**GAZİ ÜNİVERSİTESİ TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ**



**Yazılım Proje Yönetimi**

Test Planı

171816061 – Hakan Emre BAŞOL

171816033 – Bilal Nadir KARATAŞ

171816754 – Oğuzhan EYERCİ

171816008 – Ömer Fatih CANBOLAT

**MAYIS 2020**

**İçindekiler**

[1. ANALİZ 3](#_Toc40659814)

[1.1. Kimler Kullanacak 3](#_Toc40659815)

[1.2. Ne için Kullanılacak 3](#_Toc40659816)

[1.3. Yazılım Donanım Gereksinimleri 3](#_Toc40659817)

[1.4. Test Hedeflerini Tanımla 3](#_Toc40659818)

[1.4.1. Test ölçütleri 3](#_Toc40659819)

[1.5. KAYNAK PLANLAMASI 3](#_Toc40659820)

[1.6. Test Ortamını Planla 4](#_Toc40659821)

[1.7. Zamanlama ve Tahmin 4](#_Toc40659822)

[1.8. Test Çıktılarını Belirle 4](#_Toc40659823)

[2. Birim Testleri 5](#_Toc40659824)

[2.1. Test Edilecek Classlar 5](#_Toc40659825)

[2.1.1. HttpClient.java 5](#_Toc40659826)

[2.1.2. KeyboardController.java 7](#_Toc40659827)

[2.1.3. PreferencesManager.java 10](#_Toc40659828)

[2.1.4. MyTextToSpeech.java 13](#_Toc40659829)

[2.2. Testler 14](#_Toc40659830)

[2.2.1. HttpClient Test 14](#_Toc40659831)

[2.2.2. KeyboardController Test 16](#_Toc40659832)

[2.2.3. PreferencesManager Test 18](#_Toc40659833)

[2.2.4. TextToSpeech Test 19](#_Toc40659834)

[2.3. Test Sonuçları 20](#_Toc40659835)

[3. Performans Testleri 22](#_Toc40659836)

[3.1. CPU Testi 22](#_Toc40659837)

[3.2. Network Testi 22](#_Toc40659838)

[3.3. RAM Testi 23](#_Toc40659839)

[4. API Testi 24](#_Toc40659840)

[4.1. Skor Get Testi 24](#_Toc40659841)

[4.2. Skor Id ile Get Testi 24](#_Toc40659842)

[4.3. Kelime Get Testi 25](#_Toc40659843)

[4.4. Kelime id ile Get Testi 25](#_Toc40659844)

[4.5. Kelime Post/ Insert Testi 26](#_Toc40659845)

[4.6. Skor Post/Insert Testi 26](#_Toc40659846)

[4.7. Kelime Post/Update Testi 27](#_Toc40659847)

[4.8. Skor Post/Update Testi 28](#_Toc40659848)

[5. Stres Testleri: 30](#_Toc40659849)

# **ANALİZ**

## Kimler Kullanacak

Braille Keyboard, braille alfabesini öğrenmek isteyen bireyler, görme engelliler okulunda öğrenim gören öğrenciler ve görme yetisini tamamen kaybetmiş bireyler için tasarlanmış bir uygulamadır.

## Ne için Kullanılacak

Braille alfabesini öğrenmek isteyen herkes için bir öğretici uygulama olacaktır. Görme engelliler okulunda öğretici ders materyali olarak kullanılabilir. Braille alfabesini daha kolay ve hızlı yoldan öğrenmek, öğrendiklerini uygulama ile pekiştirmek ve eğlenmek isteyenler için bir uygulamadır.

## Yazılım Donanım Gereksinimleri

* Android tabanlı mobil cihaz,
* Yazılım dili için Java,
* Yazılım ortamı için Android Studio,
* İnternet.

## Test Hedeflerini Tanımla

### Test ölçütleri

## KAYNAK PLANLAMASI

|  |  |  |
| --- | --- | --- |
|  | Kaynaklar | Açıklama |
| 1 | Bilgisayar | 2 adet Windows işletim sistemli bilgisayar |
| 2 | Yazılım ekibi | 4 kişilik Yazılım ekibi ( 2 Coder, 1 Tester, 1 Manager) |
| 3 | Android telefon | 2 adet Android tabanlı telefon |
| 4 | Android Studio | Windows sistemler için yüklenecek program |
| 5 | Rxjava | Uzak sunuculara bağlanmak için çok fonksiyonlu |
| 6 | Retrofit | Uzak sunuculara bağlanmak için kullanılacak |
| 7 | Butter Knife | View bileşenlerini tanımlamak için kullanılacak |
| 8 | Gson | Java objelerini json formata çevirmek için kullanılacak |
| 9 | C# | Web Api için kullanılacak |
| 10 | MVC | .Net ' in MVC yapısı tercih edilecek |
| 11 | MsSQL | Veri tabanı olarak kullanılacak |
| 12 | Windows Sunucu | Sunucu işleminde kullanılacak |

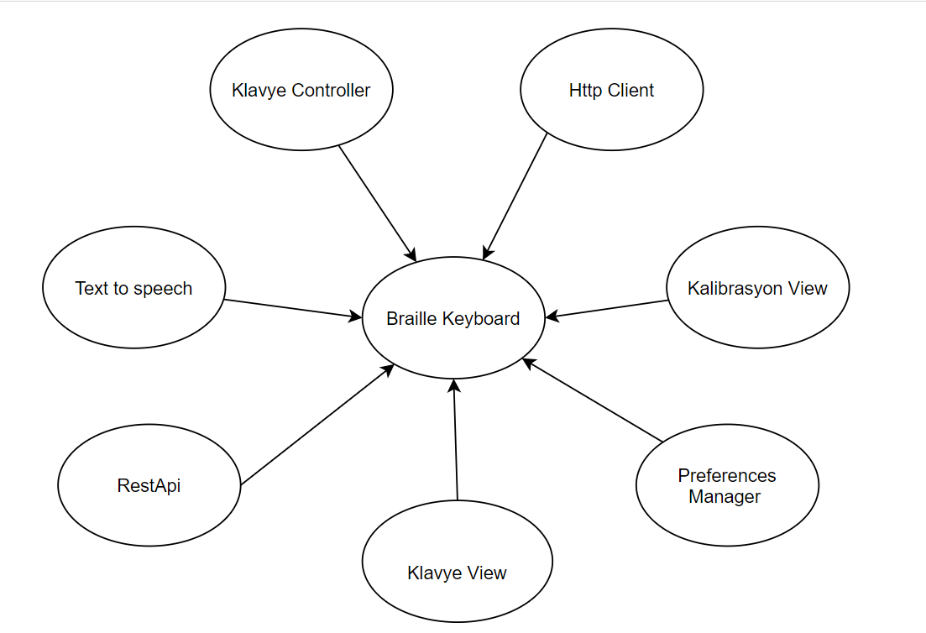
## Test Ortamını Planla

## Zamanlama ve Tahmin

## Test Çıktılarını Belirle

# Birim Testleri

Test işlemlerini kolaylaştırmak için modüler çalışma sağlanmıştır. Modüller üzerinde ayrı ayrı testler gerçekleştirilmiştir.



