infosys Responsible AI Toolkit

Upload Doc

API usage Instructions

Contents

About Upload Doc

The proliferation of large media files, particularly videos, offers numerous opportunities for communication and content sharing, but it also introduces significant challenges regarding privacy and safety. It is reshaping our digital landscape and driving the need for advanced content moderation tools. It is necessary for a system handling such media to automatically process and secure user data. Essentially, it means that the system should be able to anonymize personal information and mask inappropriate content in a way that protects individuals. Hence, this capability is crucial for building user trust, ensuring regulatory compliance, and maintaining a safe digital ecosystem.

Once this API swagger page is populated, click on 'try it out' to use the required endpoints. Details of endpoints associated with the Document Upload and Processing service are outlined below.

Dependencies

The Upload Doc APIs depend on the Responsible AI File Storage, Responsible AI Privacy, and Responsible AI Safety repositories. Please follow the setup instructions in the README files of all dependent repositories to configure them. Ensure that all dependent services are up and running before interacting with the Upload Doc APIs.

Privacy Repository (responsible-ai-privacy)

The Privacy tenet of Infosys Responsible AI toolkit facilitates redacting the PII and any sensitive data in an organization by detecting the privacy entities and giving options to analyze, anonymize and encrypt using models like Tesseract, Easy OCR and Computer Vision. We analyze privacy in images, videos, and text by reading through the above document types, processing & applying the redacting techniques.

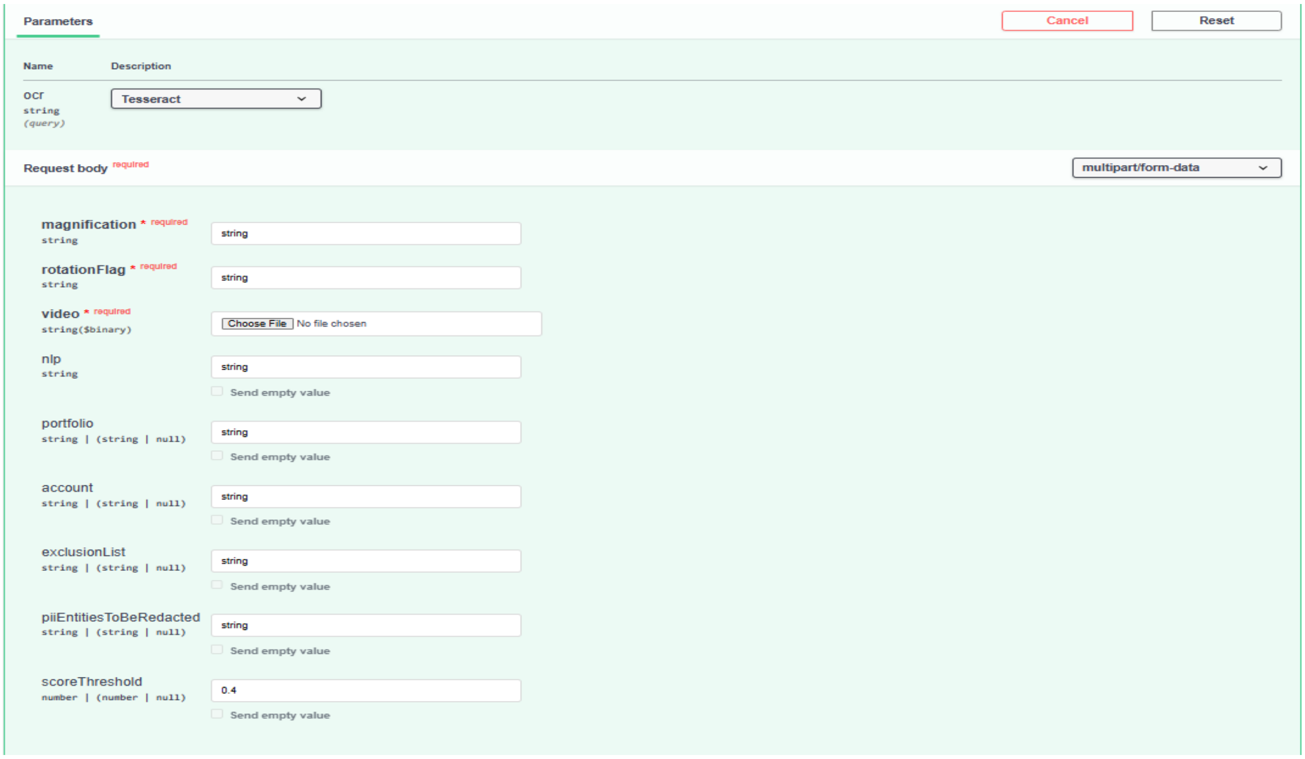
3.1: /rai/v1/video/anonymize

Using this API, detect and anonymize PII entities in video frames by drawing the bounding boxes. Provided below are the details of payloads.

- **OCR:** OCR engine for text extraction. Options: 'Tesseract', 'EasyOcr', 'ComputerVision'. Default: 'Tesseract'. Query parameter with Enum validation.
- **Payload:**
    - **Magnification:** String value to enable/disable image magnification for better OCR accuracy. Options: 'True', 'False'. Mandatory Form field.
    - **rotationFlag**: String value to enable/disable image rotation for better text detection. Options: 'True', 'False'. Mandatory Form field.
    - **Video:** Video file to be anonymized. Mandatory UploadFile parameter.
    - **NLP:** NLP model to use for PII detection. Options: "basic", "good", "roberta", "ranha". Optional Form field with default None.

- **Portfolio :** Portfolio name for account validation and tracking. Optional Form field with default None.
- **Account :** Account name for validation and billing purposes. Optional Form field with default None.
- **exclusionList :** Comma-separated list of PII entity types to exclude from anonymization (e.g., "PERSON,EMAIL"). Optional Form field with default None.
- **piiEntitiesToBeRedacted :** Comma-separated list of specific PII entity types to redact (e.g., "PHONE_NUMBER,SSN"). Optional Form field with default None.
- **scoreThreshold :** Confidence threshold for PII detection (float between 0.0-1.0). Higher values mean stricter detection. Optional Form field with default 0.4.
- **Account :** Account name for validation and billing purposes. Optional Form field with default None.

Once all fields are filled in, click "**execute**" to proceed.



