

Engineering Challenge (Take Home)

Introduction

In this exercise, you will create an imaginary real-time system that emulates a video supply chain. The system receives video metadata from a provider via a file stream and processes them for publication on TV 2 PLAY.

Even though this is an emulation, extensibility, and quality are important, so be deliberate about the architecture of the system and what kind of quality attributes you want to build in. The exercise is split into 2 parts, each extending the system's functionality.

You can use any common imperative programming language (e.g. Java, Python, Go, Javascript), framework, library, IDE, etc. to demonstrate your best work. However, to keep your code package simple, we discourage using microservices, and queuing systems such as Kafka, REST APIs, RPC, etc. If such components are needed, please emulate them in code e.g., by implementing an in-memory queue.

Ingest TV 2 video metadata

With your take-home challenge, you have received a `tv2-video-metadata-ingest.txt` file containing video metadata. Video metadata objects must be parsed and ingested into your system from this file. Each object should only be ingested once. When all video metadata has been consumed, your system should terminate.

Each line in the ingest file represents a video metadata object separated by a new line (`\n`). An object consists of an ID, title, provider, encoding time, and publication timeout.

- **id** is the UUID of the publication
- **title** is the title of the video
- **provider** is the company providing the publication.
- **encodingTime** indicates how long it takes to encode the video in seconds.
- **publicationTimeout** indicates the maximum time a publication is allowed to process from the encoding is started until the content is published on TV 2 PLAY, also in seconds. If this time is exceeded, the publication is considered a publication failure.

tv2-video-metadata-ingest.txt

```
{ // video metadata object
  "id": "020d6aba-4913-4564-a133-4f8d2f730bf5", // UUID of the publication
  "title": "Sommerdrømme", // title of the video

  "provider": "TV 2", // provider of the publication
  "encodingTime": 3, // time it takes to encode the video (seconds)
  "publicationTimeout": 7 // max time from encoding to publication (seconds)
}
...
```

Part 1 Challenge Prompt

In this first part, you will create a simulation of a system that ingests TV 2 video metadata, kicks off an encoder, and publishes the video on TV 2 PLAY. Video metadata objects from the ingest file stream should be delivered for encoding at a maximum rate of 2 objects per second. A maximum of 10 encoders are available. Encoders should start encoding as soon as one is available. Each encoder can only work on one video metadata object at a time. Once an encoder has finished, the encoding should be handed over to a TV 2 PLAY publisher for publication on TV 2 PLAY.

To simulate the time, it takes to encode a video, the encoding time should follow the `encodingTime` field on each video metadata object. To simulate the time, it takes to start a publishing job, a TV 2 PLAY publisher should wait 2 to 6 seconds before it is ready to publish to TV 2 PLAY. A publisher needs to be started for each publication. A publisher can only publish one publication before it is shut down.

As this is a simulation, there is no actual encoding of video or publication to TV 2 PLAY, and we do not expect you to implement logic to handle encoding or publication. We do expect console or log output that allows interviewers to clearly understand and follow your system's operations as it runs in real-time. Please see Grading for more information.

Part 2 Challenge Prompt

Your team has discovered that it takes a long time (2 to 6 seconds) to start up a TV 2 PLAY publisher resulting in a lot of publication failures. To compensate, your team has decided to start up a TV 2 PLAY publisher when the encoding of an object begins instead of when the encoding is complete. Each TV 2 PLAY publisher can only publish one publication before it is shut down.

Video encoding is time-consuming, leading to frequent publication failure. To better utilize encoders, your team decides to queue encoded publications before they are handed over or picked up by a TV 2 PLAY publisher. The queue holds encoded publications until a TV 2 PLAY Publisher is ready.

In this second part, you will adapt your implementation with the requirements above to minimize the chance of a video publication failure. Can you think of other ways in which you can reduce video publication failure? Maybe not all video metadata objects are equal.

Code Submission

For code submission, you have two options:

- Create a **private** repository on [GitHub](https://github.com) and share the repository with the individuals specified in the email you received together with this challenge.
- Zip/tarball your code and send it by email to the individuals specified when you received this challenge.

Please do not submit binaries or place your submission on a public repository - this could spoil the fun for others!

Grading

You are expected to build a system that fulfills each element of this grading list to the best of your ability.

- The code is valid and runnable (via CLI).
- The code has automated testing with appropriate coverage.
- The code has production-grade cleanliness and documentation (as if someone else had to work with your code).
- The implementation has appropriate usage of data structures, concurrency, and design patterns.
- The system has an extensible architecture with appropriate separation of concerns.
- The code has console or log output that allows the interviewers to clearly understand and follow the system's operations as it runs. Whenever events occur in your system (e.g., encoding, publishing, failed publications, etc.) please make sure to output a description of the event.
- Contains a README.md file containing:
 - Instructions on how to build, test, and run your code in a local environment.
 - The reasoning for your design.
 - Assumptions you have made.
 - Any other design choices you would like the interviewer to know?
 - Any relevant visuals (drawings, diagrams, etc.) you may have created as part of solving the challenge.