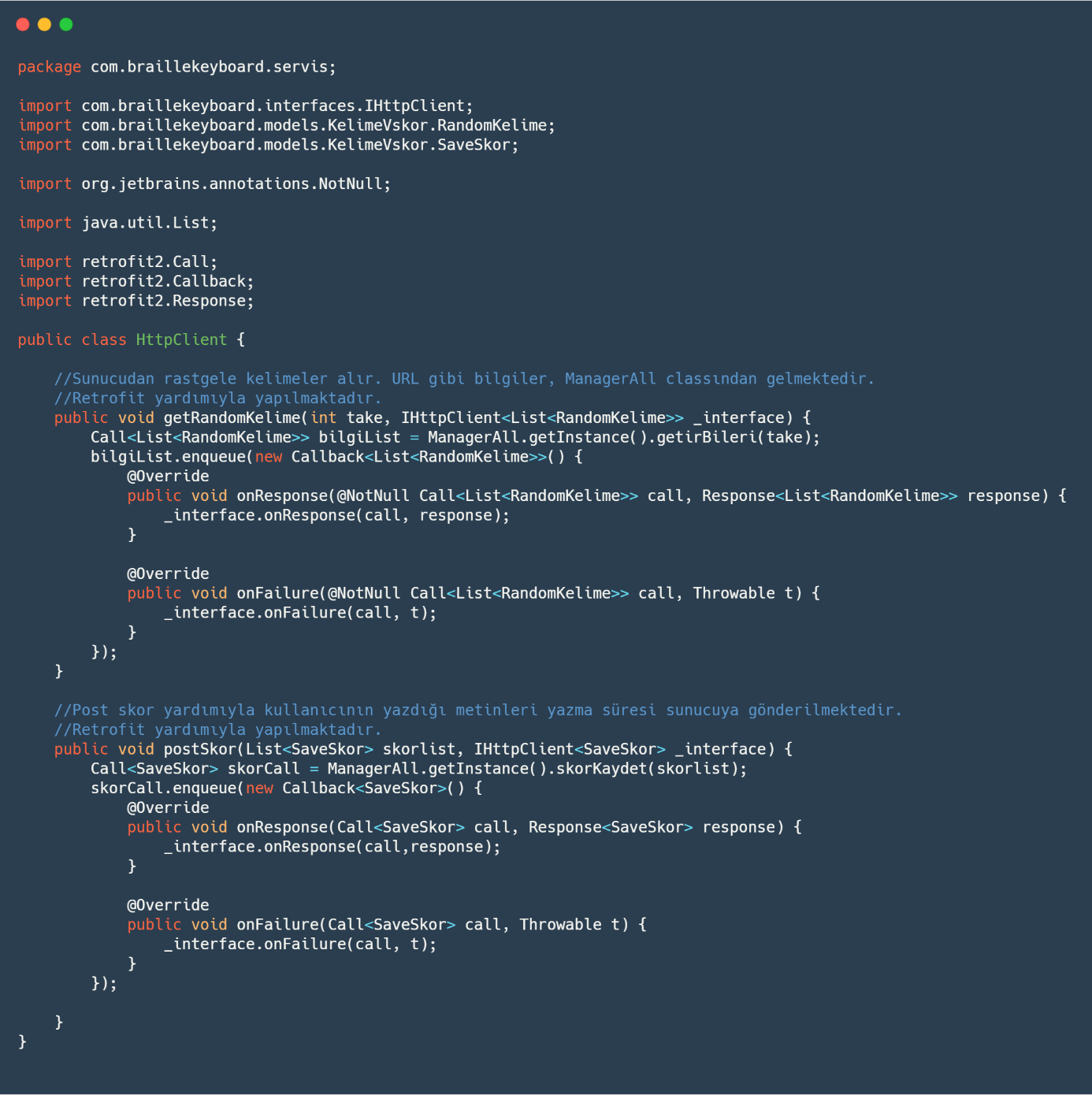
## Test Edilecek Classlar

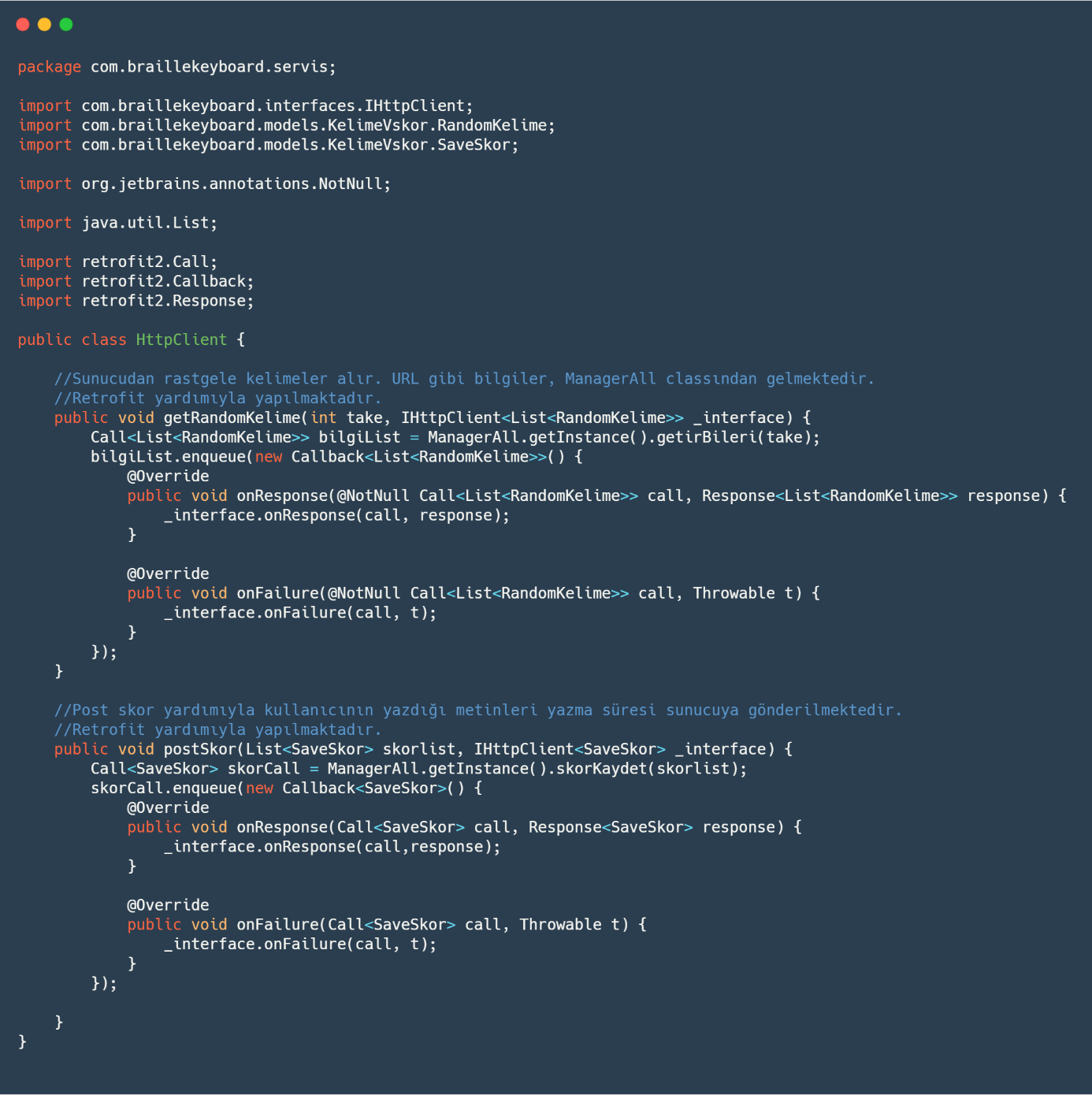
### HttpClient.java

Http client yardımıyla sunucu (API) ile uygulama arasındaki iletişim sağlanmaktadır. Temel olarak 2 adet GET ve POST metodu vardır.

Yaptığı işlem temel olarak belirtildiği gibidir.

**GET / https://localhost/api/kelime/getrandom?take=5**

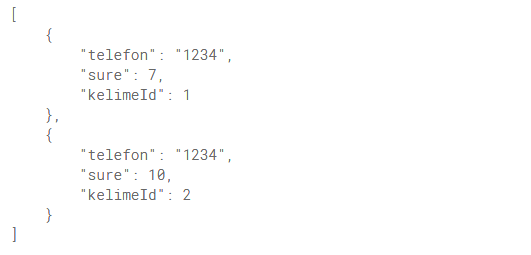




Post yardımıyla zamana karşı yarışta kullanıcının yazdığı metinlerin süreleri API’a post edilmektedir. Bu sayede profil kısmında bir sıralama oluşturulabilmektedir. Yaptığı işlem belirtildiği gibidir.

**POST /** [**https://localhost/api/skor/saveskorList**](https://localhost/api/skor/saveskorList)

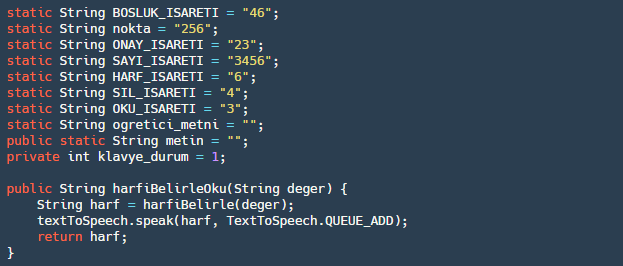
Json formatında gönderilmektedir.



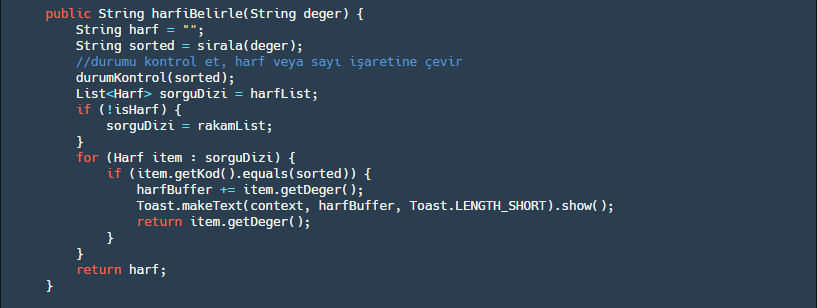
### KeyboardController.java

Keyboard controller yardımıyla klavye üzerinde basılan noktaların hangi harfe denk geldiği tespit edilmektedir. Bu controller içerisinde tüm karakter, harf ve sayılar tanımlanmıştır. Text to speech ile entegre çalışmaktadır.





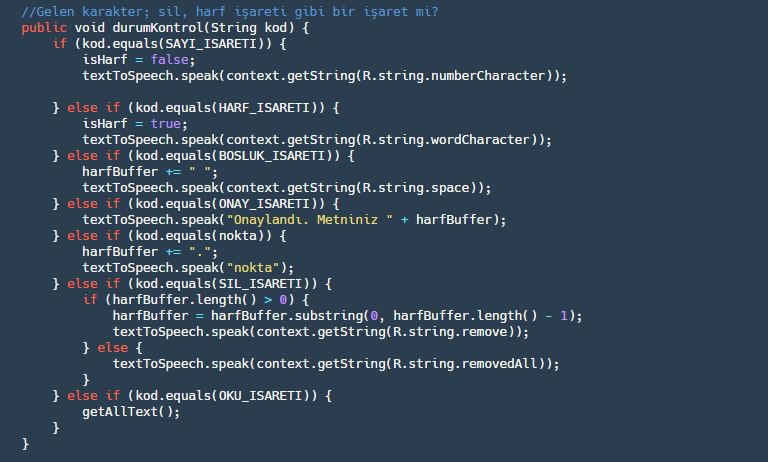
harfiBelirle() metodu yardımıyla gelen değer class başında tanımlanan diziler üzerinde sorgulanır ve gelen harf,sayı tespit edilir.



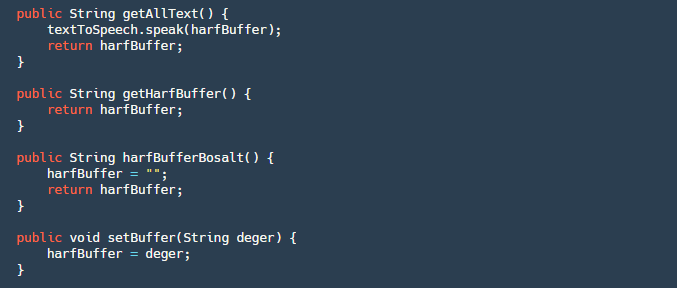
Klavye üzerinde basılan değerler sırasıyla basılmamış olabilir (1234 yerine 1342 gibi). Bu durumda karakter tespiti için sıralama işlemi yapılmak zorundadır.



Karakterler, sayı veya harf olacağı gibi özel karakter de olabilir. (Sayı işareti, harf işareti, yazıyı sil vb.) bu durumda bunların tespit edilmesi ve duruma göre işlem gerçekleştirme işlemi de durumKontrol yardımıyla kolayca yapılabilmektedir.

