

به نام خدا
گزارش پروژه نهایی درس سیستم های چند رسانه ای

امیر اله ورن ۹۷۲۳۰۱۰

توضیح کد:

نکات:

- ۲ ویدیو برتر imdb برای local dash streaming
- ۲ ویدیو برتر دوم imdb برای dash با استفاده از سرویس ویدیو آروان
- ۲ ویدیو برتر دوم imdb برای hls با استفاده از سرویس ویدیو آروان

- توضیح اسکریپت:

اسکریپت مربوطه در فایل local_dash.py قرار داده شده است.
در این اسکریپت با استفاده از کتابخانه python ffmpeg video streaming ابتدا ویدیو ورودی لود میشود. و سپس با فرمت h264 به ویدیو dash تبدیل میشود. در این مرحله presentation های مختلفی به صورت automate ایجاد میشوند. و در نهایت نتیجه در فولدر مربوطه ذخیره میشود.

- بک اند:

در فایل databae.py ابتدا اتصال به دیتابیس صورت میگیرد و سپس جدول movies ساخته میشود و داده های مورد نیاز ۶ فیلم برتر imdb در آن insert می شوند.
قسمت api ها در فایل app.py قرار گرفته اند.
در تابع index فایل html صفحه اول سایت برگردانده میشود.
در تابع getimage فایل های تصویر برگردانده میشوند.
در تابع get_dash فایل های local dash برگردانده میشوند.
در تابع movie فایل html مربوط به ویدیو انتخاب شده برگردانده میشود. با توجه به نکته بالا html برای هر نوع vod به صورت متفاوت پیاده سازی شده است.
در تابع api_movie اسم و مسیر پوستر ویدیو و امتیاز آن از ده برای همه ویدیو ها از دیتابیس خوانده شده و به عنوان نتیجه برگردانده میشود.
در تابع api_movie_info همه اطلاعات یک ویدیو با توجه به id آن از دیتابیس خوانده شده و برگردانده میشود.

- فرانت اند:

در فایل index.html در مسیر Front/index.html صفحه اول که آن لیست فیلم ها موجود است با استفاده از html css bootstrap پیاده سازی شده است.

در انتهای فایل ذکر شده با استفاده از js یک درخواست برای backend ارسال میشود. و لیست فیلم ها به صورت یک کامپوننت در html قرار میدهد.

فایل های Front/movies/x/index.html شامل صفحات جزئیات فیلم ها است.

Front/movies/۱ برای نمایش جزئیات ۲ ویدیو اول - local dash

Front/movies/۲ برای نمایش جزئیات ۲ ویدیو دوم - dash

Front/movies/۳ برای نمایش جزئیات ۲ ویدیو سوم - hls

در این صفحات مشابه صفحه اصلی با استفاده از js یک درخواست برای بک اند ارسال شده و اطلاعات را در قالب تگ های html نمایش میدهد.

سوالات:

۱.

<input type="checkbox"/> index-f1-v1-a1.m3u8	200	xhr	hls.js@latest:1	(disk cache)	1 ms	
<input type="checkbox"/> encryption-f1.key	200	xhr	hls.js@latest:1	(disk cache)	2 ms	
<input type="checkbox"/> seg-1-f1-v1-a1.ts	200	xhr	hls.js@latest:1	(disk cache)	2 ms	
<input type="checkbox"/> blob:http://localhost:8000/fcedf36d-bb40-4ba8-9205-72058c08a2f8	200	text/javascript	Other	0 B	2 ms	
<input type="checkbox"/> index-f4-v1-a1.m3u8	200	xhr	hls.js@latest:1	(disk cache)	2 ms	
<input type="checkbox"/> encryption-f4.key	200	xhr	hls.js@latest:1	(disk cache)	1 ms	
<input type="checkbox"/> seg-2-f4-v1-a1.ts	200	xhr	hls.js@latest:1	(disk cache)	3 ms	
<input type="checkbox"/> seg-3-f4-v1-a1.ts	200	xhr	hls.js@latest:1	(disk cache)	4 ms	
<input type="checkbox"/> seg-4-f4-v1-a1.ts	200	xhr	hls.js@latest:1	(disk cache)	3 ms	
<input type="checkbox"/> seg-5-f4-v1-a1.ts	200	xhr	hls.js@latest:1	(disk cache)	4 ms	
<input type="checkbox"/> seg-6-f4-v1-a1.ts	200	xhr	hls.js@latest:1	(disk cache)	3 ms	
<input type="checkbox"/> seg-7-f4-v1-a1.ts	200	xhr	hls.js@latest:1	(disk cache)	2 ms	
<input type="checkbox"/> seg-8-f4-v1-a1.ts	200	xhr	hls.js@latest:1	(disk cache)	3 ms	
<input type="checkbox"/> seg-9-f4-v1-a1.ts	200	xhr	hls.js@latest:1	(disk cache)	3 ms	
<input type="checkbox"/> seg-10-f4-v1-a1.ts	200	xhr	hls.js@latest:1	(disk cache)	3 ms	
<input type="checkbox"/> seg-11-f4-v1-a1.ts	200	xhr	hls.js@latest:1	(disk cache)	5 ms	
<input type="checkbox"/> seg-12-f4-v1-a1.ts	200	xhr	hls.js@latest:1	(disk cache)	3 ms	
<input type="checkbox"/> seg-13-f4-v1-a1.ts	200	xhr	hls.js@latest:1	(disk cache)	3 ms	
<input type="checkbox"/> seg-14-f4-v1-a1.ts	200	xhr	hls.js@latest:1	(disk cache)	3 ms	
<input type="checkbox"/> seg-15-f4-v1-a1.ts	200	xhr	hls.js@latest:1	(disk cache)	3 ms	
<input type="checkbox"/> seg-16-f4-v1-a1.ts	200	xhr	hls.js@latest:1	335 kB	216 ms	
<input type="checkbox"/> seg-17-f4-v1-a1.ts	200	xhr	hls.js@latest:1	1.2 MB	600 ms	
<input type="checkbox"/> seg-18-f4-v1-a1.ts	200	xhr	hls.js@latest:1	1.4 MB	708 ms	
<input type="checkbox"/> seg-19-f4-v1-a1.ts	200	xhr	hls.js@latest:1	1.2 MB	608 ms	
<input type="checkbox"/> seg-20-f4-v1-a1.ts	200	xhr	hls.js@latest:1	1.4 MB	705 ms	
<input type="checkbox"/> seg-21-f4-v1-a1.ts	200	xhr	hls.js@latest:1	1.3 MB	2.59 s	
<input type="checkbox"/> seg-22-f4-v1-a1.ts	200	xhr	hls.js@latest:1	1.4 MB	741 ms	
<input type="checkbox"/> seg-23-f4-v1-a1.ts	200	xhr	hls.js@latest:1	332 kB	194 ms	
35 requests 8.6 MB transferred 27.6 MB resources Finish: 7.86 s DOMContentLoaded: 94 ms Load: 183 ms						

