**Video CMS Task**

DATABASE NAME – video\_cms

DATABASE HOST – 216.240.134.115

DATABASE USER – task\_user

DATABASE PASSWORD - lahagora@2022

DATABASE PORT – 5432

This is database contains the data for a Video CMS backend.

You are free to add dummy data if you want for this task.  
  
If you do not know what a video CMS is, then you can read it from the below link <https://www.wowza.com/blog/video-cms-what-it-is>

Your task is to build a backend that will fetch the data from the database. The database is PostgreSQL and all the APIs will be in GET method.

The framework that you need to use is FatsAPI.   
<https://pypi.org/project/fastapi/><https://fastapi.tiangolo.com/>

You need to build the following APIs.

You are free to add any other API if you want to showcase your expertise.

1. Get all movies list. The response payload should be as the following  
  
[

{

id: 1,

title: some\_title,

release\_date: some\_date

},

{

id: 2,

title: some\_title,

release\_date: some\_date

},

{

id: 3,

title: some\_title,

release\_date: some\_date

}

]

2. Get Movie Details by id. The response payload should be as follows

{

title: some\_title,

slug: some\_slug,

rated: some\_rating,

quality: some\_quality,

description: some\_description,

id: some\_id,

region: some\_region,

image: some\_image,

original\_language: some\_language,

trays: [

{

title: some\_title,

slug: some\_slug,

rated: some\_rating,

quality: some\_quality,

release\_date: some\_release\_date,

description: some\_description,

id: some\_id,

region: some\_region,

image: some\_image,

original\_language: some\_language

},

{

title: some\_title,

slug: some\_slug,

rated: some\_rating,

quality: some\_quality,

release\_date: some\_release\_date,

description: some\_description,

id: some\_id,

region: some\_region,

image: some\_image,

original\_language: some\_language

}

]

}

IMPORTANT: Trays are data for content with same category.

3. Get Movie Metadata By id with the following response example

{

"subtitles": [

{

"lang": "ENGLISH",

"url": "https://english.url"

},

{

"lang": "HINDI",

"url": "https://hindi,url"

},

{

"lang": "GUJARATI",

"url": "https://gujarati.url"

}

],

"id": "1",

"audiotracks": [

{

"lang": "ENGLISH",

"url": "https://english.url"

},

{

"lang": "HINDI",

"url": "https://hindi,url"

},

{

"lang": "GUJARATI",

"url": "https://gujarati.url"

}

],

"ads": "True",

"drm\_data": {

"widevine\_laurl": {

"keySystem": "Widevine",

"licenseServerURL": "<http(s)://license-server-host/path?param=value>",

"url": "https://www.youtube.com/watch?v=OWfq\_JlvJxM"

},

"play\_ready": {

"encodingType": "PlayReadyLicenseAcquisitionUrl",

"encodingKey ": "PlayReadyLicenseServerUrl",

"url": "https://www.youtube.com/watch?v=OWfq\_JlvJxM"

}

},

"ads\_data": {

"advertising\_content\_id": 2,

"network\_id": 2,

"adFormat": "VAST",

"uri": "https://vast-videos.tech2020solutions.net/single.php?skip=1",

"cue\_points": [

{

"type": "PREROLL",

"enable": true

},

{

"type": "MIDROLL",

"enable": true,

"position": [

"120",

"300",

"480",

"910",

"1225"

]

},

{

"type": "POSTROLL",

"enable": true

}

]

},

"is\_live": false

}

]

}

NOTE: Comment each logic.   
NOTE: Finally push your code to any public repository like github/gitlab