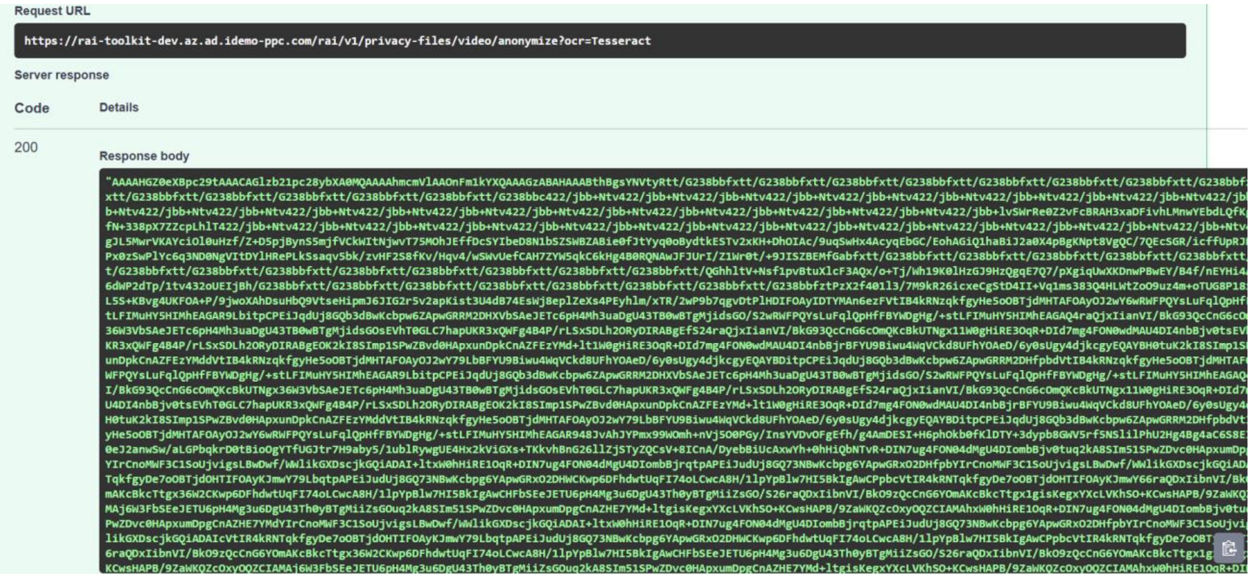
Name	Description
ocr string *(query)*	Tesseract

Request body required — multipart/form-data

magnification * required string	string
rotationFlag * required string	string
video * required string($binary)	Choose File No file chosen
nlp string	string □ Send empty value
portfolio string | (string | null)	string □ Send empty value
account string | (string | null)	string □ Send empty value
exclusionList string | (string | null)	string □ Send empty value
piiEntitiesToBeRedacted string | (string | null)	string □ Send empty value
scoreThreshold number | (number | null)	0.4 □ Send empty value

Response :

The screenshot below demonstrates the system's PII redaction process. It leverages the Presidio library to detect sensitive data (such as names, emails, and phone numbers), after which the Anonymization Engine automatically applies a blur/mask to the identified regions. This ensures the final output is fully compliant with privacy policies.



Request URL

```
https://rai-toolkit-dev.az.ad.idemo-ppc.com/rai/v1/privacy-files/video/anonymize?ocr=Tesseract
```

