

## توضیح اسکریپت DASH محلی:

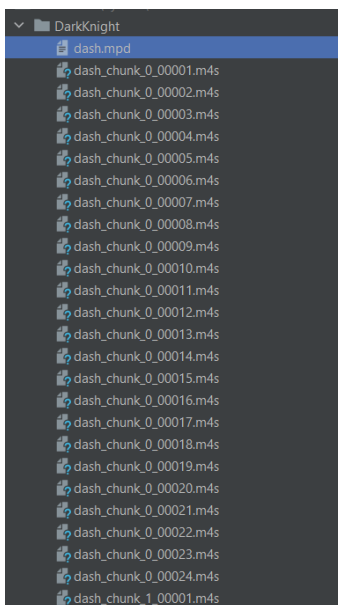
برای این کار از کد زیر استفاده شده است:

```
def monitor(ffmpeg, duration, time_, time_left, process):
    per = round(time_ / duration * 100)
    sys.stdout.write(
        "\rTranscoding...(%%) %s left [%s%s]" %
        (per, datetime.timedelta(seconds=int(time_left)), '#' * per, '-' *
        (100 - per))
    )
    sys.stdout.flush()

video = ffmpeg_streaming.input('The Dark Knight (2008).mp4')

dash = video.dash(Formats.h264())
dash.auto_generate_representations()
dash.output('./DarkKnight/dash.mpd', monitor=monitor)
```

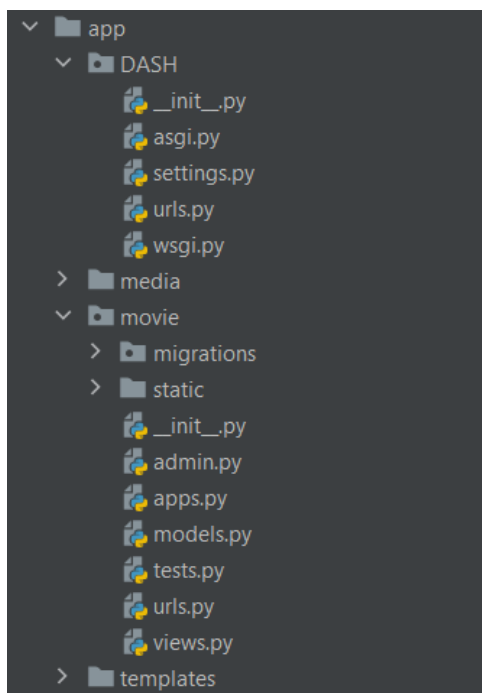
در این کد ابتدا در input آدرس یک فایل ویدیویی گرفته می شود. سپس به کمک video.dash یک شی از dash ساخته میشود و با auto\_generate\_representations تمامی representation های ممکن براساس کیفیت موجود فیلم به صورت خودکار تشخیص داده می شوند. در انتها نیز به کمک output نتیجه در دایرکتوری مورد نظر ذخیره می شود. جهت وجود progress bar حين اجرا نیز تابع monitor از داکيومنتيشن استفاده شده است.



پس از اجرای اسکریپت فوق یک فایل xml تحت عنوان dash.mpd ساخته میشود که در کنار آن های ویدیویی قرار گرفته اند.

## توضیح بک‌اند و فرانت‌اند:

برای بک‌اند از Django و فرانت‌اند از bootstrap کمک گرفته شده است. ساختار پروژه به صورت زیر می‌باشد:



