# Documentation for Classes:

This application manages video processing jobs through RESTful APIs. It allows clients to:

* Create a video processing job.
* Check the status of a job.
* Delete a job.

**Key Components:**

**VideoController**: Provides REST endpoints for managing video jobs.

**VideoService**: Handles the business logic for creating, monitoring, and deleting video jobs.

# VideoController:

**Class Description**

VideoController acts as the entry point for the REST API. It provides endpoints for creating, monitoring, and deleting video jobs.

## Endpoints

### Translate Video Job

**URL**: /translateVideo

**Method**: POST

**Request Body**: (JSON format)

{

"jobId": "string"

}

**Response**: json

{

"status": "string"

}

**Description**: Starts a new video processing job. Returns the status of the operation.

**Exceptions**: Throws BadRequestException if the job already exists.

### Get Job Status

**URL**: /status/{jobId}

**Method**: GET

**Path Variable**:

jobId: The unique identifier for the job.

**Response**: json

{

"status": "string"

}

**Description**: Retrieves the current status of the specified video job.

**Exceptions**: Throws RuntimeException if the job ID does not exist.

### Delete Video Job

**URL**: /delete/{jobId}

**Method**: POST

**Path Variable**:

jobId: The unique identifier for the job.

**Response**:

json

Copy code

{

"status": "string"

}

**Description**: Marks the specified job as deleted by setting its status to error.

## VideoService

**Class Description**

VideoService contains the business logic for managing video jobs. It supports job creation, status tracking, and deletion.

### Properties

**Map<String, String> videos**:

A static map storing job IDs and their corresponding statuses.

*Status values: pending, completed, error.*

**@Value("${video.translation.timeRequired}") Long timeRequired**:

The time required for processing a video job, configurable through application properties.

## Methods

**1. Translate Video Job:** java

public String createVideo(Job job) throws BadRequestException

**Input**:

Job job: A job object containing the job ID.

**Output**:

Returns created after the job is initiated.

**Description**: Creates a new job and starts background threads for job execution and monitoring.

**Exceptions**:

Throws BadRequestException if the job ID already exists.

**2. Perform Job (Private):** java

private void performJob(String jobId) throws InterruptedException

**Input**:

jobId: The unique identifier of the job.

**Description**: Simulates the job execution process. Updates the job status to completed after successful execution or error in case of interruptions.

**3. Get Job Status:** java

public String getStatus(String jobId) throws RuntimeException

**Input**:

jobId: The unique identifier of the job.

**Output**:

The current status of the job (pending, completed, or error).

**Exceptions**:

Throws RuntimeException if the job ID does not exist.

**4. Delete Job:** java

public String deleteVideo(String jobId)

**Input**:

jobId: The unique identifier of the job.

**Output**:

Returns deleted after marking the job as deleted.

**Description**: Updates the job status to error.

## Job

**Class Description**

Represents a video processing job.

**Properties**

**String jobId**: The unique identifier for the job.

## Example Usage

### Create a Job

**Request**: http

POST /createVideo

Content-Type: application/json

{

"jobId": "job123"

}

**Response**:

json

Copy code

{

"status": "created"

}

### Check Job Status

**Request**: http

GET /status/job123

**Response**:

json

Copy code

{

"status": "pending"

}

### Delete a Job

**Request**: http

POST /delete/job123

**Response**:

json

Copy code

{

"status": "deleted"

}

### Configuration

**Properties**

**video.translation.timeRequired**:

The time required to process a video (in seconds).

Configured in application.properties: properties

video.translation.timeRequired=5 (in seconds)