Server response

Code	Details
200	Response body

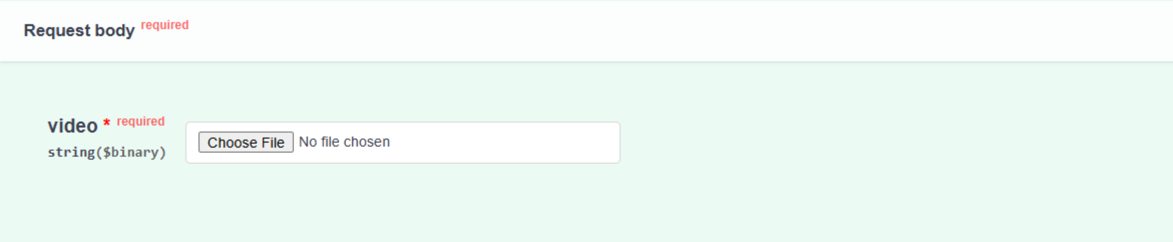
```
"AAAAHGZ0eXBpc29tAAACAGlzb21pc28ybXA0MQAAAAhmcmVlAAOnFm1kYXQAAAGzABAHAAABthBgsYNVtyRtt/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbf
xtt/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbc422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jb
b+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+lvSWrRe0Z2vFcBRAH3xaDFivhLMnwYEbdLQfK
fN+338pX7ZZcpLhlT422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv422/jbb+Ntv
gJL5MwrVKAYciOl0uHzf/Z+D5pjBynS5mjfVCkWItNjwvT75MOhJEffDcSYIbeD8N1bSZSWBZABie0fJtYyq0oBydtkESTv2xKH+DhOIAc/9uqSwHx4AcyqEbGC/EohAGiQ1haBiJ2a0X4pBgKNpt8VgQC/7QEcSGR/icffUpRJ
Px0zSwPlYc6q3ND0NgVItDYlHRePLkSsaqv5bk/zvHF2S8fKv/Hqv4/wSWvUefCAH7ZYW5qkC6kHg4B0RQNAwJFJUrI/Z1Wr0t/+9JISZBEMfGabfxtt/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbfxtt
t/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbfxtt/QGhhltV+Nsf1pvBtuXlcF3AQx/o+Tj/Wh19K0lHzGJ9HzQgqE7Q7/pXgiqUwXKDnwPBwEY/B4f/nEYHi4
6dWP2dTp/1tv432oUEIjBh/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbfxtt/G238bbfztPzX2f401l3/7M9kR26icxeCgStD4II+Vq1ms383Q4HLWtZoO9uz4m+oTUG8P18
L5S+KBvg4UKFOA+P/9jwoXAhDsuHbQ9VtseHipmJ6JIG2r5v2apKist3U4dB74EsWj8eplZeXs4PEyhlm/xTR/2wP9b7qgvDtPlHDIFOAyIDTYMAn6ezFVtIB4kRNzqkfgyHe5oOBTjdMHTAFOAyOJ2wY6wRWFPQYsLuFqlQpHf
tLFIMuHY5HIMhEAGAR9LbitpCPEiJqdUj8GQb3dBwKcbpw6ZApwGRRM2DHXVbSAeJETc6pH4Mh3uaDgU43TB0wBTgMjidsGO/S2wRWFPQYsLuFqlQpHfFBYWDgHg/+stLFIMuHY5HIMhEAGAQ4raQjxIianVI/BkG93QcCnG6cO
36W3VbSAeJETc6pH4Mh3uaDgU43TB0wBTgMjidsGOsEVhT0GLC7hapUKR3xQWFg4B4P/rLSxSDLh2ORyDIRABgEfS24raQjxIianVI/BkG93QcCnG6cOmQKcBkUTNgx11W0gHiRE3OqR+DId7mg4FON0wdMAU4DI4nbBjv0tsEV
KR3xQWFg4B4P/rLSxSDLh2ORyDIRABgEOK2kI8SImp1SPwZBvd0HApxunDpkCnAZFEzYMd+lt1W0gHiRE3OqR+DId7mg4FON0wdMAU4DI4nbBjrBFYU9Biwu4WqVCkd8UFhYOAeD/6y0sUgy4djkcgyEQAYBH0tuK2kI8SImp1S
unDpkCnAZFEzYMddVtIB4kRNzqkfgyHe5oOBTjdMHTAFOAyOJ2wY79LbBFYU9Biwu4WqVCkd8UFhYOAeD/6y0sUgy4djkcgyEQAYBDitpCPEiJqdUj8GQb3dBwKcbpw6ZApwGRRM2DHfpbdVtIB4kRNzqkfgyHe5oOBTjdMHTAF
WFPQYsLuFqlQpHfFBYWDgHg/+stLFIMuHY5HIMhEAGAR9LbitpCPEiJqdUj8GQb3dBwKcbpw6ZApwGRRM2DHXVbSAeJETc6pH4Mh3uaDgU43TB0wBTgMjidsGO/S2wRWFPQYsLuFqlQpHfFBYWDgHg/+stLFIMuHY5HIMhEAGAQ
I/BkG93QcCnG6cOmQKcBkUTNgx36W3VbSAeJETc6pH4Mh3uaDgU43TB0wBTgMjidsGOsEVhT0GLC7hapUKR3xQWFg4B4P/rLSxSDLh2ORyDIRABgEfS24raQjxIianVI/BkG93QcCnG6cOmQKcBkUTNgx11W0gHiRE3OqR+DId7
U4DI4nbBjv0tsEVhT0GLC7hapUKR3xQWFg4B4P/rLSxSDLh2ORyDIRABgEOK2kI8SImp1SPwZBvd0HApxunDpkCnAZFEzYMd+lt1W0gHiRE3OqR+DId7mg4FON0wdMAU4DI4nbBjrBFYU9Biwu4WqVCkd8UFhYOAeD/6y0sUgy4
H0tuK2kI8SImp1SPwZBvd0HApxunDpkCnAZFEzYMddVtIB4kRNzqkfgyHe5oOBTjdMHTAFOAyOJ2wY79LbBFYU9Biwu4WqVCkd8UFhYOAeD/6y0sUgy4djkcgyEQAYBDitpCPEiJqdUj8GQb3dBwKcbpw6ZApwGRRM2DHfpbdVt
yHe5oOBTjdMHTAFOAyOJ2wY6wRWFPQYsLuFqlQpHfFBYWDgHg/+stLFIMuHY5HIMhEAGAR948JvAhJYPmx99WOmh+nVj5O0PGy/InsYVDvOFgEfh/g4AmDESI+H6phOkb0fKlDTY+3dypb8GWV5rf5NSlilPhU2Hg4Bg4aC6S8E
0eJ2anwSw/aLGPbqkrD0tBioOgYTfUGJtr7H9aby5/1ublRywgUE4Hx2kViGXs+TKkvhBnG26llZjSTyZQCsV+8ICnA/DyebBiUcAxwYh+0hHiQbNTvR+DIN7ug4FON04dMgU4DIombBjv0tuq2kA8SIm51SPwZDvc0HApxumDp
YIrCnoMWF3C1SoUjvigsLBwDwf/WWlikGXDscjkGQiADAI+ltxW0hHiRE1OqR+DIN7ug4FON04dMgU4DIombBjrqtpAPEiJudUj8GQ73NBwKcbpg6YApwGRxO2DHfpbYIrCnoMWF3C1SoUjvigsLBwDwf/WWlikGXDscjkGQiAD
TqkfgyDe7oOBTjdOHTIFOAyKJmwY79LbqtpAPEiJudUj8GQ73NBwKcbpg6YApwGRxO2DHWCKwp6DFhdwtUqFI74oLCwcA8H/1lpYpBlw7HI5BkIgAwCPpbcVtIR4kRNTqkfgyDe7oOBTjdOHTIFOAyKJmwY66raQDxIibnVI/Bk
mAKcBkcTtgx36W2CKwp6DFhdwtUqFI74oLCwcA8H/1lpYpBlw7HI5BkIgAwCHFbSEeJETU6pH4Mg3u6DgU43Th0yBTgMiiZsGO/S26raQDxIibnVI/BkO9zQcCnG6YOmAKcBkcTtgx1gisKegxYXcLVKhSO+KCwsHAPB/9ZaWKQ
MAj6W3FbSEeJETU6pH4Mg3u6DgU43Th0yBTgMiiZsGOuq2kA8SIm51SPwZDvc0HApxumDpgCnAZHE7YMd+ltgisKegxYXcLVKhSO+KCwsHAPB/9ZaWKQZcOxyOQZCIAMAhxW0hHiRE1OqR+DIN7ug4FON04dMgU4DIombBjv0tu
PwZDvc0HApxumDpgCnAZHE7YMdYIrCnoMWF3C1SoUjvigsLBwDwf/WWlikGXDscjkGQiADAI+ltxW0hHiRE1OqR+DIN7ug4FON04dMgU4DIombBjrqtpAPEiJudUj8GQ73NBwKcbpg6YApwGRxO2DHfpbYIrCnoMWF3C1SoUjvi
likGXDscjkGQiADAIcVtIR4kRNTqkfgyDe7oOBTjdOHTIFOAyKJmwY79LbqtpAPEiJudUj8GQ73NBwKcbpg6YApwGRxO2DHWCKwp6DFhdwtUqFI74oLCwcA8H/1lpYpBlw7HI5BkIgAwCPpbcVtIR4kRNTqkfgyDe7oOBT
6raQDxIibnVI/BkO9zQcCnG6YOmAKcBkcTtgx36W2CKwp6DFhdwtUqFI74oLCwcA8H/1lpYpBlw7HI5BkIgAwCHFbSEeJETU6pH4Mg3u6DgU43Th0yBTgMiiZsGO/S26raQDxIibnVI/BkO9zQcCnG6YOmAKcBkcTtgx1g
KCwsHAPB/9ZaWKQZcOxyOQZCIAMAj6W3FbSEeJETU6pH4Mg3u6DgU43Th0yBTgMiiZsGOuq2kA8SIm51SPwZDvc0HApxumDpgCnAZHE7YMd+ltgisKegxYXcLVKhSO+KCwsHAPB/9ZaWKQZcOxyOQZCIAMAhxW0hHiRE1OqR+DI
```

Safety Repository (responsible-ai-safety)

AI models can sometimes generate harmful content, such as profanity, toxic language, explicit images, or sexually suggestive text. To ensure safe and responsible AI, it's crucial to implement measures that filter and prevent the generation of such content. This involves using advanced techniques to detect and mitigate harmful outputs, protecting users from exposure to inappropriate material, and maintaining a positive and inclusive online environment.

4.1 : /api/v1/safety/profanity/videosafety :

Processes uploaded video files to detect NSFW content and returns comprehensive safety analysis results.