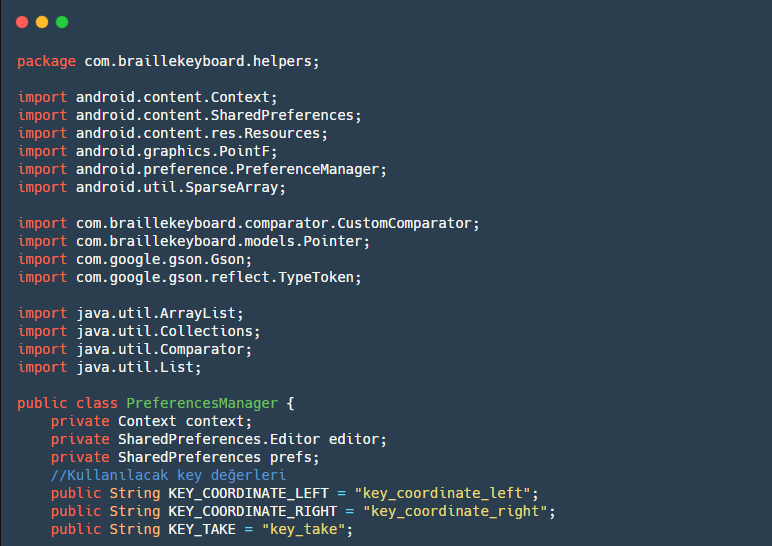


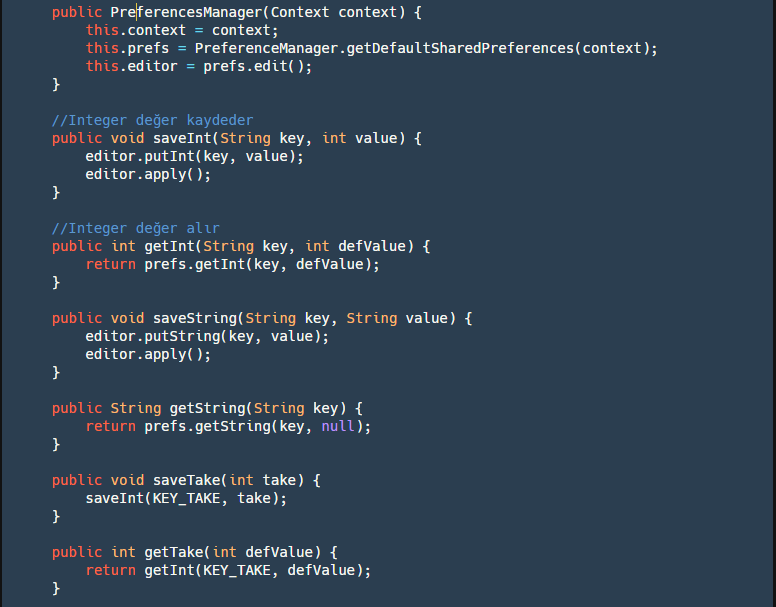
Controller üzerinde çeşitli yardımcı fonksiyonlar bulunmaktadır. Bunlar yardımıyla harfBuffer(şimdiye kadar yazılan tüm metin), metin silme gibi yardımcı işler gerçekleştirilebilmektedir.

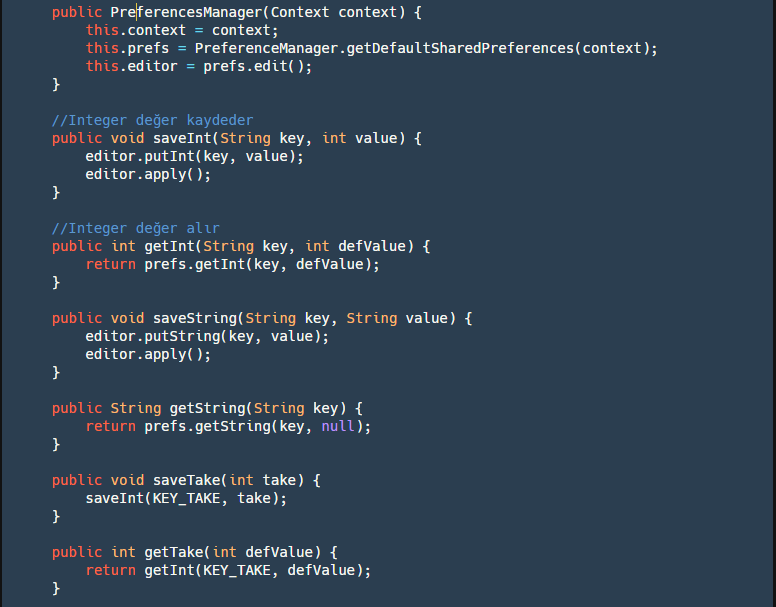


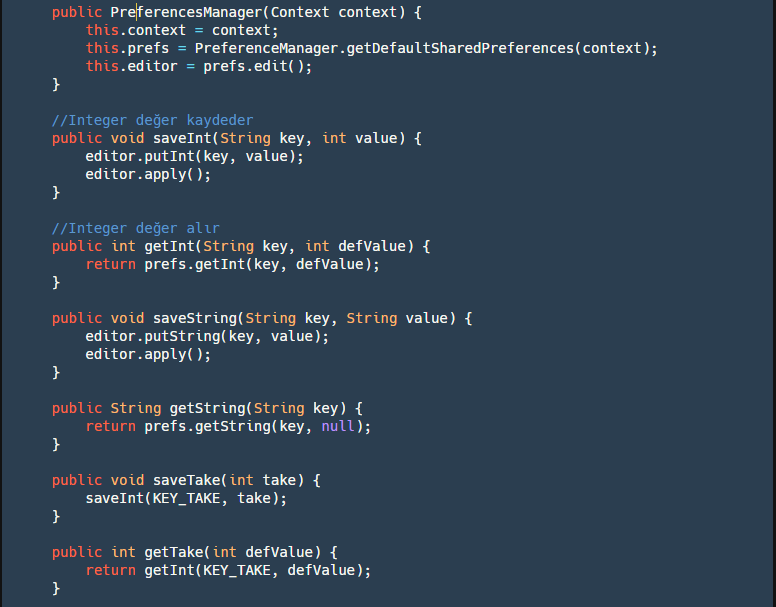
### PreferencesManager.java

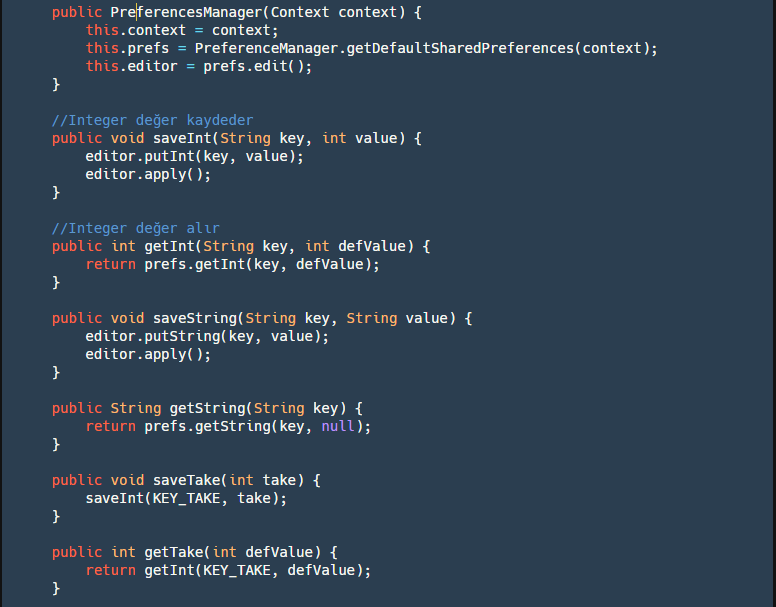
Preferences Android ekosistemlerinde kullanılan XML tipinde bir veri kaydetme yöneticisidir. Bu yönetici türetilerek içerisine yardımcı fonksiyonlar yazılmıştır. Klavye kalibrasyonu, çeşitli ayarlar bu manager yardımıyla saklanmaktadır. Her tipte değişken için yardımcı fonksiyon (getter,setter) yazılmıştır (int,string vb.).

