

# برنامه نویسی وب

## تمرین سوم

استاد درس:

مهسا سعیدی

مهلت نهایی ارسال پاسخ:

دوشنبه ۵ آذر ۱۴۰۳ - ساعت ۲۳:۵۹

امکان ارسال تمرین با تاخیر یک روز و کسر نمره با ضریب ۱۰ درصد امکانپذیر است.

## سرویس آپلود فایل

در این تمرین هدف ایجاد یک سرویس ساده جهت آشنایی و درک اولیه از معماری مبتنی بر میکروسرویس ها است. همچنین در این تمرین با Docker آشنایی اولیه پیدا خواهید کرد زیرا تمامی سرویس های مورد نیاز باید dockerize شده اجرا شوند.

سرویزی که شما پیاده سازی خواهید کرد یک سرویس آپلود فایل ساده جهت کار با انواع Message broker ها و Object Storage ها به صورت dockerize شده است.

در این سرویس کاربر فایلی را برای مثال در قالب عکس یا ویدیو در طی روندی که توضیح داده خواهد شد در یک ذخیره ساز مبتنی بر شی (Object storage) بارگذاری می کند. برای پیاده سازی این سرویس از لازم است که از سرویس های زیر استفاده کنید:

- Apache kafka + Zookeeper
- MinIO
- PostgreSQL
- Backend\_Scripts = Producer + Consumer

## سرویس Apache Kafka و Zookeeper

Apache kafka و Zookeeper با کمک همدیگر، به عنوان یک Message broker قدرتمند عمل می کنند که لازم است در این تمرین مورد استفاده قرار گیرند.

همانطور که می دانید Message broker ها نرم افزار های واسطه مانندی هستند که به برنامه های مختلف اجازه می دهند تا به طور مستقل و بدون نیاز به ارتباط مستقیم با هم، با یکدیگر پیام رد و بدل کنند. این نرم افزار ها مانند یک پستچی عمل می کنند که پیام ها را جمع آوری کرده، مرتب سازی می کنند و به مقصد مورد نظر می رسانند.

همچنین امکان موازی سازی در آن ها وجود دارد که در معماری های نظیر Pub/Sub و افزایش سرعت تبادل پیام کاربرد دارند.

لازم به ذکر است که Message broker ها از معماری Producer و Consumer بهره می‌برند و بسته ها توسط Producer در صف گذاشته شده و توسط Consumer برداشته می‌شود.

همچنین در صورت لزوم هر Consumer می‌تواند در چندین کانال ارتباطی Subscribe کند و بسته ها را از کانال های مختلف بخواند.

لینک زیر جهت آشنایی بیشتر با Apache Kafka و Zookeeper است:

<https://www.openlogic.com/blog/using-kafka-zookeeper>

## سرویس MinIO

در این تمرین لازم است از ذخیره ساز مبتنی بر شی MinIO استفاده کنید که مبتنی بر S3 است و عموماً در محیط های Local و Test کاربرد دارد.

لینک زیر جهت آشنایی بیشتر با MinIO است:

<https://www.simplyblock.io/glossary/what-is-minio/>

## شرح تمرین

**گام اول:** ابتدا یک Script به عنوان Producer بنویسید که به عنوان ورودی موارد زیر را از کاربر دریافت کرده و در سرویس پایگاه داده PostgreSQL ذخیره کرده و در kafka قرار دهد.

- یک آدرس (url) عکس یا ویدیو ( آدرس اینترنتی یا یک آدرس local در سیستم)
- شناسه فایل
- bucket\_name
- نام شی

**گام دوم:** یک Script به عنوان Consumer بنویسید که اطلاعات قرار گرفته شده فوق در Producer را دریافت کند و از url استخراج شده، فایل را به صورت stream دریافت کرده و به صورت stream در MinIO بارگذاری کند.

**گام سوم:** پس از بارگذاری در MinIO، Consumer باید یک Pre-signed URL توافق شده که دارای زمان انقضا می‌باشد را دریافت کرده و آن را در خروجی چاپ کند.

همچنین Consumer باید Etag و Size فایل بارگذاری شده را دریافت کرده و در جدول پایگاه داده PostgreSQL به‌روزرسانی کند. Size دریافت شده باید با Size واقعی فایل یکسان باشد و برای اطمینان از صحت عملیات بارگذاری می‌توانید پس از بارگذاری، فایل را در سیستم از MinIO بارگیری کنید و نمایش دهید و یا از طریق محیط گرافیکی MinIO که جلوتر معرفی خواهد شد آن را مشاهده کنید.