Request body required

video * required
string($binary)

Choose File No file chosen

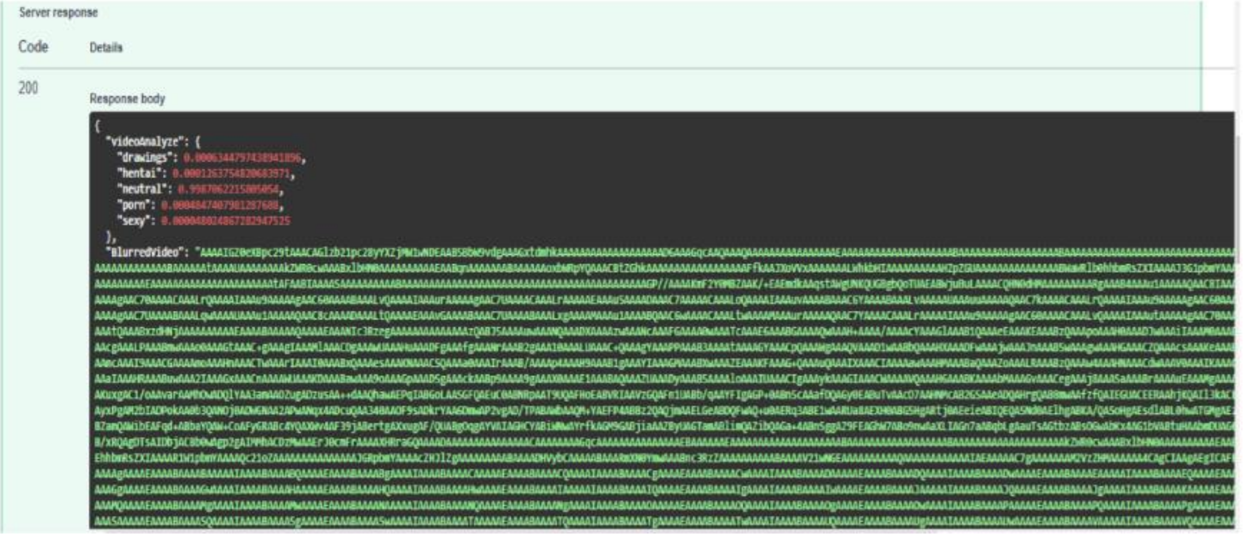
Payload :

- **video :** Video file upload (required) - The video file to be analyzed for NSFW content.

Returns analysis results indicating NSFW content detection in the uploaded video.

Response:

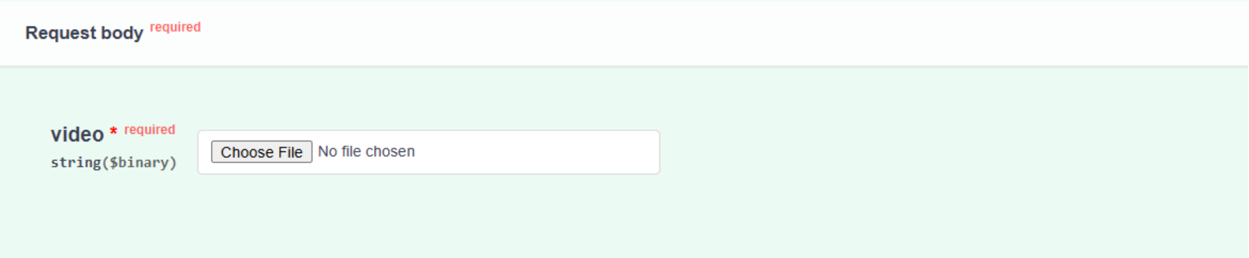
Analyzes a video for content types (e.g., hentai, porn, sexy, neutral, drawings) and returns a classification score along with the base64-encoded video data.



```
{
  "videoAnalyze": {
    "drawings": 0.0006344797438941896,
    "hentai": 0.0001263754820683971,
    "neutral": 0.9987062215005054,
    "porn": 0.0004847407981287688,
    "sexy": 0.000048024867282947525
  },
  "BlurredVideo": "AAAAIGZ0eXBpc29tAAACAGlzb21pc28yYXZjMW1wNDEAABS8bW9vdgAAAGxtdmhkAAAAAAAAAAAAAAAAAAAD6AAAGqCAAQAAAQAAAAAAAAAAAAAAAAEAAAAAAAAAAAAAAAAAAAABAAAAAAAAAAAAAAAAAAB [illegible]
```

4.2 : /api/v1/safety/profanity/nudvideosafety :

Processes uploaded video files to detect nudity content and returns safety analysis results.



Request body required

video * required
string($binary)

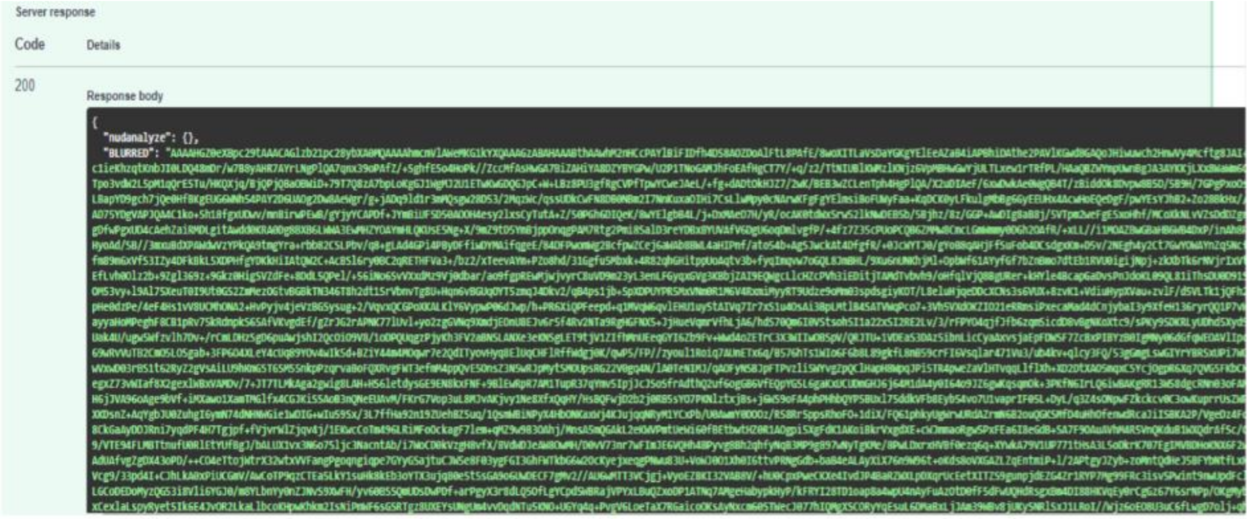
Payload :

- **video :** Video file upload (required) - The video file to be analyzed for NSFW content.

Returns analysis results indicating nudity detection in the uploaded video.

Response:

Returns the video with the part to be blurred in base-64 encoded format.