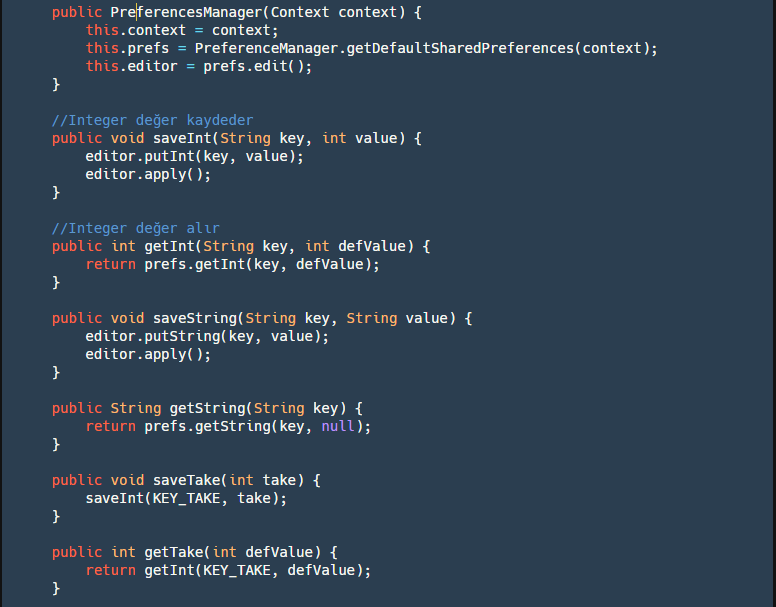


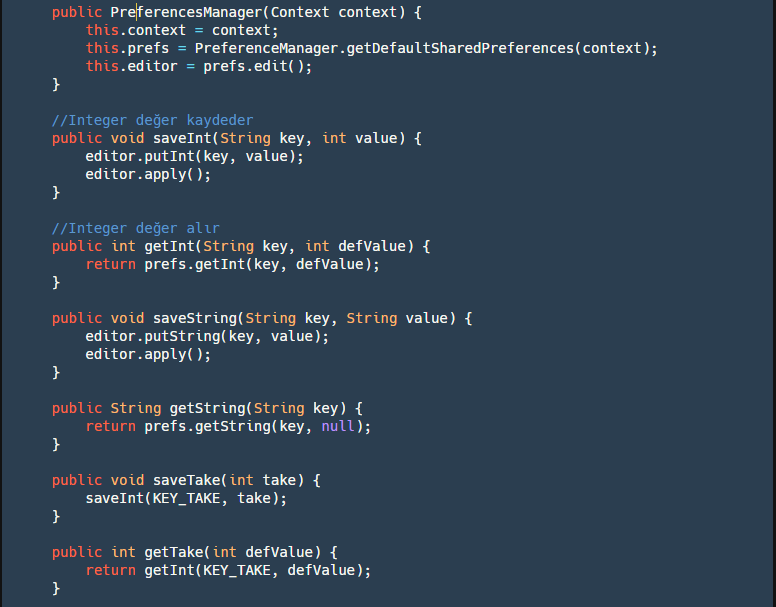


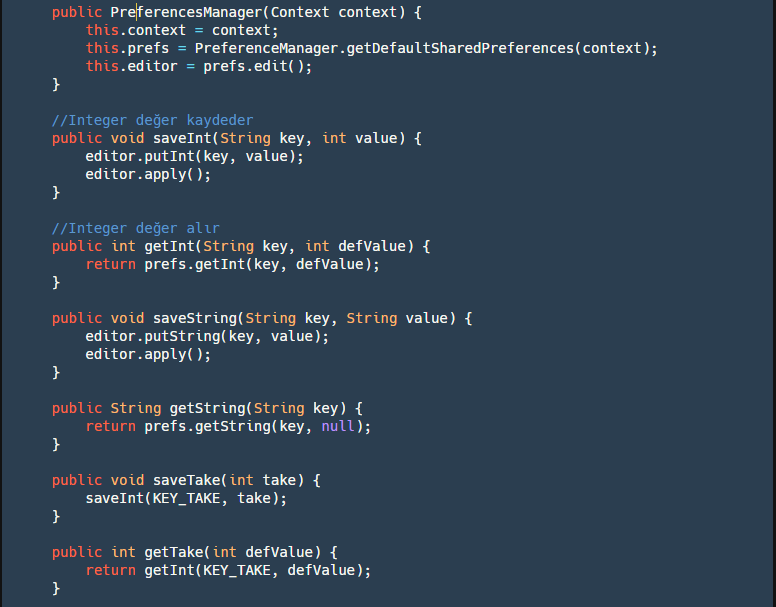


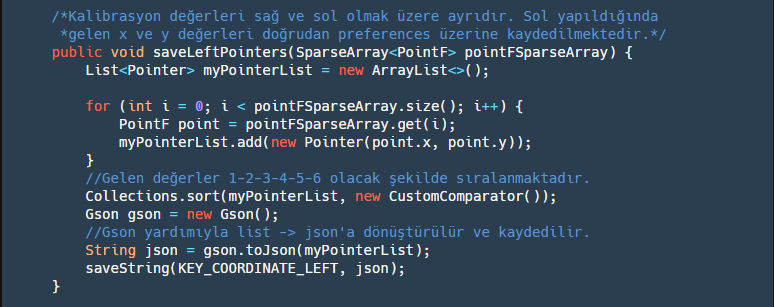




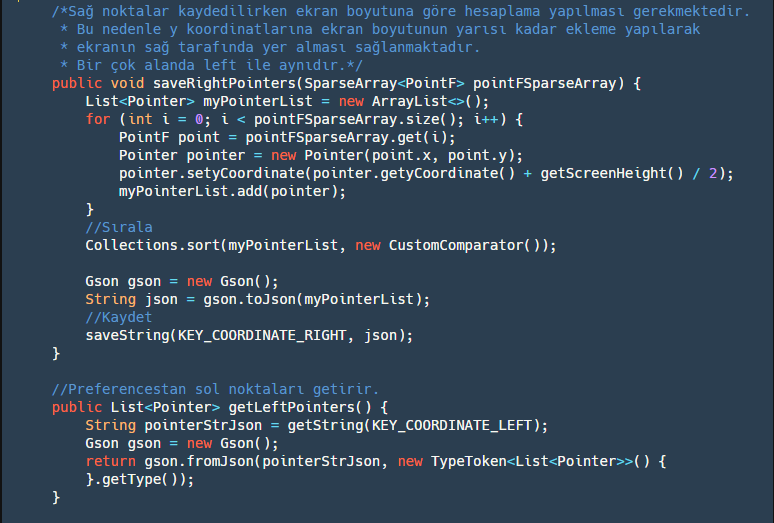




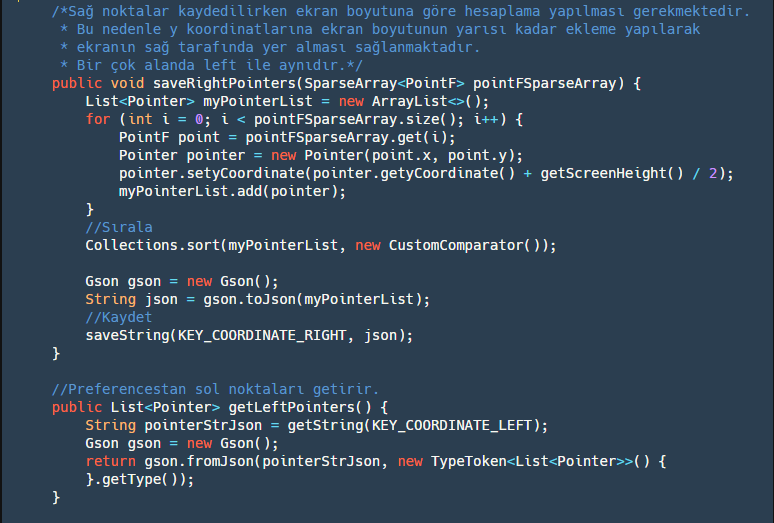


Kalibrasyon yapıldıktan sonra sağ ve sol kısımlar ayrı ayrı kaydedilmektedir. Bunun en temel nedeni sağ kısmın sağ tarafta bulunabilmesi için ona ekranın yarısı kadar koordinat eklemesi yapılması gerektiğindendir. Sol taraf için bir fark bulunmamaktadır ancak bu gelen 6 nokta bir adet custom sorter ile sıralanmaktadır. (Noktaların yukarıdan aşağı 1-2-3-4-5-6 olması için.)

Sağ noktalar da sol noktalar gibidir ancak belirtildiği üzere ekranın boyutunun yarısı kadar y eksenine ekleme yapılmak zorundadır.



Kalibrasyon yapıldıktan sonra getter fonksiyonları yardımıyla klavye noktalarının koordinatları alınabilmektedir.

