STARZPLAY

Java Developer - Code Assignment

Objective

The purpose of the assignment is to implement a service that will provide content metadata information to client applications of a video streaming platform.
The content will be served based on the requirements and business rules detailed in this document.
The implementation of the service will be done in Java 8. A recommended framework for this is Spring Boot, but any other framework can be used. Any available open source java library can be used as a maven dependency.

Requirements

The service will expose one single GET API:
GET http://server_address/media?filter=censoring&level=uncensored

Only valid value for query parameter 'filter' is 'censoring'. But new filters could be added in the future, so the solution must be designed taking this into account, being flexible to add new filters.

Valid values for query parameter 'level' are 'uncensored' and 'censored'.

This API will retrieve the content metadata from this external address:
https://de8a7d97-b45e-401b-b30e-39ae7b922405.mock.pstmn.io/api/v1.0/mediaCatalog/titles/movies

Based on the content received from the external provider, our API will apply a filter to the response, based on the level requested as query parameter, and will return the result after the filter is applied, using the same JSON format.

The content JSON will contain an array of "entries", each entry representing a movie. Each entry will have a "peg$contentClassification" field. This field will determine if the movie has only an uncensored version ("peg$contentClassification": "Uncensored") or if the movie includes both a censored and uncensored version ("peg$contentClassification": "Censored"). This field can also be empty, meaning that the movie is not classified, and no filter should be applied. Our filter will only be applied to movies classified as "Censored", and will remove one of the versions included, returning only the censored version of the movie or the uncensored version of the movie, depending on the requested level.

The different versions of the movie are returned in the “media” array of the “entry” entity. Each media is identified by a “guid” string. Censored medias have a guid that ends with the letter “C”. For example: "guid": "THINKLIKEAMANY2012MC".

Our filter has to return the correct media for censored movies. If we are requesting “censored” level, the censored media should be returned, and the uncensored media should be removed from the media array. If we are requesting “uncensored” level, the uncensored media should be returned and the censored media should be removed from the media array.

In any case, our response should only contain 1 element in the media array.

This is a summary table of the cases:

Query parameter level	entry contentClassification	Action
-	empty/null	No filter required
-	Uncensored	No filter required
censored	Censored	Remove uncensored media
uncensored	Censored	Remove censored media

Response

The response should contain all the project sources, together with a documentation file.

Preferred delivery method for the project to be evaluated is a Git repository, like GitHub or BitBucket.

The source must contain the implementation of the assignment, and a unit test suite that verifies the correct behavior.

The documentation must cover the following items:

- Detailed instructions about how to compile and deploy the service
- Documentation about the framework or libraries used
- Any assumption considered for the implementation