

## AdventureWorks Ürün İnceleme Uygulaması

Açıklama: Çevrimiçi bir ürün inceleme özelliği oluşturun. özellik, ürün incelemelerini kabul edecek, incelemeleri herhangi biri için tarayacak Uygunsuz dilde, incelemeyi onaylandıktan sonra canlı yayınlayın veya Onaylanmadıysa ve incelemeyi e-posta yoluyla bildirdiğinde arşiv Gözden geçirme durumu tamamlandıktan sonra (simüle edilir).

Gerekli bileşenler:

- Lay API Katmanı (HTTP)
- W AdventureWorks kullanarak veritabanı (Postgres veya MySQL)
- Postgres veya MySQL için
  - İşçi Katmanı # 1 - Gözden geçirme işlemcisi
  - Kuyruk, Mesajlaşma Veriyolu
  - İşçi Katmanı # 2 - Bildirimler
  - Raporlar / Sorgular

Gereksinimler, Bileşenler ve Kabul kriterleri:

Lay API Katmanı (HTTP):

- Müşteri HTTP üzerinden bir ürün incelemesi sunar:

```
kıvırmak -X POST http://0.0.0.0:8888/api/reviews \
-H 'İçerik Türü: uygulama / json' \
-d '{
  "name": "Elvis Presley",
  "e-posta": "theking@elvismansion.com",
  "productid": "8",
  "review": "Ürünü gerçekten seviyorum ve
    tavsiye etmek!"
}
```

- API HTTP yanıtı:

```
{
  "başarı": doğru
  "reviewID": [id tamsayı]
}
```

API, ürün incelemesini veritabanına yerleştirir ve işleme için bir sıraya.

W AdventureWorks Veritabanı - MySQL veya Postgres:

- <https://github.com/lorint/AdventureWorks-for-postgres>
- <https://sourceforge.net/projects/awmysql/files/latest/indir>
- Bunu desteklemek için gerekli veritabanı değişikliklerini oluşturun iş akışı

- İşçi # 1 - İnceleme İşlemcisi
  - İşçi kuyruk ve taramalardan bir inceleme kullanıyor uygunsuz dil için
  - Filtrelenmesi gereken kötü kelimeler: **BAD\_WORDS** = ["ücret", "nee", "vurgulu", "lütuf"] . Bir mesaj varsa Kötü bir kelime içeriyorsa, mesaj işaretlenmeli uygunsuz olarak
- Sıra, Mesaj Veriyolu
  - İnceleme için sıraya alma ve mesajlaşma mekanizması işlemci (gerektiği gibi) (Redis, RabbitMQ, vb.)
  - Mesajlar API'den İşçiye sıraya alınmalıdır # 1 ve İşçi # 2
- İşçi # 2 - Bildirim:
  - İnceleme onaylandıktan ve yayınlandıktan sonra, inceleyen e-posta yoluyla bilgilendirilir.
  - İşçi, işlenmesi için ilgili verileri almalıdır bildirim ancak bildirim tamamen simüle edildi (aslında hiç e-posta gönderilmez).

Yapılandırma notları:

Lütfen uygulama kurulum talimatlarını içeren bir README oluşturun tüm bileşenler için. Bu şunları içerebilir:

- Oluşturmak / çalıştırmak için Bash veya diğer kabuk komut dosyaları
- sql dosyaları
- Docker dosyaları (kullanılıyorsa)
- Varsa web sayfaları
- Günlüklerin yolu

- Ön gereklilikler (çalışma süreleri, lib'ler, vb.)

Not: Bu cilalı bir README olmak zorunda değildir ama Uygulamayı çalıştıran anlayışlı bir okuyucu olsun.

Yayıncılık:

Lütfen Git deposuna bağlantı verin. API konuşlandırılmışsa çevrimiçi, lütfen test etmek için cURL betiği veya komutu verin.

İpuçları

- Bazı sorunlarla karşılaşabileceğinizi unutmayın. sahip olabileceğiniz AdventureWorks veritabanındaki veriler düzeltmek. Bu bekleniyor. Lütfen bunlara dikkat edin sorunlar.
- Kuyruklama için lütfen Redis'i tercih ediyoruz ancak lütfen Hangi kuyruk çerçevesine aşına olursanız olun.
- İdeal olarak, tamamlanmış projeyi alabilmeliyiz ve yerel olarak komut dosyaları aracılığıyla çalıştırın. Sistemlerimiz Ubuntu Linux.
- Lütfen bana bazı talimatlar veren bir README oluşturun API ve ilgili bileşenleri oluşturma ve çalıştırma hakkında. README'nin fantezi olması gerekmez. Basit bir çek liste iyi olacak. Güvenle kim olduğunu varsayabilirsiniz README'yi inceleyin ve uygulamanızı çalıştırmayı deneyin geliştiricileri.
- Tüm işi bitirmezseniz, bu olduğunuz anlamına gelmez diskalifiye. Ne kadarının tamamlandığını gözden geçireceğiz ve işin kalitesi.
- Tamamlanmış ve çalışan bir projeye tüm diğerlerine öncelik verin hususlar
- Lütfen sorduğunuz soruları veya sorunları not edin karşılama. Bunları kesinlikle dikkate alacağız.

Bonus puanlar!

- Çalışan tüm bileşen hizmetleri (API, veritabanı, sıralar) Liman işçisi
- Gerektiğinde veritabanı geçişleri
- Gerektiği gibi birim testleri
- AWS'ye veya herhangi bir genel buluta dağıtım
- API güvenliği