Server response

Code Details

200

Response body

```
{
  "nudanalyze": {},
  "BLURRED": "AAAAHGZ0eXBpc29tAAACAGlzb21pc28ybXA0MQAAAAhmcmVlAAeMKG1kYXQAAAGzABAHAAABthAAwM2mHCcPAYlBiFIDfh4DS8AOZDoAlFtL8PAfE/8woXITLaVsDaYGKgYElEeAZaB4iAP6hiDAthe2PAV1KGwdBGAQoJHiwawch2HmWy4McftgBJAI
[illegible]
```

File Storage Repository (responsible-ai-file-storage)

The File Storage repository provides a unified interface for managing file operations across multiple cloud storage platforms. It abstracts the complexity of different storage providers and offers consistent APIs for uploading, retrieving, and managing files. This repository supports Azure Blob Storage, Google Cloud Platform (GCP) Storage, Amazon Web Services (AWS) S3 as storage backends.

5.1 : api/v1/azureBlob/addFile :

Uploads a processed file to a specified Azure Blob Storage container and returns the blob reference details.

5.2 : api/v1/azureBlob/getBlob:

Downloads a file from Azure Blob Storage using the blob name and container name parameters.

5.3 : api/v1/gcs/addFile :

Uploads a processed file to a specified GCP Storage bucket and returns the object reference details.

5.4 : api/v1/gcs/getObject :

Downloads a file from GCP Storage using the object name and bucket name.

5.5 : api/v1/s3/uploadFile:

Uploads a processed file to a specified AWS S3 bucket and returns the object key reference details.

5.6 : api/v1/s3/getObject:

Downloads a file from AWS S3 Storage using the object key and bucket name parameters.

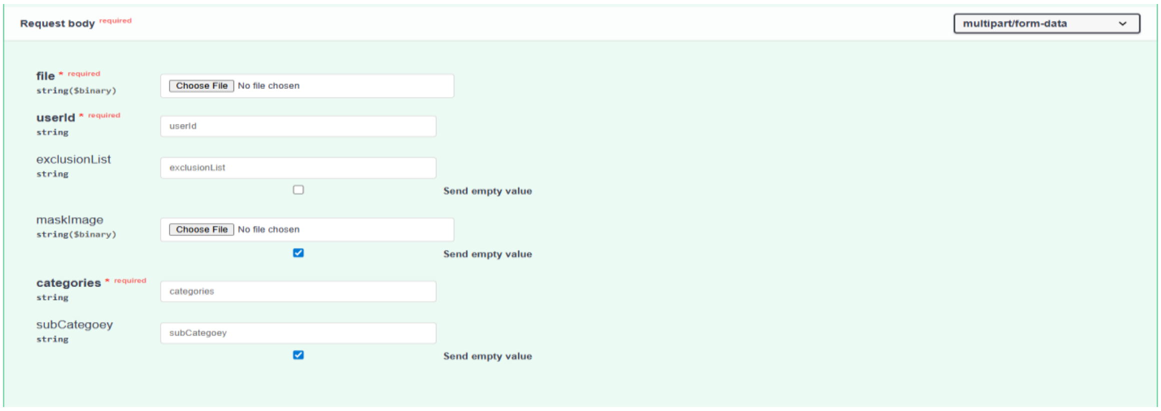
Features & API End Points:

• Upload Doc APIs

We have four APIs available in the Upload Doc repository, each serving a different purpose as detailed below:

6.1: /docProcess/uploadFile - Uploads a file (currently supports video, Excel, and PowerPoint) for processing and storage. It can perform various operations like PII anonymization, face anonymization, and content masking based on the provided sub-categories.

Payload



Request body required

multipart/form-data

file * required
string($binary)
Choose File No file chosen

userId * required
string
userId

exclusionList
string
exclusionList
□ Send empty value

maskImage
string($binary)
Choose File No file chosen
☑ Send empty value

categories * required
string
categories

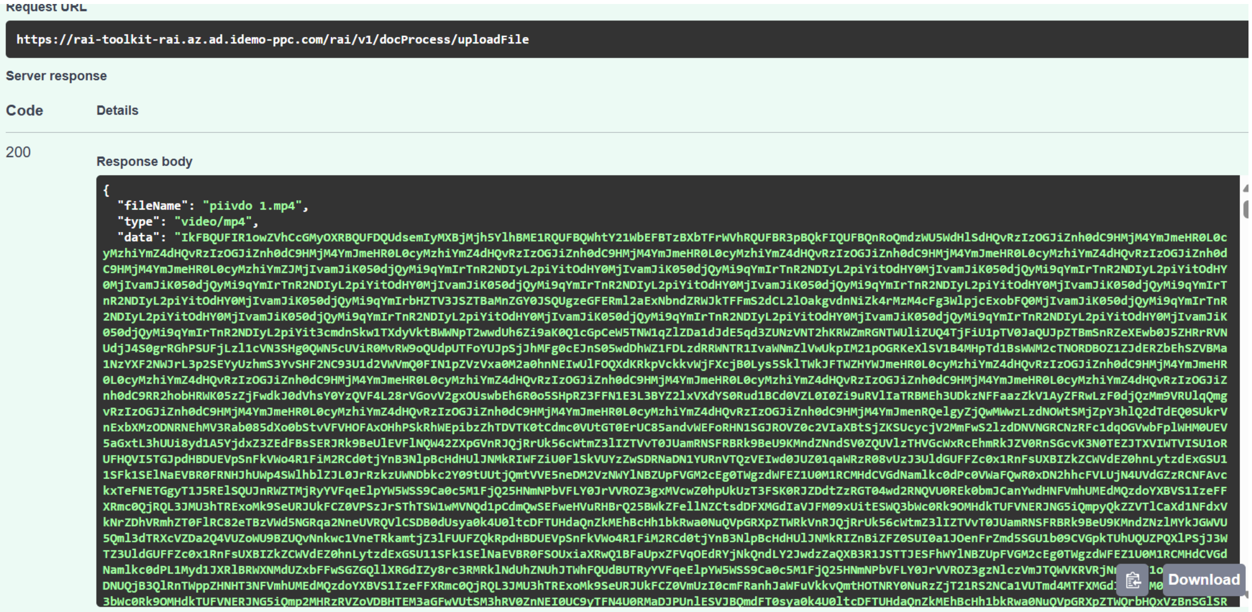
subCategoey
string
subCategoey
☑ Send empty value

- **File** : The file to be uploaded (e.g., .mp4). This is a mandatory field.
- **userId** : A unique identifier for the user uploading the file. This is a mandatory field.
- **exclusionList** : A list of items to exclude during processing (e.g., for PIIAnonymize).
- **maskImage** : An image file required only when CustomMask is specified in subCategory for videos.
- **Categories** : The category for the document (e.g., "videos", "documents"). This is a mandatory field.
- **subcategory** : A comma-separated string specifying the processing steps to be applied.
  For videos :-
  (video/mp4): PIIAnonymize, FaceAnonymize, SafetyMasking, NudityMasking, CustomMask.