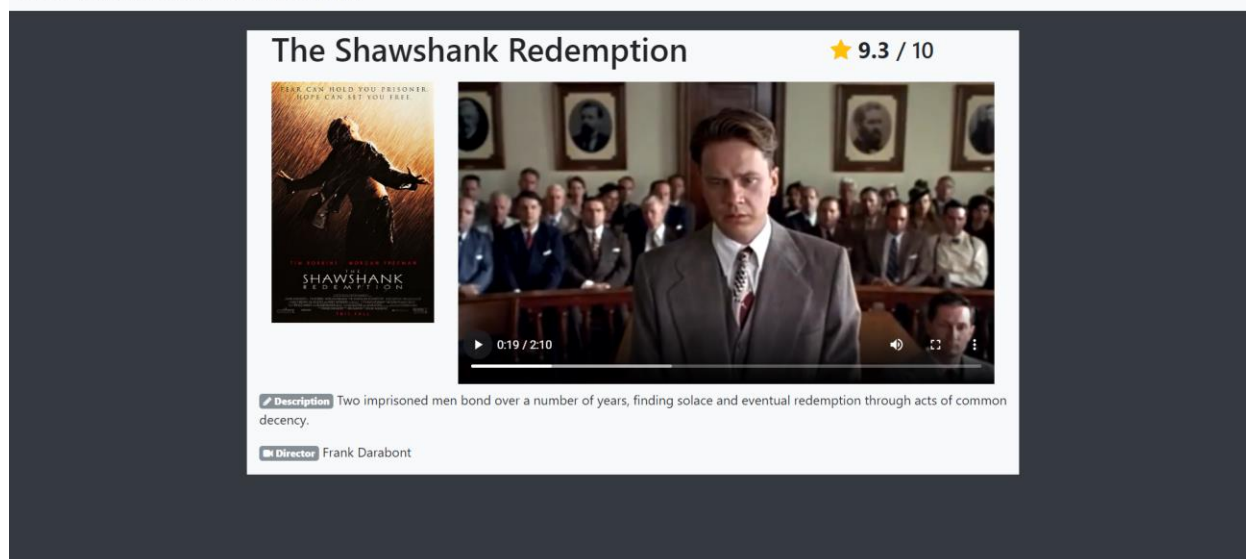
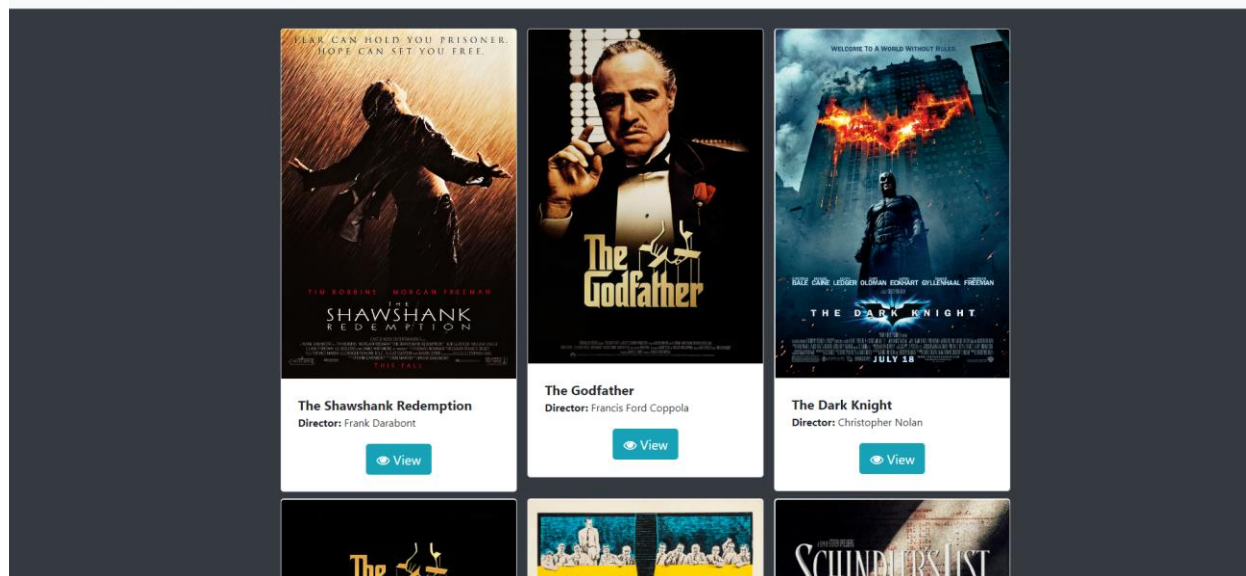
**DASH:** در این فولدر اپلیکیشن اصلی جنگو که حاوی settings.py می‌باشد و از wsgi آن توسط gunicorn در ادامه استفاده شده است قرار دارد.

**media:** تمامی تصاویر پوستر ویدیو که هنگام آپلود در دیتابیس به کمک پنل ادمین جنگو صورت می‌گیرد در این محل ذخیره می‌شود.

**movie:** اپلیکیشنی که کاربر توسط nginx مشاهده می‌کند این اپلیکیشن می‌باشد. در models.py جدول به صورت ORM و در views.py نیز endpoint های مربوطه قرار گرفته شده است. برای اتصال این view ها به url نیز از urls.py استفاده شده است.