جریان HLS یک فناوری نرخ بیت تطبیقی است. این بدان معناست که وقتی ویدیو به HLS کدگذاری می‌شود، چندین فایل برای پهنای باند مختلف و وضوح‌های مختلف (اندازه‌های صفحه) ایجاد می‌شوند. جریان ها با استفاده از یک فایل فهرست M3U8 بر اساس اندازه صفحه نمایش و پهنای باند موجود به صورت بلادرنگ به مشتری نگاشت می شوند.

4	200	document	Other	8.8 kB	6 ms
bootstrap.bundle.min.js	200	script	⚡	(memory cache)	0 ms
dash.all.min.js	200	script	⚡	(memory cache)	0 ms
bootstrap.min.css	200	stylesheet	⚡	(disk cache)	9 ms
4	200	xhr	4242	615 B	4 ms
manifest.mpd	200	xhr	XHRLoader.is:90	(disk cache)	6 ms
The%20Godfather%20Part%20(1974).jpg	304	jpeg	4280	304 B	6 ms
init-f4-v1-x3.m4	200	xhr	XHRLoader.is:90	(disk cache)	2 ms
init-f4-a1-x3.m4	200	xhr	XHRLoader.is:90	(disk cache)	2 ms
seg-1-f4-v1-x3.m4s	200	xhr	XHRLoader.is:90	(disk cache)	4 ms
seg-1-f4-a1-x3.m4s	200	xhr	XHRLoader.is:90	(disk cache)	1 ms
seg-2-f4-v1-x3.m4s	200	xhr	XHRLoader.is:90	(disk cache)	19 ms
seg-2-f4-a1-x3.m4s	200	xhr	XHRLoader.is:90	(disk cache)	9 ms
seg-3-f4-v1-x3.m4s	200	xhr	XHRLoader.is:90	(disk cache)	3 ms
seg-4-f4-v1-x3.m4s	200	xhr	XHRLoader.is:90	(disk cache)	3 ms
seg-5-f4-v1-x3.m4s	200	xhr	XHRLoader.is:90	(disk cache)	2 ms
seg-6-f4-v1-x3.m4s	200	xhr	XHRLoader.is:90	(disk cache)	3 ms
seg-3-f4-a1-x3.m4s	200	xhr	XHRLoader.is:90	(disk cache)	4 ms
seg-4-f4-a1-x3.m4s	200	xhr	XHRLoader.is:90	(disk cache)	5 ms
seg-5-f4-a1-x3.m4s	200	xhr	XHRLoader.is:90	(disk cache)	4 ms
seg-6-f4-a1-x3.m4s	200	xhr	XHRLoader.is:90	(disk cache)	2 ms
seg-7-f4-a1-x3.m4s	200	xhr	XHRLoader.is:90	(disk cache)	2 ms

22 requests | 9.8 kB transferred | 8.2 MB resources | Finish: 929 ms | DOMContentLoaded: 92 ms | Load: 172 ms

وقتی کاربر پخش را فشار می دهد، برنامه یا پخش کننده ویدیو فایل MPD ویدیو را درخواست می کند. پس از

دریافت MPD، پخش کننده آن را تجزیه می کند تا نحوه پخش ویدیو را بفهمد.

سپس پخش کننده شروع به درخواست تکه های ویدیو به ترتیب از پیش تعیین شده می کند، تکه ها را

رمزگشایی می کند و ویدیو را به کاربر نمایش می دهد.

پخش کننده همچنین به طور مداوم شرایط پهنای باند را کنترل می کند. بسته به پهنای باند موجود، پخش کننده

یکی از نرخ بیت های تبلیغ شده در MPEG-DASH mpd را انتخاب می کند و از CDN درخواست می کند تا تکه

بعدی ویدیو را از آن نوع ارسال کند.

فایل MPD فقط فهرستی از جریان ها با فرمت های مختلف است تا با پهنای باند سازگار شود.