

- در root پروژه دو فایل Dockerfile و docker-compose.yml ایجاد شده است که برای ران کردن برنامه در هر محیط دیگه ای نوشته شده است و به این دلیل دو تا فایل را ایجاد کردم که اگر هر کس دیگه ای خواست برنامه را ران کند نیاز نداشته باشد که حتما Cassandra و Kafka را در محیط خود نصب کرده باشد. ابتدا با استفاده از Dockerfile یک image از خود پروژه درست می کنیم و سپس با استفاده از فایل docker-compose.yml برنامه را اجرا می کنیم.

- در پکیج service یک پکیج به اسم config داریم که در این پکیج 3 کلاس وجود دارد که به اختصار آنها را توضیح می دهم

1. CassandraConfig : در این کلاس کانفیگ های لازم برای ایجاد جدول و key-space انجام شده است.

2. KafkaTopicConfig : در این کلاس دو topic برای پیام هایی که قراره بفرستیم و استفاده کنیم ایجاد شده است.

3. KafkaProducerConfig : در این کلاس تنظیمات مربوط به خود Kafka را انجام دادم.

در پکیج kafka نیز 3 تا کلاس داریم که هر 3 را به ترتیب زیر بررسی خواهیم کرد

1. KafkaMessageSender : در این کلاس یک متد نوشته ام که دو ورودی میگیرد یک ورودی آن پیغام می باشد و ورودی دیگر آن تاپیکی هست که قرار هست پیغام در آن تاپیک فرستاده شود.

2. MessageProducer : در این کلاس با استفاده از اسکچولر اسپرینگ هر 10 ثانیه یک نمایش تبلیغ و یک کلیک تبلیغ را ایجاد می کنم و از آن طرف با استفاده از کلاس MessageConsumer دو پیغام گرفته شده را در کلاس AddEvent اضافه میکنم و سپس به کمک پیغام دوم یک فیلد به نام clickTime کلاس AddEvent را آپدیت میکنم.

3. MessageConsumer : ابتدا پیغام اول در کلاس AddEvent مپ می شود و سپس این کلاس در دیتابیس ذخیره می شود و وقتی که پیغام دوم می رسد با استفاده از requestId آن کلاس AddEvent ذخیره شده را در دیتابیس میگیریم و سپس فیلد clickTime آن آپدیت می شود و در نهایت در دیتابیس کلاس AddEvent آپدیت می شود.