

GO Provider Migration Dikkat Edilmesi Gerekenler

Başlamadan Önce

1. Başlamadan önce provider'ın restrictionları konusunda emin olun. Gerekirse ilgili providerın entegrasyon dokümanına yeniden başvurun, olabildiğince az inisiyatif alın. Çünkü python geliştirmelerinde gözden kaçmış bazı noktalar olabilir. Doküman erişimi için Corepro production ekibinden yardım alabilirsiniz.

Restriction'lar :

- Multiple Destination Search Desteği (True or False)
- Tek transaction'da Max Hotel Search Count
- Tek transaction'da Max Booking Count
- Desteklenen Currency'ler
- Subquery'ler oluşturulurken pax gruplamaları (Groupby fonksiyonları)
- Her türlü search restriction'ı (Max adult per room, max children per room, total pax , total room sayısı, age çevirimleri, age restrictionları, booking window süreleri, maksimum konaklama süreleri vb.)

Entegrasyon Sırasında

1. Şu an Go projesinin sadece search kısmı çalışır vaziyette olduğundan sadece search endpoint'inin geliştirilmesi yeterlidir. Fakat geliştirme yaparken diğer endpointlerin de geliştirileceğini göz önüne alın ve generic yapıyı korumaya özen gösterin.
2. Geliştirme yaparken projede önceden geliştirilmiş providerlara göz atın ve ilgili kullanım ve konvansiyonları korumaya özen gösterin.
3. Generate Subqueries, Search, Sanitize Raw Request methodlarının geliştirilmesi yeterlidir.
4. Generate Subqueries methodunda olan çoğu restrictionlar Base Adaptör kısmında geliştirildi. Bunları kullanmak için geliştirme yapılan provider adaptöründe ilgili propertie lerin setlenmesi yeterli olacaktır. Ama provider özelindeki search restriction'ları yine provider adaptöründeki generate subqueries methodunda uygulamanız gerekmektedir.
5. Python projesinde kullanılan çoğu utility fonksiyonunu Go projesinde helpers modülünde geliştirmeye çalıştık. İhtiyaç duyduğunuz utility fonksiyonunu geliştirmeden helpers modülüne göz atabilirsiniz.
6. İlgili adaptörü kodlarken Go'nun kodlama konvansiyonlarını göz önünde bulundurun. Bunu takip etmek için builtin modül olan gofmt modülünü veya IDE'nin formatlama tool'unu kullanabilirsiniz. Ayrıca kodlamaya başlamadan gopro projesi bitbucket sayfasındaki pre-commit hookunu git environment'ınıza eklemeyi unutmayın. Bu sayede formata uymayan kodlar, commit atmanız engellenerek karşınıza gelecek.

Entegrasyondan Sonra

1. Genel olarak amacımız python projesinde elde ettiğimiz request ve response'ların aynısını Go projesinde yakalayabilmek üzerine. Fakat bunu yaparken 2 projenin response'larını detaylı olarak inceleyin.Örneğin 2 identical search requesti aynı şartlar altında aynı sayıda ve doğru bilgileri içeren ürünü veriyor mu bunu kontrol edin. Bu sırada python projesindeki geliştirmede de eksiklikler olabileceğini göz önüne alın ve gerekirse düzeltmeleri 2 adaptör için de uygulayın.
2. Öncelikli olarak 2 projedeki search raw request'lerini aynı search parametreleri için karşılaştırın ve aynı olduklarından emin olun.

3. Yaptığınız geliřtirmeyi kontrol ederken 2 projede de aynı search parametrelerini kullanarak, sonuçları ařağıdaki parametreleri dikkate alarak kontrol edin.
 - Single oda (Single çocuklu oda)
 - Multiple oda (Multiple çocuklu oda)
 - Subproduct sayıları
 - Subproduct ierikleri (**ref_code, cancellation policies, subroom bilgileri, price, room ve meal type, hotel , destination id**)
4. Son olarak Coral ekibinin geliřtirdiğı zen projesini sisteminize klonlayın. Coral ve zen üzerinden aynı 2 istek ile response'ları kontrol edin. Eğıer sonuçlar aynıysa (ürün sayısı ve ierikleri aısından) geliřtirmeyi tamamlamışsınız demektir.
5. Provider'ınız canlıya alındıktan sonra newrelic üzerindeki gopro-prod makinesi üzerinden ve sentry'yi kullanarak hatalarınızı trace etmeyi unutmayın.