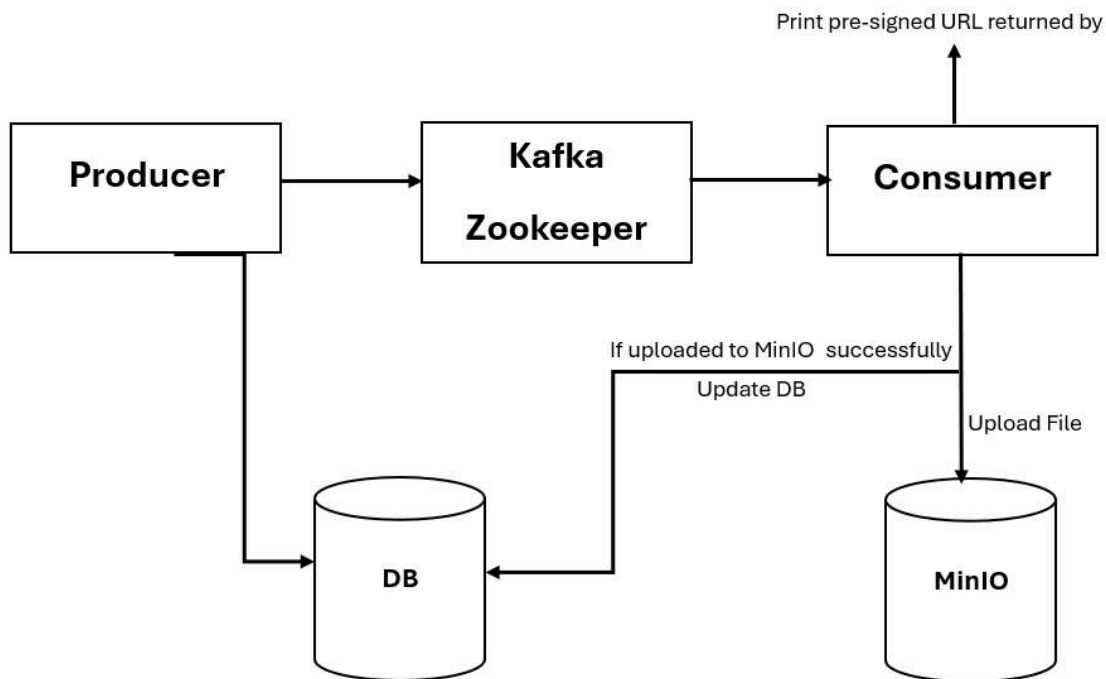
### توجه:

اگر فایلی که دریافت و بارگذاری می‌کنید دارای حجم زیاد باشد، می‌توانید به صورت Chunk های پشت سر هم فایل را دریافت و بارگذاری کنید. اندازه Chunk ها را می‌توانید به دلخواه خودتان تنظیم کنید.

شمای پیشنهادی جدول پایگاه داده:

id	File url	Bucket name	Object Name	Etag	Size
----	----------	-------------	-------------	------	------

## معماری کلی سرویس آپلود فایل



همچنین جهت کمک به شما فایل پیشنهادی `Dockerfile` و `docker-compose.yml` در پوشه تمرین جهت استفاده و بالا آوردن سرویس های جانبی قرار گرفته شده است.

شما باید واسط گرافیکی `kafka` و `MinIO` را در مرورگر در آدرس های زیر مشاهده کنید.

`Kafka_ui = localhost:8080`

`MinIO_ui = localhost:9001`

## امتیازی

در صورت بروز خطا در زمان آپلود یا پردازش، پیام به صف دیگری ارسال شود یا Consumer طوری برنامه ریزی شود که که چند بار سعی در پردازش دوباره پیام داشته باشد.

## نحوه تحویل

لطفا تمرین خود را به صورت یک فایل زیپ با نام StudentID\_Name\_Lastname\_HW3 و شامل فایل های زیر در courses ارسال کنید:

- تمامی فایل های کد و وابستگی های آن.
- گزارش تمرین که شامل اسکرین شات هایی از مراحل و توضیح آن هاست.
- تمرین دارای تحویل آنلاین است، بنابراین در استفاده از منابع و کدهایی که در اینترنت وجود دارد و شما قادر به توضیح آن ها نیستید خودداری کنید.
- در صورت استفاده از کدهای موجود در منابع آنلاین اطفأ منبع کد را به صورت comment در کد درج کنید.