**templates:** در این محل فایل های HTML قرار گرفته اند.











**static:** در این محل فایل های js و bootstrap قرار گرفته اند.



**نکته:** در قسمت دیتابیس جهت فعال شدن حالت DASH، یک فیلدی تحت عنوان dash mode تعریف شده است. لذا جهت آپلود فایل DASH چه به صورت local چه CDN، پس از دریافت URL ای که فایل mpd در آن موجود می‌باشد، ابتدا بایستی به <http://localhost:3000/admin> مراجعه کنید و سپس با یوزرنیم و پسورد admin و admin لاگین کنید. در جدول Movies روی فیلمی که می‌خواهید تریلر آن به صورت DASH اجرا شود علاوه بر trailer link بایستی تیک گزینه dash mode را نیز بزنید. اگر این تیک غیرفعال باشد حالت HLS در نظر می‌گیرد و در سمت فرانت‌اند اسکرپت HLS را بارگذاری می‌کند.

## توضیح داکرایز کردن پروژه

ساختار کلی پروژه به صورت زیر می باشد:

 .idea	6/30/2022 2:27 PM	File folder	
 app	6/30/2022 1:59 PM	File folder	
 nginx	6/25/2022 9:48 PM	File folder	
 .env	6/25/2022 9:36 PM	ENV File	1 KB
 .gitignore	6/25/2022 12:59 AM	Text Document	1 KB
 docker-compose.yml	6/25/2022 9:59 PM	YML File	1 KB
 Dockerfile	6/25/2022 9:45 PM	File	1 KB
 entrypoint.sh	6/25/2022 9:35 PM	Shell Script	1 KB
 readme.md	6/30/2022 2:25 PM	MD File	1 KB
 requirements.txt	6/25/2022 9:51 PM	TXT File	1 KB

طبق ساختار فوق، مشاهده می کنیم یک Dockerfile قرار گرفته است که برای اجرای جنگو به کمک gunicorn است. به صورتی که داریم:

```
FROM python:3.8.5-alpine

RUN pip install --upgrade pip

RUN apk add build-base

COPY ./requirements.txt .
RUN pip install -r requirements.txt

COPY ./app /app

WORKDIR /app

COPY ./entrypoint.sh /
ENTRYPOINT ["sh", "/entrypoint.sh"]
```

در این داکر فایل ابتدا pip آپگریت شده است. سپس build-base که حاوی gcc است (و کتابخانه pillow که image جنگو از آن استفاده می کند) نصب شده است. سپس requirements.txt به محل فعلی در container کپی و محتویات آن نصب شده است. در ادامه فولدر app که حاوی کل کدهای جنگو است به /app کانتینر منتقل شده است و با تغییر دایرکتوری به /app، اسکریپت entrypoint.sh کپی و اجرا می شود که حاوی محتویات زیر است:

```
#!/bin/sh



python manage.py migrate
```

```
python manage.py collectstatic

gunicorn DASH.wsgi:application --bind 0.0.0.0:8000
```

طبق اسکریپت فوق مشاهده میکنیم ابتدا database migration صورت میگیرد و با collectstatic نیز محتویات فایل static جنگو به /static کپی میشود. سپس به کمک gunicorn، اپلیکیشن جنگو روی پورت 8000 اجرا می شود.

با توجه به ساختار پروژه مشاهده میکنیم فولدر nginx هم موجود است که در داخل آن داریم:

Name	Date modified	Type	Size
 default.conf	6/30/2022 1:38 PM	CONF File	1 KB
 Dockerfile	6/25/2022 9:42 PM	File	1 KB

که در default.conf کانفیگ جهت اجرای جنگو قرار گرفته شده است که مشابه کانفیگ قرار گرفته در صورت پروژه است. Dockerfile نیز این فایل را به جای default ای که در nginx دارد کپی میکند.

و در آخر فایل docker-compose.yml را داریم:

```
version: "3"
services:
  django_gunicorn:
    volumes:
      - static:/static
    env_file:
      - .env
    build:
      context: .
    container_name: django
    ports:
      - "8000:8000"
  nginx:
    build: ./nginx
    volumes:
      - static:/static
    ports:
      - "3000:3000"
    depends_on:
      - django_gunicorn

volumes:
  static:
```

در این فایل دو سرور میسازد که یکی برای جنگو و دیگری nginx است. هر دو سرویس از static volume استفاده میکنند و سرویس nginx وابسته به Django است. در آخر میتوان با دستور زیر پروژه را اجرا کرد و در پورت 3000 برنامه در دسترس است:

```
docker-compose up --build
```

ارتباط فایل mpd. با تب network:

<input type="checkbox"/> dash_init_0.m4s	200	xhr	XHRLoader.j...	1.1 kB	13 ms		
<input type="checkbox"/> dash_init_3.m4s	200	xhr	XHRLoader.j...	1.0 kB	15 ms		
<input type="checkbox"/> dash_chunk_0_00001.m4s	200	xhr	XHRLoader.j...	688 kB	31 ms		
<input type="checkbox"/> dash_chunk_3_00001.m4s	200	xhr	XHRLoader.j...	36.1 kB	17 ms		
<input type="checkbox"/> dash_chunk_3_00002.m4s	200	xhr	XHRLoader.j...	41.9 kB	14 ms		
<input type="checkbox"/> dash_chunk_0_00002.m4s	200	xhr	XHRLoader.j...	555 kB	25 ms		
<input type="checkbox"/> dash_chunk_0_00003.m4s	200	xhr	XHRLoader.j...	384 kB	16 ms		
<input type="checkbox"/> dash_chunk_0_00004.m4s	200	xhr	XHRLoader.j...	663 kB	25 ms		
<input type="checkbox"/> dash_chunk_0_00005.m4s	200	xhr	XHRLoader.j...	1.1 MB	37 ms		
<input type="checkbox"/> dash_chunk_3_00003.m4s	200	xhr	XHRLoader.j...	41.7 kB	12 ms		
<input type="checkbox"/> dash_chunk_3_00004.m4s	200	xhr	XHRLoader.j...	41.9 kB	13 ms		
<input type="checkbox"/> dash_chunk_3_00005.m4s	200	xhr	XHRLoader.j...	41.8 kB	13 ms		
<input type="checkbox"/> dash_chunk_3_00006.m4s	200	xhr	XHRLoader.j...	41.6 kB	12 ms		
<input type="checkbox"/> dash_chunk_3_00007.m4s	200	xhr	XHRLoader.j...	41.6 kB	13 ms		
<input type="checkbox"/> dash_chunk_3_00008.m4s	200	xhr	XHRLoader.j...	41.8 kB	12 ms		

همانطور که مشاهده میکنیم هنگام پخش در تب network مجموعه ای از فایل های m4s موجود می باشد. این فایل ها داخل فایل mpd. موجود می باشد که در این فال xml، مشخص شده است که چه chunk هایی از هر ویدیو در دسترس است که هر کدام از مجموعه chunk ها برای کیفیت های متفاوتی نیز طراحی شده اند. سپس به کمک الگوریتم ABR به صورت خودکار تشخیص داده می شود که طبق پهنای باند موجود در کاربر کدام یکی از مجموعه chunk ها انتخاب شوند و طبق آن فایل های m4s مربوطه رکوئست و دریافت می شوند و در سمت کلاینت پخش خواهند شد.

## ارتباط فایل m3u8 با تب network:

<input type="checkbox"/> seg-1-f1-v1-a1.ts	200	xhr	hls.js@1:1	217 kB	1.87 s	
<input type="checkbox"/> blob:http://localhost:300...	200	text/ja...	Other	0 B	1 ms	
<input type="checkbox"/> index-f2-v1-a1.m3u8	200	xhr	hls.js@1:1	1.2 kB	1.04 s	
<input checked="" type="checkbox"/> js.js	200	script	content.js:32	1.4 kB	3 ms	
<input type="checkbox"/> encryption-f2.key	200	xhr	hls.js@1:1	138 B	2.04 s	
<input type="checkbox"/> data:image/svg+xml;...	200	svg+x...	Other	(mem...	0 ms	
<input checked="" type="checkbox"/> favicon.ico	404	text/h...	Other	3.0 kB	31 ms	
<input type="checkbox"/> seg-2-f2-v1-a1.ts	200	xhr	hls.js@1:1	253 kB	1.67 s	
<input type="checkbox"/> seg-3-f2-v1-a1.ts	200	xhr	hls.js@1:1	386 kB	1.72 s	
<input type="checkbox"/> index-f3-v1-a1.m3u8	200	xhr	hls.js@1:1	1.2 kB	1.11 s	
<input type="checkbox"/> encryption-f3.key	200	xhr	hls.js@1:1	142 B	1.08 s	
<input type="checkbox"/> seg-4-f3-v1-a1.ts	200	xhr	hls.js@1:1	675 kB	2.34 s	
<input type="checkbox"/> seg-5-f3-v1-a1.ts	200	xhr	hls.js@1:1	534 kB	1.90 s	
<input type="checkbox"/> seg-6-f3-v1-a1.ts	200	xhr	hls.js@1:1	506 kB	1.86 s	
<input type="checkbox"/> seg-7-f3-v1-a1.ts	200	xhr	hls.js@1:1	648 kB	1.98 s	
<input type="checkbox"/> seg-8-f3-v1-a1.ts	(pendi...	xhr	hls.js@1:1	0 B	Pending	