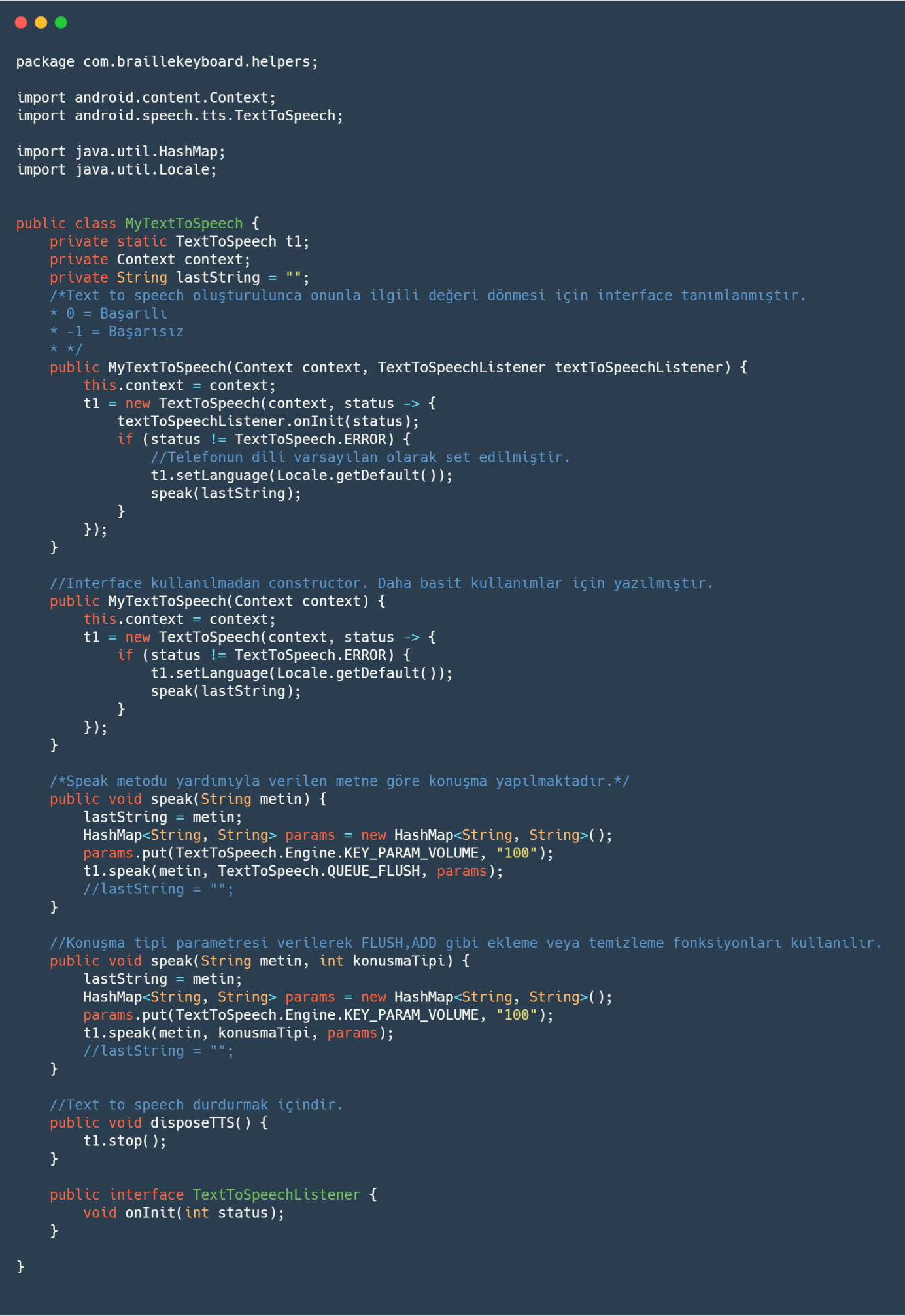
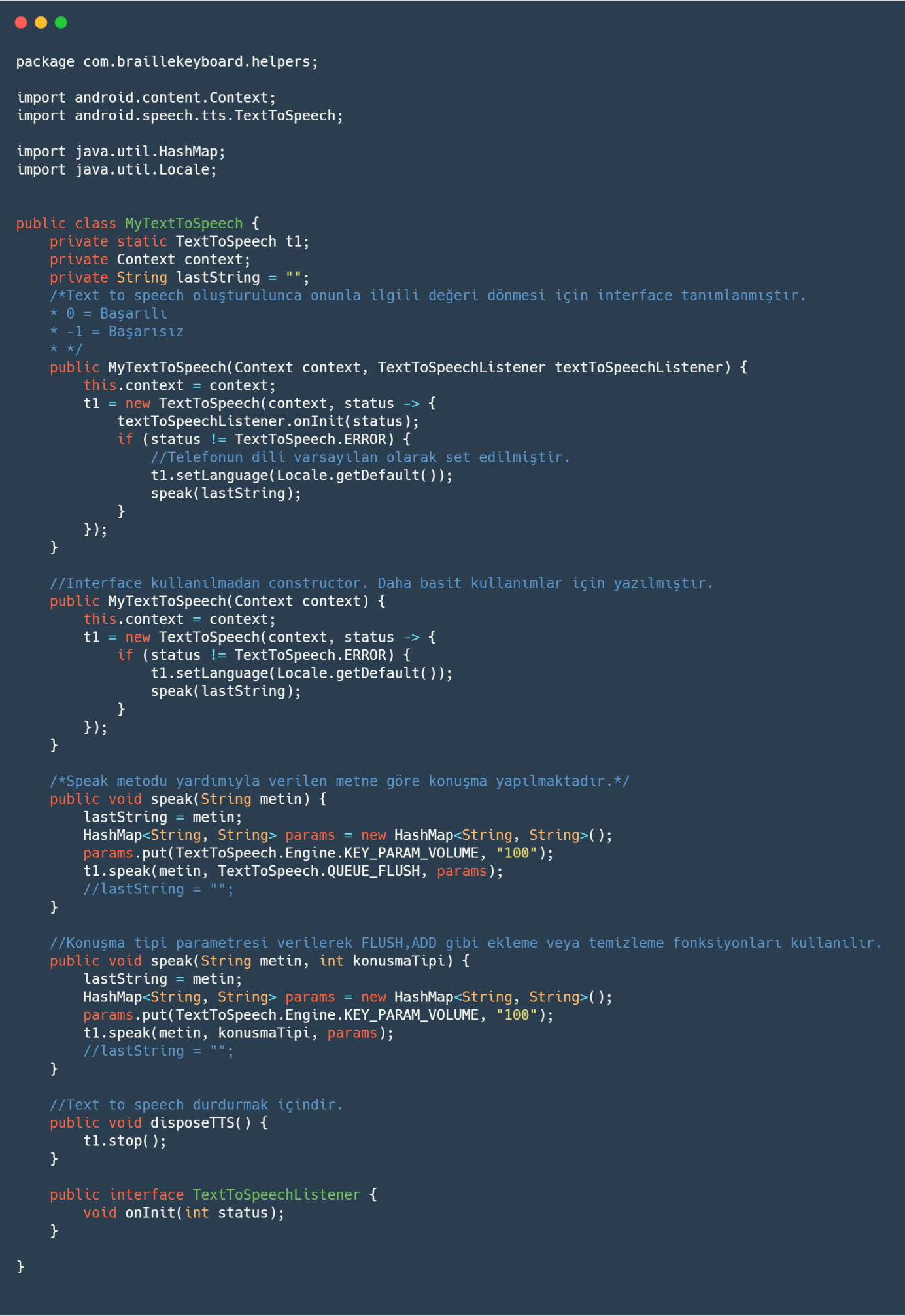
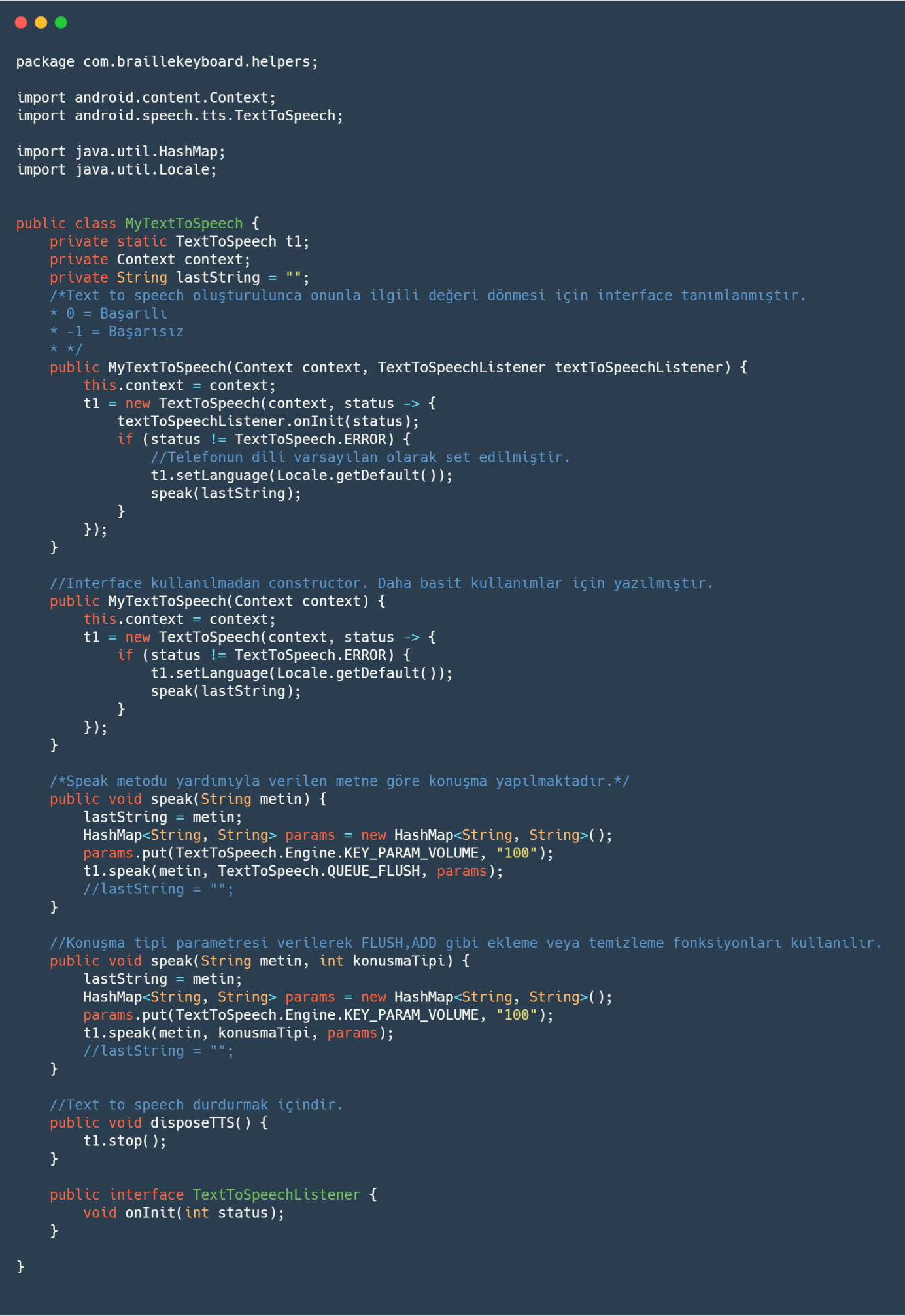


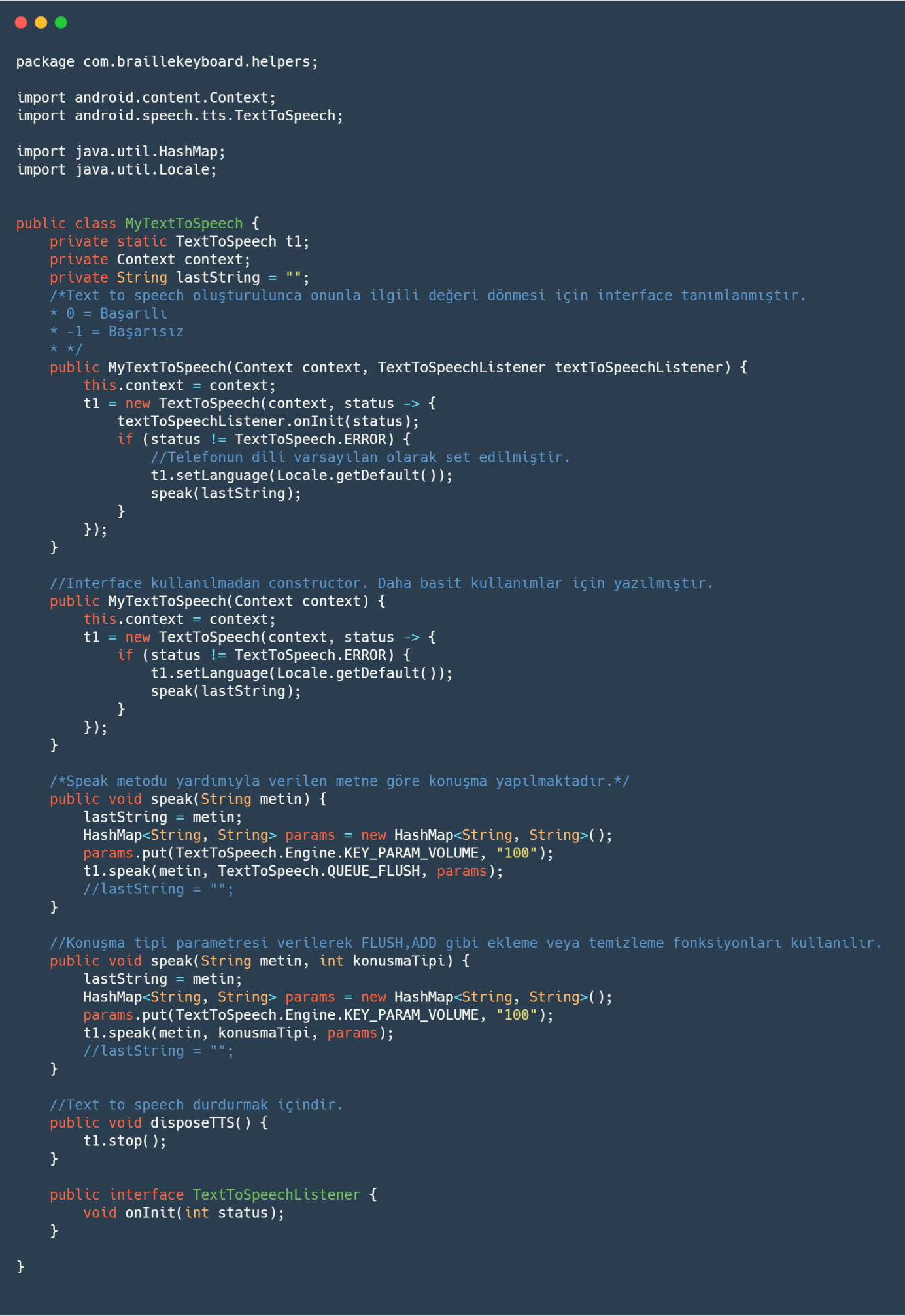
Pointerlara yardımcı olması için height getter’ı yazılmıştır.