Returns A JSON object containing the fileName, type, and the base64-encoded content of the processed file.

Response :

Analyzes a video based on the sub category given and returns e base64-encoded video data.



Request URL

```
https://rai-toolkit-rai.az.ad.idemo-ppc.com/rai/v1/docProcess/uploadFile
```

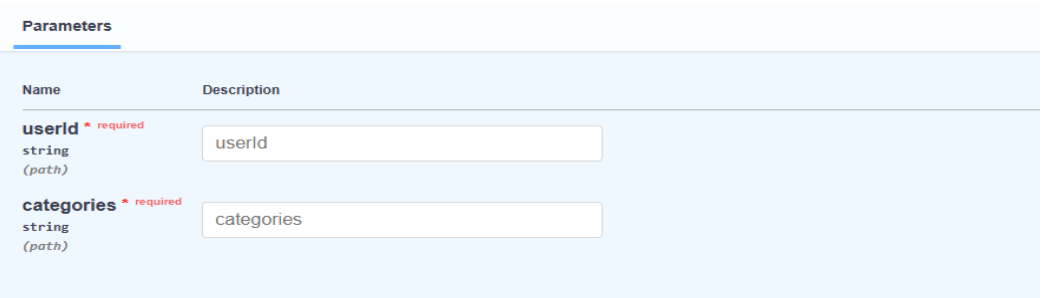
Server response

Code	Details
200	Response body

```
{
  "fileName": "piivdo 1.mp4",
  "type": "video/mp4",
  "data": "IkFBQUFIR1owZVhCcGMyOXRBQUFDQUdsemIyMXBjMjh5YlhBME1RQUFBQWhtY21WbEFBTzBXbTFrWVhRQUFBR3pBQkFIQUFBQnRoQmdzWU5WdHlSdHQvRzIzOGJiZnh0dC9HMjM4YmJmeHR0L0c
yMzhiYmZ4dHQvRzIzOGJiZnh0dC9HMjM4YmJmeHR0L0cyMzhiYmZ4dHQvRzIzOGJiZnh0dC9HMjM4YmJmeHR0L0cyMzhiYmZ4dHQvRzIzOGJiZnh0dC9HMjM4YmJmeHR0L0cyMzhiYmZ4dHQvRzIzOGJiZnh0d
C9HMjM4YmJmeHR0L0cyMzhiYmZJMjIvamJiK050djQyMi9qYmIrTnR2NDIyL2piYitOdHY0MjIvamJiK050djQyMi9qYmIrTnR2NDIyL2piYitOdHY0MjIvamJiK050djQyMi9qYmIrTnR2NDIyL2piYitOdHY
0MjIvamJiK050djQyMi9qYmIrTnR2NDIyL2piYitOdHY0MjIvamJiK050djQyMi9qYmIrTnR2NDIyL2piYitOdHY0MjIvamJiK050djQyMi9qYmIrTnR2NDIyL2piYitOdHY0MjIvamJiK050djQyMi9qYmIrT
nR2NDIyL2piYitOdHY0MjIvamJiK050djQyMi9qYmIrbHZTV3JSZTBaMnZGY0JSQUgzeGFERml2aExNbndZRWJkTFFmS2dCL2lOakgvdnNiZk4rMzM4cFg3WlpjcExobFQ0MjIvamJiK050djQyMi9qYmIrTnR
2NDIyL2piYitOdHY0MjIvamJiK050djQyMi9qYmIrTnR2NDIyL2piYitOdHY0MjIvamJiK050djQyMi9qYmIrTnR2NDIyL2piYitOdHY0MjIvamJiK050djQyMi9qYmIrTnR2NDIyL2piYitOdHY0MjIvamJiK
050djQyMi9qYmIrTnR2NDIyL2piYit3cmdnSkw1TXdyVktBWWNpT2wwdUh6Zi9aK0Q1cGpCeW5TNW1qZlZDa1dJdE5qd3ZUNzVNT2hKRWZmRGNTWUliZUQ4TjFiU1pTV0JaQUJpZTBmSnRZeXEwb0J5ZHRrRVN
UdjJ4S0grRGhPSUFjLzl1cVN3SHg0QWN5cUViR0MvRW9oQUdpUTFoYUJpSjJhMFg0cEJnS05wdDhWZ1FDLzdRRWNTR1IvaWNmZlVwUkpIM21pOGRKeXlSV1B4MHpTd1BsWWM2cTNORDBOZ1ZJdERZbEhSZVBMa
1NzYXF2NWJrL3p2SEYyUzhmS3YvSHF2NC93U1d2VWVmQ0FIN1pZVzVxa0M2a0hnNEIwUlFOQXdKRkpVckkvWjFXcjB0Lys5SklTWkJFTWZHYWJmeHR0L0cyMzhiYmZ4dHQvRzIzOGJiZnh0dC9HMjM4YmJmeHR
0L0cyMzhiYmZ4dHQvRzIzOGJiZnh0dC9HMjM4YmJmeHR0L0cyMzhiYmZ4dHQvRzIzOGJiZnh0dC9HMjM4YmJmeHR0L0cyMzhiYmZ4dHQvRzIzOGJiZnh0dC9HMjM4YmJmeHR0L0cyMzhiYmZ4dHQvRzIzOGJiZ
nh0dC9RR2hobHRWK05zZjFwdkJ0dVhsY0YzQVF4L28rVGovV2gxOUswbEh6R0o5SHpRZ3FFN1E3L3BYZ2lxVXdYS0Rud1BCd0VZL0I0Zi9uRVlIaTRBMEh3UDkzNFFaazZkV1AyZFRwLzF0djQzMm9VRUlqQmg
vRzIzOGJiZnh0dC9HMjM4YmJmeHR0L0cyMzhiYmZ4dHQvRzIzOGJiZnh0dC9HMjM4YmJmeHR0L0cyMzhiYmZ4dHQvRzIzOGJiZnh0dC9HMjM4YmJmenRQelgyZjQwMWwzLzdNOWtSMjZpY3hlQ2dTdEQ0SUkrV
nExbXMzODNRNEhMV3Rab085dXo0bStvVFVHOFAxOHhPSkRhWEpibzZhTDVTK0tCdmc0VUtGT0ErUC85andvWEFoRHN1SGJROVZ0c2VIaXBtSjZKSUcycjV2MmFwS2lzdDNVNGRCNzRFc1dqOGVwbFplWHM0UEV
5aGxtL3hUUi8yd1A5YjdxZ3ZEdFBsSERJRk9BeUlEVFlNQW42ZXpGVnRJQjRrUk56cWtmZ3lIZTVvT0JUamRNSFRBRk9BeU9KMndZNndSV0ZQUVlzTHVGcWxRcEhmRkJZV0RnSGcvK3N0TEZJTXVIWTVISU1oR
UFHQVI5TGJpdHBDUEVpSnFkVWo4R1FiM2RCd0tjYnB3NlpBcHdHUlJNMkRIWFZiU0FlSkVUYzZwSDRNaDN1YURnVTQzVEIwd0JUZ01qaWRzR08vUzJ3UldGUFFZc0x1RnFsUXBIZkZCWVdEZ0hnLytzdExGSU1
1SFk1SElNaEVBR0FRNHJhUWp4SWlhblZJL0JrRzkzUWNDbkc2Y09tUUtjQmtVVE5neDM2VzNWYlNBZUpFVGM2cEg0TWgzdWFEZ1U0M1RCMHdCVGdNamlkc0dPc0VWaFQwR0xDN2hhcFVLUjN4UVdGZzRCNFAvc
kxTeFNETGgyT1J5RElSQUJnRWZTMjRyYVFqeElpYW5WSS9Ca0c5M1FjQ25HNmNPbVFLY0JrVVROZ3gxMVcwZ0hpUkUzT3FSK0RJZDdtZzRGT04wd2RNQVU0REk0bmJCanYwdHNFVmhUMEdMQzdoYXBVS1IzeFF
XRmc0QjRQL3JMU3hTRExoMk9SeURJUkFCZ0VPSzJrSThTSW1wMVNQd1pCdmQwSEFweHVuRHBrQ25BWkZFellNZCtsdDFXMGdIaVJFM09xUitESWQ3bWc0Rk9OMHdkTUFVNERJNG5iQmpyQkZZVTlCaXd1NFdxV
kNrZDhVRmhZT0FlRC82eTBzVWd5NGRqa2NneUVRQVlCSDB0dUsya0k4U0ltcDFTUHdaQnZkMEhBcHh1bkRwa0NuQVpGRXpZTWRkVnRJQjRrUk56cWtmZ3lIZTVvT0JUamRNSFRBRk9BeU9KMndZNzlMYkJGWVU
5Qml3dTRXcVZDa2Q4VUZoWU9BZUQvNnkwc1VneTRkamtjZ3lFUUFZQkRpdHBDUEVpSnFkVWo4R1FiM2RCd0tjYnB3NlpBcHdHUlJNMkRIZnBiZFZ0SUI0a1JOenFrZmd5SGU1b09CVGpkTUhUQUZPQXlPSjJ3
TZ3UldGUFFZc0x1RnFsUXBIZkZCWVdEZ0hnLytzdExGSU11SFk1SElNaEVBR0FSOUxiaXRwQ1BFaUpxZFVqOEdRYjNkQndLY2JwdzZaQXB3R1JSTTJESFhWYlNBZUpFVGM2cEg0TWgzdWFEZ1U0M1RCMHdCVGd
Namlkc0dPL1Myd1JXRlBRWXNMdUZxbFFwSGZGQllXRGdIZy8rc3RMRkl NdUhZNUhJTWhFQUdBUTRyYVFqeElpYW5WSS9Ca0c5M1FjQ25HNmNPbVFLY0JrVVROZ3gzNlczVmJTQWVKRVRjN[illegible]1o
DNUQjB3QlRnTWppZHNHT3NFVmhUMEdMQzdoYXBVS1IzeFFXRmc0QjRQL3JMU3hTRExoMk9SeURJUkFCZ0VmUzI0cmFRanhJaWFuVkkvQmtHOTNRY0NuRzZjT21RS2NCa1VUTmd4MTFXMGd[illegible]M0
3bWc0Rk9OMHdkTUFVNERJNG5iQmp2MHRzRVZoVDBHTEM3aGFwVUtSM3hRV0ZnNEI0UC9yTFN4U0RMaDJPUnlESVJBQmdFT0sya0k4U0ltcDFTUHdaQnZkMEhBcHh1bkRwa0NuQVpGRXpZTWQrbHQxVzBnSGlSR
```

