Merhaba,

Seni tanıdığımız için çok mutluyuz. Umarız şirket olarak çalışma yöntemlerimizi ve hedeflediğimiz noktayı iyi bir şekilde ifade edebilmişizdir.

Aşağıda isterleri bulunan challenge'ı değerlendirme sürecinin bir adımı olarak sana yolluyoruz.

Proje için;
Başlıkların hepsini veya;

- Yalnızca API
- Yalnızca Worker
- API + Callback
- Worker + Callback
- API + Worker

Kısımlarından istediğin seçeneği seçebilirsin. Challenge'da **süre kısıtımız 2.5 haftadır**.

Bu challenge için kullanmak istersen, yaygın ve güncel olan istediğin PHP frameworkunu kullanabilirsin.

Seçtiğin kısımlar ve tamamlama süren bizim için ekstra değerlendirme kriterleri olacaktır.

Kolaylıklar,

Mobile Application Subscription Management API /Callback / Worker

iOS ya da Google mobile application'lar bu API'ı kullanarak in-app-purchase satın alma / doğrulama ve mevcut abonelik kontrolü yapabileceklerdir.

Worker tarafında ise database'de bulunan mevcut aktif aboneliklerin expire-date'leri gelenleri tekrar iOS ya da Google üzerinden sorgulayıp durumlarını ve expire-date lerini güncellenecektir.

Bu sistem birden fazla mobil uygulamaya aynı anda destek verebilir.

Detayları şu şekildedir:

API

Genel olarak mobil device lardan gelecek HTTP isteklerini karşılayacak.

Register

Bir mobil cihaz ilk defa açıldığında API'ımıza register olmalıdır, register işleminde ilgili cihazın uid, app id, language, operating-system (os) değerleri alınıp bir device tablosuna kayıt edilmelidir.

Aynı uid ile defalarca register isteği gelebilir. Bu durum handle edilmeli ve register OK cevabı döndürülmelidir. Register OK cevabında her device'a farklı olması koşulu ile bir client-token hazırlanıp response da döndürülmelidir.

Purchase

Mobil uygulama içerisinden yapılan satın alma isteğidir. Mobil client bu API'ya parametre olarak client-token ve receipt (anlamlı olmayan, rastgele bir bir hash olabilir) parametrelerini iletir.

API'mız bu gelen parametrelerdeki receipt hash'i ile iOS ya da Google'a doğrulama isteği atmalıdır, iOS ve Google API'larını mocklayarak kendiniz oluşturmanızı bekliyoruz, basitçe şu şekilde olabilir:

Aldığı receipt string değerinin son karakteri tek bir sayı ise OK cevabı verip bu cevap içerisinde status:true/false ve expire-date: Y-m-d H:i:s UTC +3 timezonenunda parametreleri döndürmesi yeterli olacaktır.

API'ımız client'tan aldığı isteği iOS ya da Google mock platformlarında doğrulayıp sonucu DB'ye işleyip client'a response dönmelidir.

Check Subscription

Mobil client her açıldığında veya gerekli gördüğü her adımda bu endpoint'i çağırabilir. Sadece client-token parametresi ile yaptığın isteğin sonucunda güncel abonelik durumu döndürülmelidir.

Worker

Cron'dan ya da supervisord gibi çeşitli server side tetikleyiciler vasıtası ile başlayıp DB de expire-date'i geçen ama iptal olmamış kayıtları os değerine göre tek tek iOS ya da Google mock platformları üzerinden yine aynı şekilde doğrulayıp DB'deki değerler güncellenmeli.

Buradaki önemli noktalardan bir tanesi DB'de bekleyen kayıtların adet olarak milyonlarca (10 milyon+) olabileceği göz önünde bulundurulup bu bekleyen kayıtların kısa sürede eritilebilmesi gerekiyor.

Ayrıca iOS ve Google API'ları mobile application bazlı olarak rate-limitleri bulunmaktadır. Bekleyen kayıtları eritirken bazı istekler (receipt değerindeki son 2 basamak 6'ya bölünebiliyorsa) iOS ve Google API'larından rate-limit hatası alabilir, bu durumda ilgili kayıt daha sonra tekrar denenmelidir.