### MyTextToSpeech.java

Google tarafından sunulan text to speech kütüphanesi için yardımcı bir class yazılmıştır. Bu sayede merkezi olarak tüm konuşma işlemleri yönetilebilmektedir. Fonksiyonların işlevleri açıklama satırlarıyla belirtilmiştir.

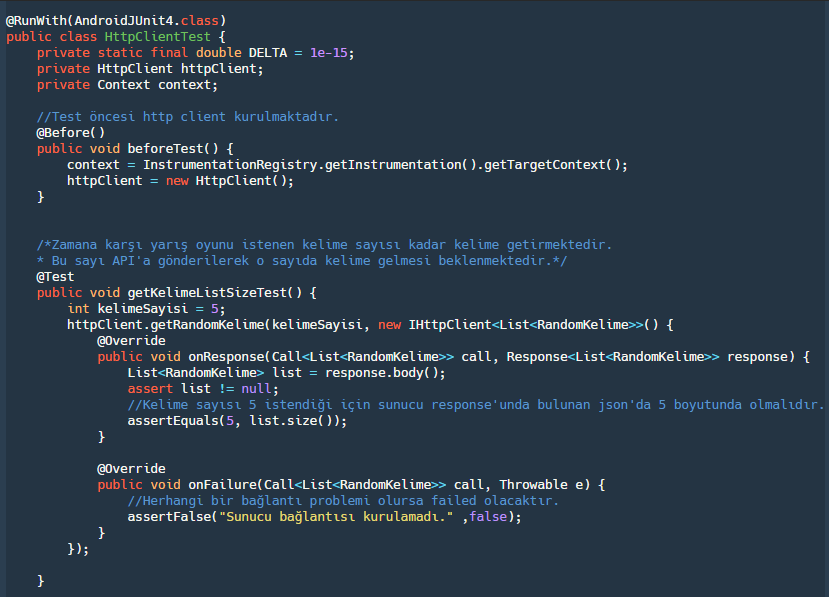




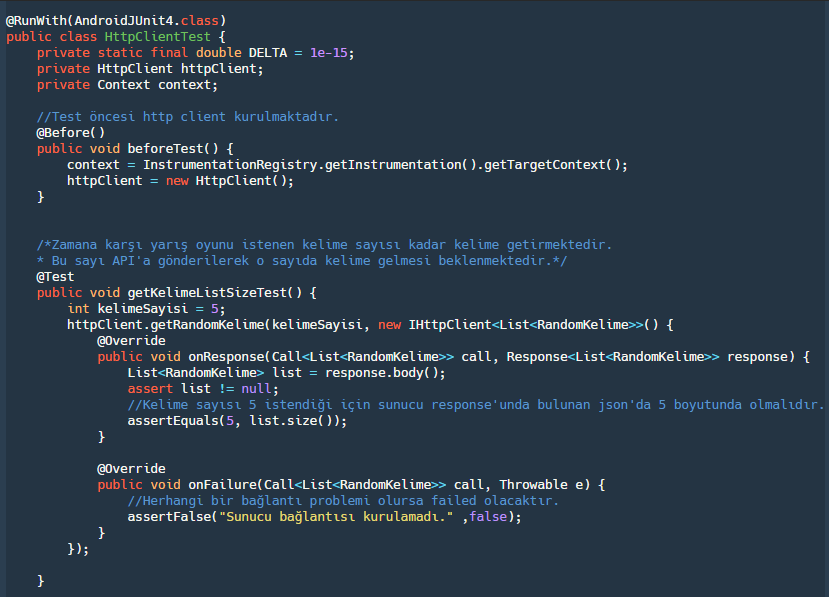
## Testler

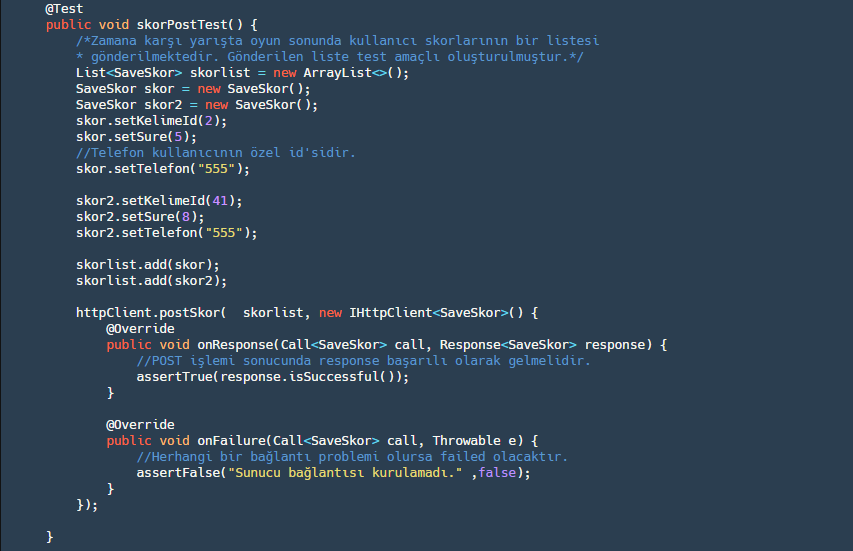
### HttpClient Test

Before test yardımıyla context ve httpClient classımın türetilmiştir.



Global olarak atanmış referanslar yardımıyla kelimenin istenilen sayıda gelip gelmediği kontrol edilmiştir. Uygulama yapısında zamana karşı yarış kısmında kullanıcının istediği kadar kelime sunucudan rastgele talep edilmektedir. Test içeriğinde 5 olarak belirlenen kelime sayısı sunucudan gelen kelime sayısı ile karşılaştırılmıştır. Bu sayede istenilen sayıda gelip gelmediği kolaylıkla anlaşılamabilmektedir.

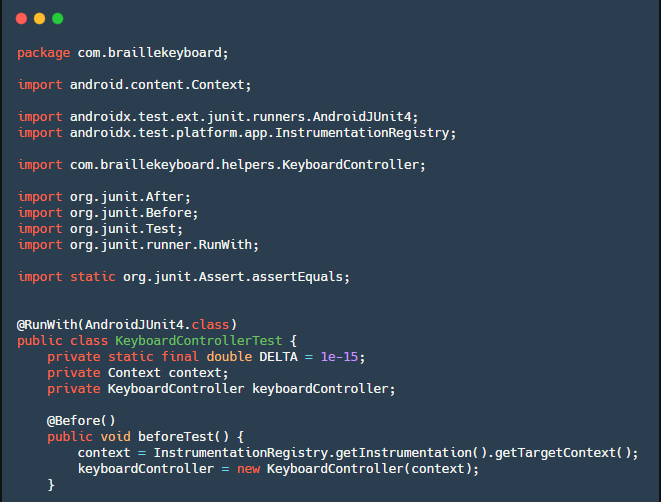


Http Client’ın ikinci bir testi ise zaman karşı yarış sonucunda elde edilen skorların sunucuya gönderilmesidir. Skorlar çoklu olacağı için list şeklinde post edilmelidir. Eğer sucunuya başarılı şekilde ulaşmış ve başarılı bir response dönmüş ise test başarılı olarak sonuçlanacaktır.

Test işlemi sonuçlandığında httpClient null olarak set edilir.

### KeyboardController Test

Belirtildiği üzere keyboard controller tüm klavye işlemlerinin yapıldığı classtır. Test öncesinde klavye controller ve context global değişkene atanmaktadır.



Uygulama içerisinde; harf, sayı gibi farklı elementler vardır. Bu nedenle bu elementlerin testleri ayrı ayrı yapılmıştır. Örneğin, 1235 numaralarına basıltığında R harfi olarak çıktı vermelidir. Basit bir şekilde beklenen değer girilerek test gerçekleştirilmiştir.



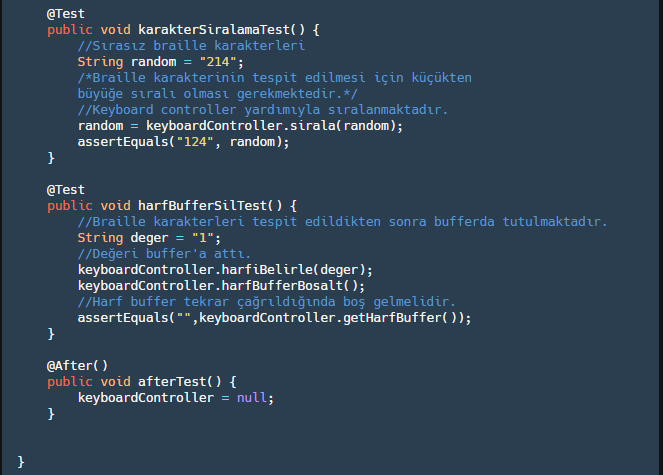
Klavye üzerinde sayı yazılabilmesi için sayı işareti kullanılmalıdır. Sayı işareti 3456’dır ve günlük hayatta kullanılan Braille Alfabesinde de o şekilde belirtilmiştir. Klavye controllera gönderilen sayı işaretinin artından 145 yani 4’e denk gelen numaralar gönderilmiştir. Test sonucunda beklenilen değer 4 olarak girilmiştir ve test gerçekleştirilmiştir.