6.2: /docProcess/getFiles/{userId}/{categories} - Retrieves a list of file metadata for a specific user and category.

Payload

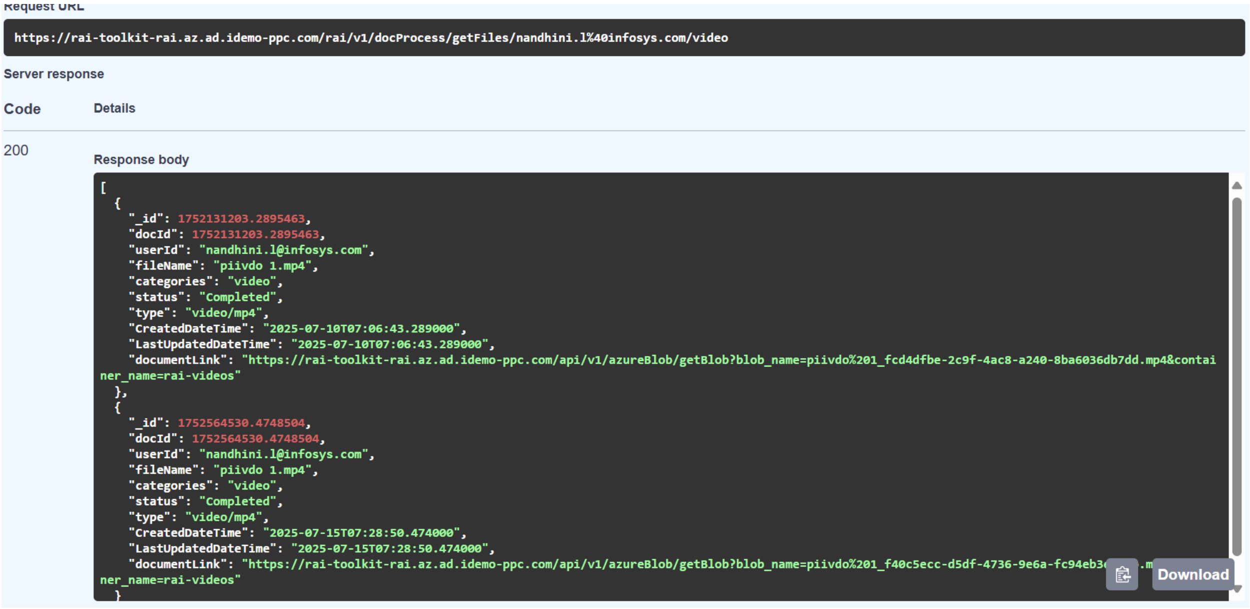


Name	Description
userId * required string *(path)*	userId
categories * required string *(path)*	categories

- **userID** : The ID of the user whose files you want to retrieve.
- **categories** : The category of files to retrieve (e.g., "video").

Response:

A JSON list of documents, where each document contains metadata about a file, such as fileName, documentLink, and status.



