

Proje - Remember Your Emojis

Bu projenin amacı, `Codable` protokolünü, `FileManager`'i, ve Data üzerinde metodlar kullanarak uygulamanız kapatıldıktan sonra bile girilen bilgilerin silinmemesini sağlamaktır. Bu projeye, daha önce geliştirdiğiniz "EmojiDictionary" projesi ile başlayacaksınız. "EmojiDictionary" uygulamasının kodunu hatırlamak için projeyi gözden geçirin. Uygulamayı çalıştırdığınızda, varsayılan bir emoji listesi ile başladığınızı, ancak kendi eklediğiniz emojielerin uygulamayı yeniden başlattığınızda silindiğini göreceksiniz. Bu projede, Emoji objelerinizin uygulama yeniden başlatılınca bile silinmemesini sağlayacaksınız.

Emoji Objelerine Kaydetme Ekleme

Adım 1: Uygulamanızdaki bilgilerin uygulama kapatılınca silinmemesi için, diskteki bir dosyaya bu bilgileri yazabilmeniz gerekir. İlk olarak, `Emoji` struct'ınızın tanımlamasında `Codable` protokolüne uymasını sağlayın.

Adım 2: Hatırlarsanız kaydetme ve yüklemeyi model üzerinde static metodlar olarak kodlayabilirsiniz. `Emoji` sınıfına `dosyayaKaydet(emojiIler:)` adlı static bir metodun tanımını ekleyin. Bu metod, parametre olarak bir `Emoji` objesi dizisi (Array) almalıdır. Şimdi static bir `dosyadanYukle()` metodu tanımlayın. Bu metod herhangi bir parametre almayıp bir `Emoji` objesi dizisi (Array) döndürmelidir.

Adım 3: Bu iki metodun her birinde, ya bir `PropertyListEncoder` ile `Emoji` objenizi encode edecek ya da `PropertyListDecoder` ile decode edeceksiniz. Ancak bir dosyaya bilgi yazmak için, bu dosyanın file path'ine (diskteki yerine) ihtiyacınız var. `Emoji` sınıfınıza, `Documents/emojis.plist` dosyasının file path'ini döndüren `ArchiveURL` adlı bir static property ekleyin.

Adım 4: Artık `Emoji` objelerinizi kaydedecek bir file path'iniz olduğuna göre, `dosyayaKaydet(emojiIler:)` ve `dosyadanYukle()` metodlarının gövdelerini doldurun ve file path olarak `Emoji.ArchiveURL`'i kullanın. `dosyayaKaydet(emojiIler:)` metodunuz, verilen emoji dizisini encode edip `Emoji.ArchiveURL`'e yazmalıdır, ve `dosyadanYukle()` metodunuz dosyadan bilgileri okuduktan sonra, bir `[Emoji]` dizisi olarak decode edip döndürmelidir.

Adım 5: Önceden tanımlanmış bir emoji dizisini (`[Emoji]`) yaratıp döndürecek olan static bir `ornekEmojiYukle()` metodu tanımlayın. Bu emoji dizisi için `EmojiTableViewCellController` dosyasında `emojis` değişkenindeki diziyi kullanabilirsiniz.

Kaydetme ve Yükleme

Adım 7: `EmojiTableviewController` dosyasında `emojis` değişkenini örnek bir dizi yerine boş bir dizi olarak tanımlanacak şekilde değiştirin. `viewDidLoad()` metodu çağrıldığında, `dosyadanYukle()` metodunu kullanarak `Documents/emojis` directory'sinde önceden kaydedilmiş emojiiler olup olmadığını kontrol edin. Eğer önceden kaydedilmiş emojiiler varsa, bunları `emojis` dizisine kaydededin. Yoksa, `emojis` değişkeninin değerini `ornekEmojiileriYukle()` metodunun sonucu olarak belirleyin.

Adım 8: `Emoji` objelerinizi kaydetmek için uygun zamanlar neler olabilir? Bir süre düşünün.

Adım 9: Bu durumda, bilginin depolandığı ana yer `EmojiTableviewController`'daki `emojis` dizisi--yani `emojis` değişkeni her değiştiğinde, `dosyayakaydet(emojiiler:)` metodunu çağırmak uygun olacaktır. Şimdi bunu uygulayın.

Adım 10: Peki, arşivlenmiş `Emoji` objelerini yüklemek için uygun zamanlar neler olabilir? Bu basit projede, bunu yapmamız gereken tek bir nokta var: view ilk olarak yüklendiğinde. Bunu zaten `viewDidLoad()` metodunun içerisinde yapıyor olmalısınız.

Adım 11: Artık, emoji data'nız doğru bir şekilde kaydedilip yükleniyor olmalı. Uygulamayı çalıştırın. Yeni bir emoji ekleyip Save butonuna tıklayın. Sonra uygulamayı kapatıp yeniden açın ve yeni eklediğiniz emojiinin Table View'da yer alıp almadığını gözlemleyin. Aynı deneyi, zaten var olan bir emojiyi düzenlemek için yapın.

Tebrikler! Bu projeyi başarılı bir şekilde bitirdiniz! 🎉