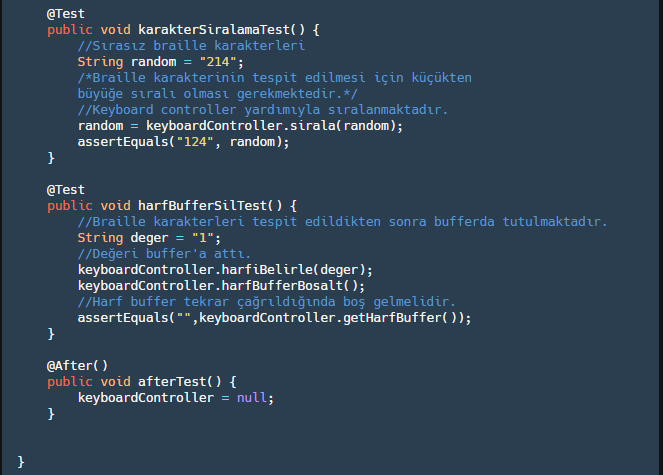
Farklı elementleri birleştirmesi açısından tek karakter yerine dizi yardımıyla bir metin testi yapılmıştır. Dizi içerisinde girilen değerler sırasıyla O,Ğ,U,Z harflerine denk gelmektedir. Bu sayede beklenen değer girilerek test gerçekleştirilmiştir.



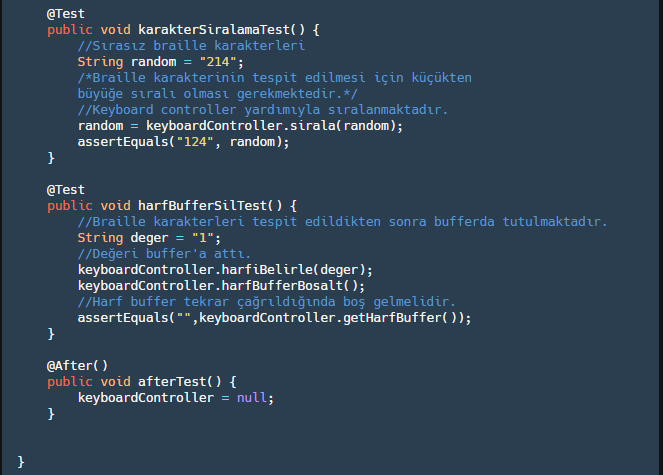
Klavye controller içerisinde rastgele gelen numaralar sıralanmaktadır. Bu işlem için de basit bir test yazılmıştır.



Harf buffer üzerinde yazılan tüm karakterler tutulmaktadır. Silme işlemi sonucunda içeriği boş kalmalıdır.

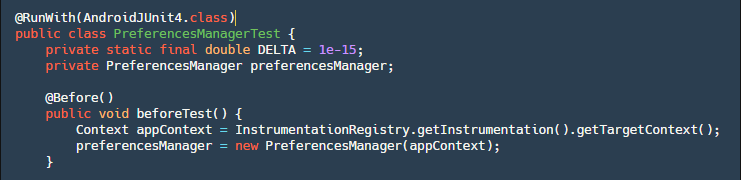


Test bitiminde keyboardController null olarak atanmaktadır.

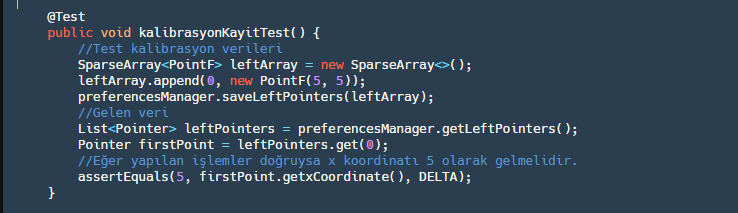


### PreferencesManager Test

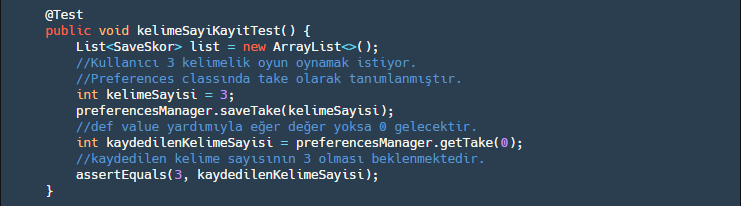
Test öncesinde context ve preferences manager classı oluşturulmuştur.



Kalibrasyon kaydı için sol kısım seçilmiştir. Bunun için deneme amaçlı x=5 ve y=5 değerleri gönderilmiştir. Gönderilen değerler 1 noktasını temsil etmektedir. Bu sayede kayıt sonucunda kaydedilen veri tekrar alındığında gelen x değeri de 5 olmalıdır.

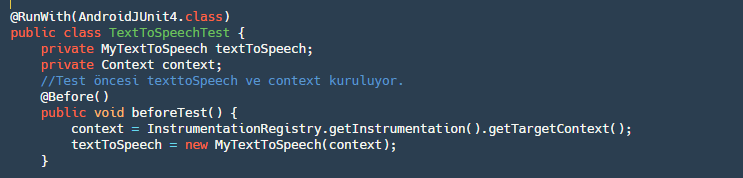


Ayarlar kısmından kullanıcının gelmesini istediği kelime sayısı girilip, kaydedilmektedir. Örnek olarak 3 değeri seçilmiştir ve gelen değer de bu beklentiye göre test edilmitir.

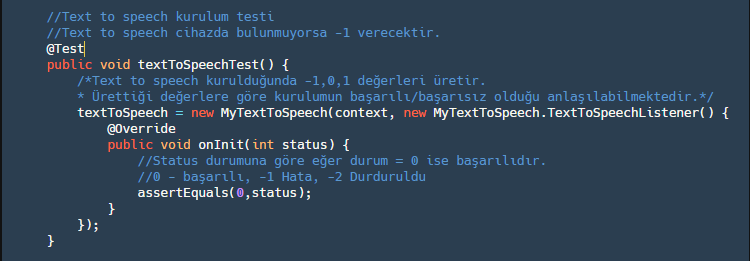


### TextToSpeech Test

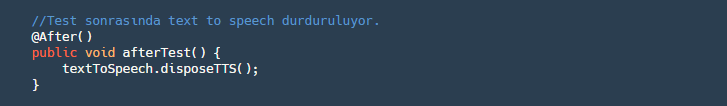
Text to Speech testi için çok fazla imkan bulunmamaktadır ancak Text to Speech’in başarılı bir şekilde kurulup kurulamadığı kontrol edilebilmektedir. Test öncesinde @Before() yardımıyla class oluşturulmuştur.



Bir interface yardımıyla oluşturulan class içerisindeki durum kodu alınmıştır. Alınan durum kodu = 0 olursa kurulum başarılı bir şekilde yapılmış demektir.

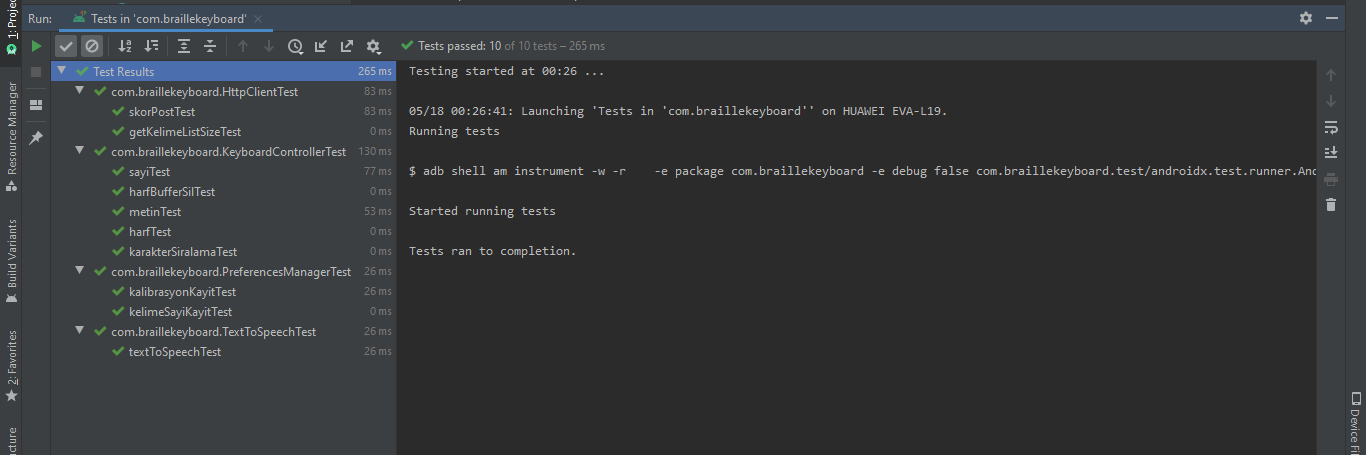


@After kısmında test artık sonuçlanmış ve Text to Speech durdurulmuştur.



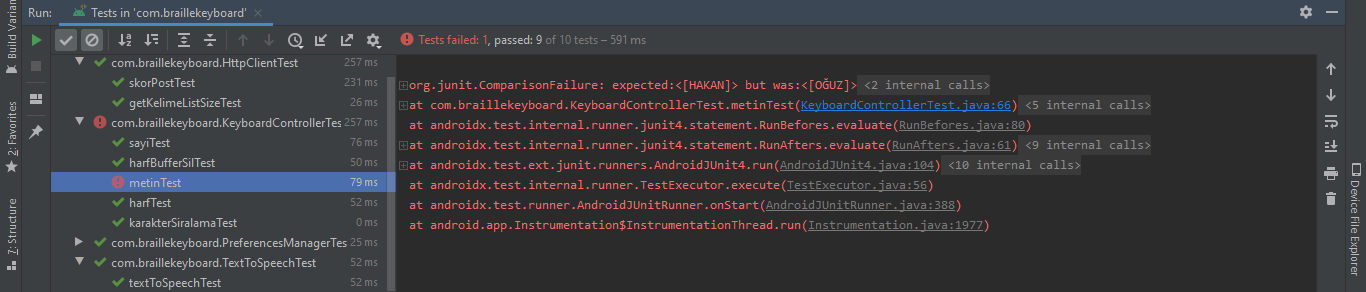
## Test Sonuçları

Belirtilen değerler girildiği takdirde tüm testler başarıyla sonuçlanmaktadır. Toplamda 10 test yapılmıştır.