وقتی به محتوای فایل m3u8 ای که برای ویدیو دادیم توجه کنیم مشاهده میکنیم ساختار زیر را دارد:

```
EXTM3U#
EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=238742,RESOLUTION=256x144,FRAME-#
"RATE=23.974,CODECS="avc1.4d400c,mp4a.40.2
index-f1-v1-a1.m3u8
EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=429494,RESOLUTION=426x240,FRAME-#
"RATE=23.974,CODECS="avc1.4d4015,mp4a.40.2
index-f2-v1-a1.m3u8
EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=806292,RESOLUTION=640x360,FRAME-#
"RATE=23.974,CODECS="avc1.4d401e,mp4a.40.2
index-f3-v1-a1.m3u8
EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1450553,RESOLUTION=854x480,FRAME-#
"RATE=23.974,CODECS="avc1.4d401e,mp4a.40.2
index-f4-v1-a1.m3u8
EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1940872,RESOLUTION=1280x720,FRAME-#
"RATE=23.974,CODECS="avc1.4d401f,mp4a.40.2
index-f5-v1-a1.m3u8
```

با توجه به ساختار فوق مشاهده میکنیم برای رزولوشن های مختلف فایل های index متفاوتی را قرار داده است. در tab نتورک رزولوشن 640x480 را انتخاب کرده است و index-f3-v1-a.m3u8 را بارگذاری کرده است. داخل آن نیز داریم:

```
EXTM3U#  
EXT-X-TARGETDURATION:12#  
EXT-X-ALLOW-CACHE:YES#  
EXT-X-PLAYLIST-TYPE:VOD#  
"EXT-X-KEY:METHOD=AES-128,URI="encryption-f3.key#  
EXT-X-VERSION:3#  
EXT-X-MEDIA-SEQUENCE:1#  
.EXTINF:5.756#  
seg-1-f3-v1-a1.ts  
.EXTINF:5.756#  
seg-2-f3-v1-a1.ts  
.EXTINF:5.756#  
...  
seg-24-f3-v1-a1.ts  
.EXTINF:2.669#  
seg-25-f3-v1-a1.ts  
EXT-X-ENDLIST#
```

مشاهده میکنیم که اطلاعات segment های مختلف در این فایل موجود است که در تب network به ترتیب دریافت و به صورت streaming به کاربر پخش می شوند.



لینک های استفاده شده از CDN در پروژه:

**DASH:**

[https://mmimdbtrailer.arvanvod.com/00aYnokPbg/qD1B20Jl9E/h\\_108\\_200,182\\_400,272\\_448,362\\_448,544\\_448,k.mp4.list/manifest.mpd](https://mmimdbtrailer.arvanvod.com/00aYnokPbg/qD1B20Jl9E/h_108_200,182_400,272_448,362_448,544_448,k.mp4.list/manifest.mpd)

[https://mmimdbtrailer.arvanvod.com/00aYnokPbg/0Kydv8DaNG/h\\_144\\_200,238\\_400,358\\_800,478\\_999,716\\_999,k.mp4.list/manifest.mpd](https://mmimdbtrailer.arvanvod.com/00aYnokPbg/0Kydv8DaNG/h_144_200,238_400,358_800,478_999,716_999,k.mp4.list/manifest.mpd)

**HLS:**

[https://mmimdbtrailer.arvanvod.com/00aYnokPbg/LgAQYZPaj8/h\\_144\\_200,240\\_400,360\\_800,480\\_1500,720\\_1986,k.mp4.list/master.m3u8](https://mmimdbtrailer.arvanvod.com/00aYnokPbg/LgAQYZPaj8/h_144_200,240_400,360_800,480_1500,720_1986,k.mp4.list/master.m3u8)

[https://mmimdbtrailer.arvanvod.com/00aYnokPbg/xJ7d0OlX1z/h\\_144\\_200,240\\_400,360\\_586,480\\_586,720\\_586,k.mp4.list/master.m3u8](https://mmimdbtrailer.arvanvod.com/00aYnokPbg/xJ7d0OlX1z/h_144_200,240_400,360_586,480_586,720_586,k.mp4.list/master.m3u8)