**Quiz App Documentation**

Version: 1.0

Tarih : 8 Ocak 2019

Açıklama : Proje Bitişi

**Kütüphaneler**

*Kütüphaneleri proje pod kullanarak entegre ettim.*

SwiftyJSON -> Json üzerinde daha kolayca işlem yapabilmek için bu kütüphaneyi kullandım.

Socket.IO-Client-Swift -> Socket dinleme işlemleri için kullanılır.

**Design Pattern**

Model View Presenter (MVP) kullandım. Kullanımı basit olduğu ve mantıksal ayrımı sağladığı için bunu kullandım. Kısacası herkesin kendi işini yaptığı bir yapı olduğu bunu kullanmayı seçtim.

**Sınıflar**

***QuestionModel.swift***

* Apiden dönen question’ı karşılayan Model sınıfı
* Daha sonra QuestionViewData sınıfına dönüştürülüyor
* **id** **->** row id
* **question** **->** sorunun olduğu alan
* **opt1 ->** 1.seçenek
* **opt2 ->** 2.seçenek
* **opt3 ->** 3.seçnek
* **opt4 ->** 4.seçenek
* **correctIndex ->** Doğru cevabın indexi

***UserModel.swift***

* Apiden dönen user’ı karşılayan Model sınıfı
* Daha sonra UserViewData sınıfına dönüştürülüyor
* **id->** row id
* **name ->** kullanıcının adı
* **joker ->** kullanıcının joker sayısı

***QuestionPresenter.swift***

* Bu sınıf, verileri servisten alıp viewa gönderir.
* **getQuestion(id: Int) ->** apiye question getirme isteği atmak için aracı görevi görüyor, gelen datayı ekrana basmak için view’a gönderiyor.
* **func checkCorrectQuestion(questionIndex: Int, selectedIndex: Int) ->** Cevabının doğruluğunu kontrol etmek için service sınıfına gönderiyor, dönene cevaba göre viewda ilgili methodu tetikliyor.
* **func newQuestion() ->** yeni soru üretmek
* **func getUser(id: Int) ->** Serverdan user çekmek için service sınıfına gidiyor, dönen cevabı view’a gönderiyor.
* **func checkUserJoker(user: UserViewData) ->**  Kullanının joker sayısını kontrol etmek için işlemleri gerçekleştiriyor.
* **updateUserJoker(...) ->**  Kullanının joker kullandığında sunucudaki joker sayısını update etmek için servis sınıfına haber veriyor ve dönen cevabı view’a gönderiyor.

***QuestionService.swift***

* Question api servisi
* **func getQuestion(id: Int ... ) ->** Belirtilen ID değerine göre sunucudan soru çeker. ID verilmesinin sebebi sorunların tek tek çekilecek olması, bu yüzden uygulama açılışında bütün ID’ler okunup daha sonra ID’ye göre soru çekilir.
* **func checkCorrectQuestion(...) ->**  Sorunun doğruluğunu servera sorarak kontrol eder.
* **func getUser(id: Int …) ->** Belirtilen ID’ye göre sunucudan kullanıcı çekilir. yerine username/password kullanılacaktır.
* **func checkUserJoker(id: Int…) ->** Kullanıcının jokerinin olup olmadığı sunucuya sorar.
* **func updateUserJoker(id: Int…) ->** Kullanıcını sunucudaki jokerini günceller.