Callback

API ve Worker kısımlarında herhangi bir device'ın abonelik durumunda bir değişiklik olursa; started, renewed, canceled şeklinde 3 farklı event oluşturulmalı ve bu eventler application bazlı olarak önceden DB'ye set edilmiş olan 3rd-party bir endpoint'e HTTP POST ile bildirilmelidir. Bu bildirimde appID, deviceID ve event bilgisi iletilmesi yeterli olacaktır.

Opsiyonel olarak eğer ilgili event için set edilmiş 3rd-party endpoint HTTP 200 veya 201 harici bir status verdiğinde bu event bildirimi tekrar gönderilebilmesi için bir mekanizma oluşturulabilir.

Callback sistemi API ve Worker yapıları içerisinde olabileceği gibi opsiyonel olarak ayrı asenkron kuyruk sistemi ile çalışan bir modül olarak da tasarlanabilir.

Raporlama

Uygulama, gün, işletim sistemi bazında başlayan, biten ve yenilenen abonelik sayılarını almak için rapor oluşturulmalı. Raporun DB tablosundan sql olarak çekilebilmesi yeterli, herhangi bir arayüze ihtiyaç yoktur.

Raporlama sayımı API, Worker veya Callback modüllerinde uygun yerlerde yapılabilir.

Teknik Koşullar

- Minimum PHP 8.1 kullanılmalıdır.
- Tercihen Laravel kullanılmalıdır.
- Bir device bir app altında aynı anda sadece bir aboneliği bulunabilir
- Bir device farklı app'ler olmak şartı ile birden fazla aboneliği olabilir.
- Her app'in iOS ve Google credential bilgileri farklıdır. (Mock API'mıza istek atarken header da basic olarak username:password şeklinde kullanılabilir.)
- DB dokümantasyonu dbdocs kullanılarak hazırlanıp bizlere iletilmelidir. (dbdocs)
- DB tablolarının özellikle device tablosunun milyonlarca kayıt altında çalışması beklenmektedir.

- API tarafında bir çok farklı app'ten aynı anda HTTP istekler gelecek yüksek trafik olacağı göz önünde bulundurulmalıdır.
- API dokümantasyonu swagger kullanarak docker build alındığında otomatik olarak oluşturulmalıdır.
- Yazılı beklentiler dışında geliştirilmiş dikkat çekmek istediğiniz özellikleri README dosyasında kısa başlıklar ile belirtebilirsiniz.
- İstenen yerde, dizayn şemaları için draw.io kullanılıp, xml dosyası iletilebilir.
- Proje github üzerinden bizimle paylaşılmalıdır.
- Projenin tüm bileşenleri dockerize edilmeli ve docker-compose ile localde çalıştırılabilir olmalıdır.

Değerlendirme

Değerlendirmede API ve Worker tarafında performans önemli bir kriterimiz olacak.

DB tablolarının performansı da diğer bir değerlendirme noktamız olacak. "Device", "Subscription" gibi verileri tutan tabloların milyonlarca kayda ulaşması durumlarının göz önüne alınması gerekmektedir.

Ayrıca;
- Hata yakalama
- Object oriented programlama ilkelerine uyum

konuları da değerlendirmede göz önüne alınacak.

Challenge yapmayı seçeceğiniz başlıkların değerlendirmemizdeki ağırlıklarını aşağıdaki gibi belirledik.

API	MEDIUM
Worker	MEDIUM
Callback	BASIC
Performans (DB, API, Worker) / Race Condition	HIGH
Raporlama	BONUS
Callback Event Queue Modülü	BONUS
Event Resend	BONUS
Modüllerin Dizayn Şeması	BONUS

Ödev ile ilgili herhangi bir sorun olduğunda bizimle ile iletişime geçmekten çekinme

Teşekkürler,

Kolay gelsin.