

Documentation



YouTube Downloader API — Usage Guide

1. POST /verification

Description

Verifies whether a given YouTube URL is valid and retrieves video metadata.

Request

```
{
  "url": "https://www.youtube.com/watch?v=dQw4w9WgXcQ"
}
```

Responses

✓ 200 OK

```
{
  "title": "Example Video Title",
  "channel": "Channel Name",
  "duration": "3:45",
  "views": "123456",
  "thumbnail": "https://i.ytimg.com/vi/dQw4w9WgXcQ/hqdefault.jpg"
}
```

✗ 400 Bad Request

- When `url` is missing:

```
{ "error": "Please provide a URL" }
```

- When the URL is invalid:

```
{ "error": "URL is not valid" }
```

! 500 Internal Server Error

- When metadata could not be fetched:

```
{ "error": "Failed to fetch video data: <error_message>" }
```

Example CURL

```
curl -X POST http://127.0.0.1:5000/verification \
-H "Content-Type: application/json" \
-d '{"url": "https://www.youtube.com/watch?v=dQw4w9WgXcQ"}'
```

2. POST /downloads

Description

Downloads the requested YouTube video or audio file and sends it back as an attachment.

Request

```
{
  "url": "https://www.youtube.com/watch?v=dQw4w9WgXcQ",
  "format": "mp3"
}
```

Responses

✓ 200 OK

- Returns the requested file for download.
- Content-Type:
 - `audio/mpeg` → for MP3
 - `video/mp4` → for MP4

Example response headers:

```
Content-Disposition: attachment; filename="video_title.mp3"
Content-Type: audio/mpeg
```

✖ 400 Bad Request

- When JSON body is missing:

```
{ "error": "No JSON data provided" }
```

- When `url` or `format` is missing:

```
{ "error": "Please provide both 'url' and 'format'" }
```

⚠ 500 Internal Server Error

- When download fails or file is missing:

```
{ "error": "Download failed or file missing" }
```

- When file unexpectedly doesn't exist:

```
{ "error": "File /path/to/file does not exist" }
```

Example CURL

```
curl -X POST http://127.0.0.1:5000/downloads \  
-H "Content-Type: application/json" \  
-d '{"url": "https://www.youtube.com/watch?v=dQw4w9WgXcQ", "format":  
"mp3"}' \  
-o downloaded_audio.mp3
```

Supported Formats

Format	Description	MIME Type
mp3	Audio only	audio/mpeg
mp4	Video + audio	video/mp4

Progress bars /progress-stream

Description

Server-Sent Events (SSE) provide **real-time progress updates** from the backend to the frontend over a **single, persistent HTTP connection**.

```
data: {"status": "idle/downloading/finished", 'percentage'"0,1,2...":
}
```

Summary

Endpoint	Method	Purpose	Successful Response
/verification	POST	Check if a YouTube URL is valid and get metadata	JSON with video info
/downloads	POST	Download YouTube video or audio	File (MP3 or MP4)
/progress-stream	GET	Sens stream of data about the Download	Text response(status and percentage)