Request URL

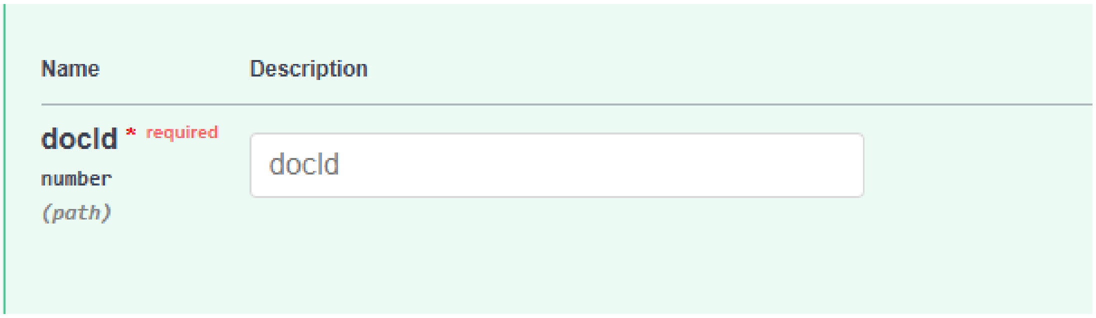
```
https://rai-toolkit-rai.az.ad.idemo-ppc.com/rai/v1/docProcess/getFiles/nandhini.l%40infosys.com/video
```

Server response

Code	Details
200	Response body

```
[
  {
    "_id": 1752131203.2895463,
    "docId": 1752131203.2895463,
    "userId": "nandhini.l@infosys.com",
    "fileName": "piivdo 1.mp4",
    "categories": "video",
    "status": "Completed",
    "type": "video/mp4",
    "CreatedDateTime": "2025-07-10T07:06:43.289000",
    "LastUpdatedDateTime": "2025-07-10T07:06:43.289000",
    "documentLink": "https://rai-toolkit-rai.az.ad.idemo-ppc.com/api/v1/azureBlob/getBlob?blob_name=piivdo%201_fcd4dfbe-2c9f-4ac8-a240-8ba6036db7dd.mp4&contai
ner_name=rai-videos"
  },
  {
    "_id": 1752564530.4748504,
    "docId": 1752564530.4748504,
    "userId": "nandhini.l@infosys.com",
    "fileName": "piivdo 1.mp4",
    "categories": "video",
    "status": "Completed",
    "type": "video/mp4",
    "CreatedDateTime": "2025-07-15T07:28:50.474000",
    "LastUpdatedDateTime": "2025-07-15T07:28:50.474000",
    "documentLink": "https://rai-toolkit-rai.az.ad.idemo-ppc.com/api/v1/azureBlob/getBlob?blob_name=piivdo%201_f40c5ecc-d5df-4736-9e6a-fc94eb3 .m
ner_name=rai-videos"
  }
```

6.3: /docProcess/getFileContent/{docId} - Retrieves the content of a previously processed file.

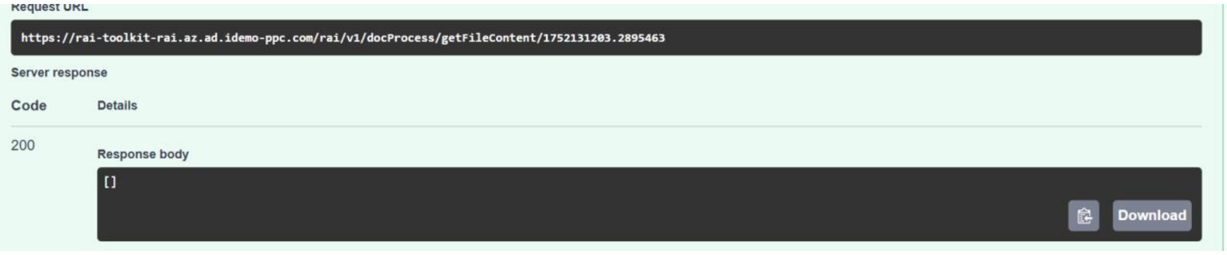


Name	Description
docId * required number *(path)*	docId

- **docId :** The unique ID of the document whose content is to be retrieved. This ID is obtained from the getFiles endpoint.

Reponse:

A JSON list containing the file data, where the file content is base64-encoded.



Request URL

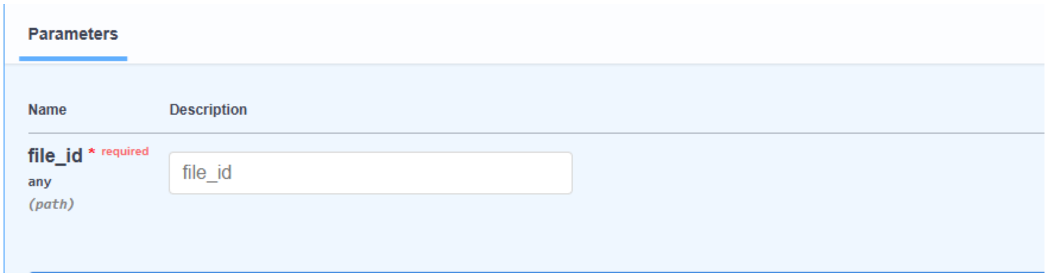
```
https://rai-toolkit-rai.az.ad.idemo-ppc.com/rai/v1/docProcess/getFileContent/1752131203.2895463
```

Server response

Code	Details
200	Response body `[]`

6.4: /docProcess/download/{file_id} - Downloads the raw file directly from the storage system (if storage_option=mongodb).

Payload



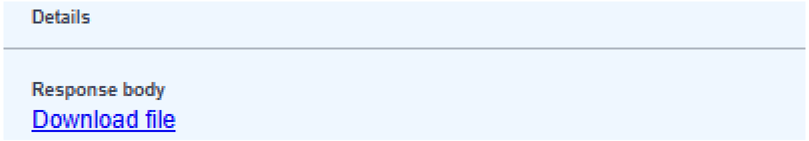
Parameters

Name	Description
file_id * required any *(path)*	file_id

- **file_id** : The unique identifier of the file in the storage system (MongoDB Object Id). This ID is part of the metadata returned by the getFiles endpoint.retrieve.

Respone:

Returns the file as a streaming response, which will trigger a download in the browser or can be consumed by a client.



Details

Response body

Download file

- ## Endpoints Usage Flow

Endpoint	Description
/docProcess/uploadFile	Uploads and processes files (videos, Excel, PowerPoint) with configurable anonymization, masking, and safety features before storing them.
/docProcess/getFiles/{userId}/{categories}	Retrieves a list of all processed files and their metadata for a specific user and category.
/docProcess/getFileContent/{docId}	Fetches the actual content of a previously processed file using its document ID.
/docProcess/download/{file_id}	Downloads the raw processed file directly from the storage system as a streaming response.