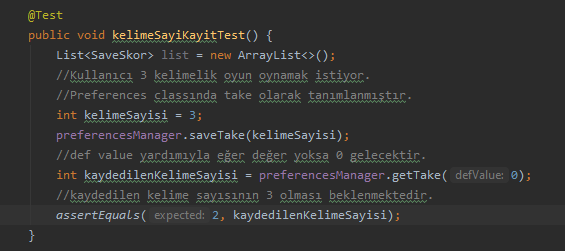


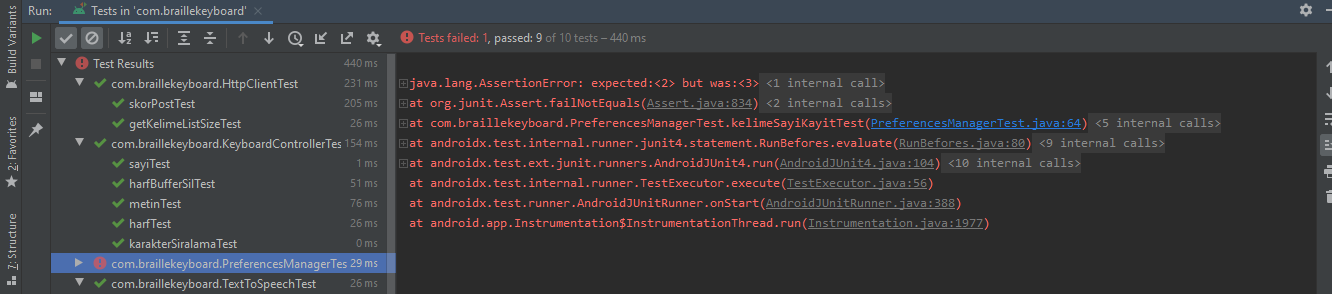
Kasıtlı yapılan hatalı sonuç ile birlikte yanlış test sonuçları da üretilmiştir. Örneğin, metinTest kısmında gelmesi gereken metin **OĞUZ olduğu halde HAKAN girilmiştir**. Bunun sonucunda test başarısızlıkla sonuçlanmıştır.





Bir diğer kasıtlı yapılan başarısız test ise kelime sayısı kısmındadır. Senaryoya göre kullanıcı, ayarlardan 3 kelimelik bir yarış seçmiştir ancak preferencestan dönen değerin 2 olması beklenmiştir. Bunun sonucunda test başarısızlıkla sonuçlanmıştır.



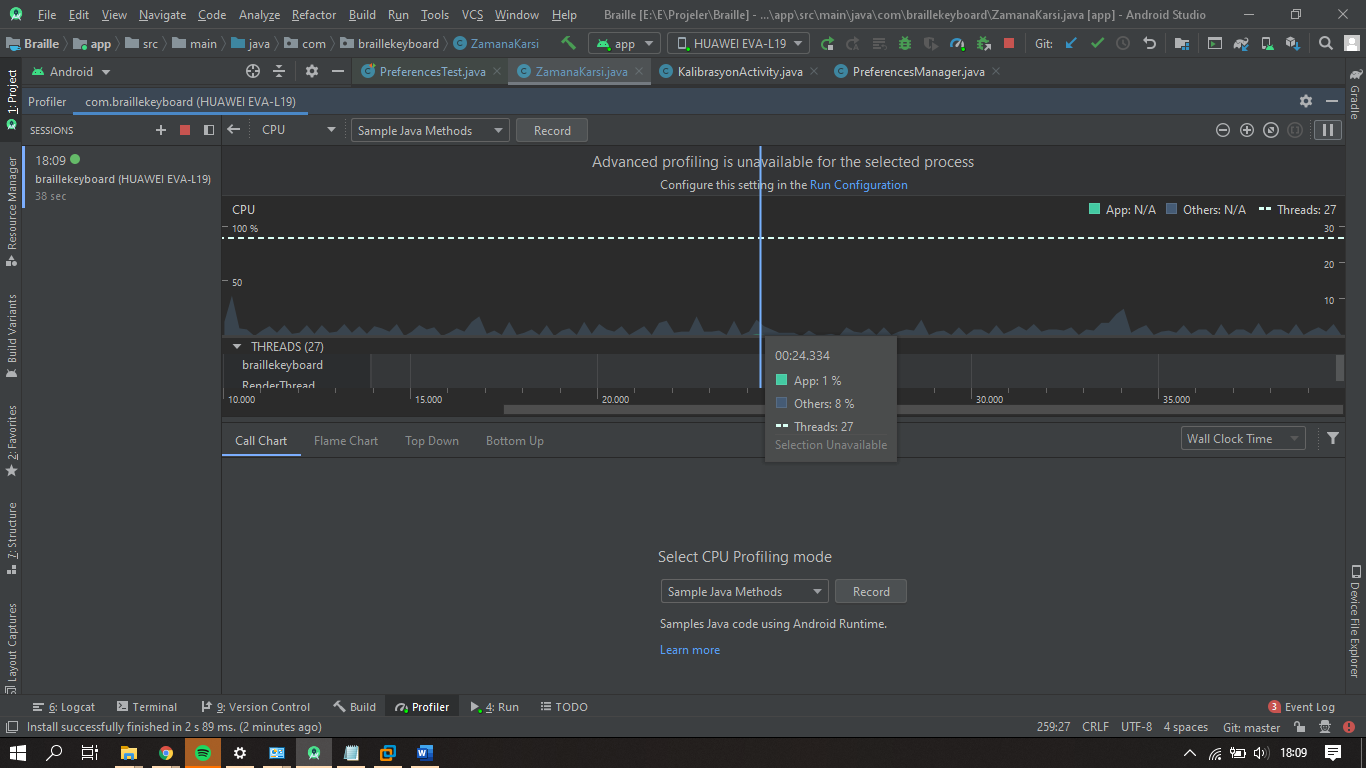


# Performans Testleri

Performans tesleri Android Profiler yardımıyla gerçekleştirilmiştir.

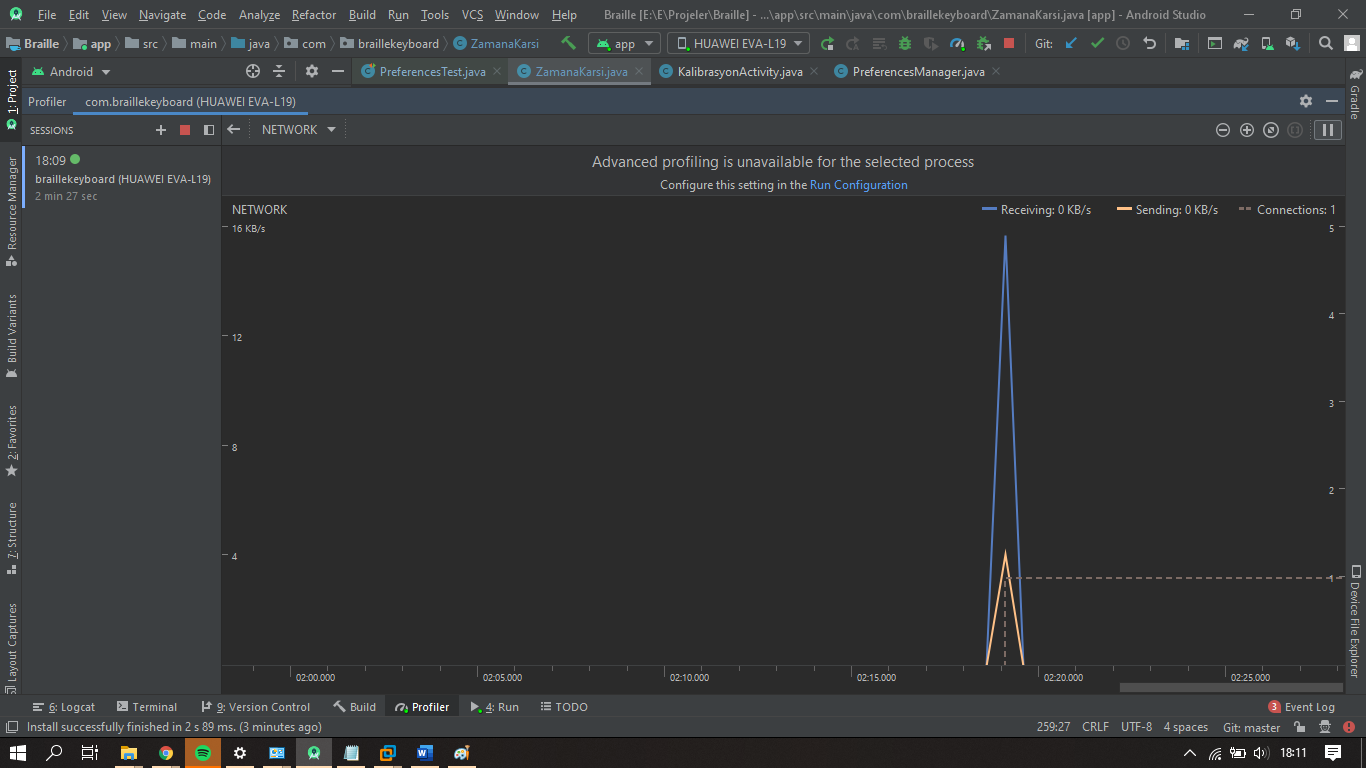
## CPU Testi

CPU testi incelendiğinde, uygulamanın sadece **%1 CPU** kullandığı görülmektedir. Oldukça düşük seviye olan bu değerden çıkarılacak sonuç ise birçok telefonda rahatlıkla bu uygulamanın çalıştırılabilecek olmasıdır.



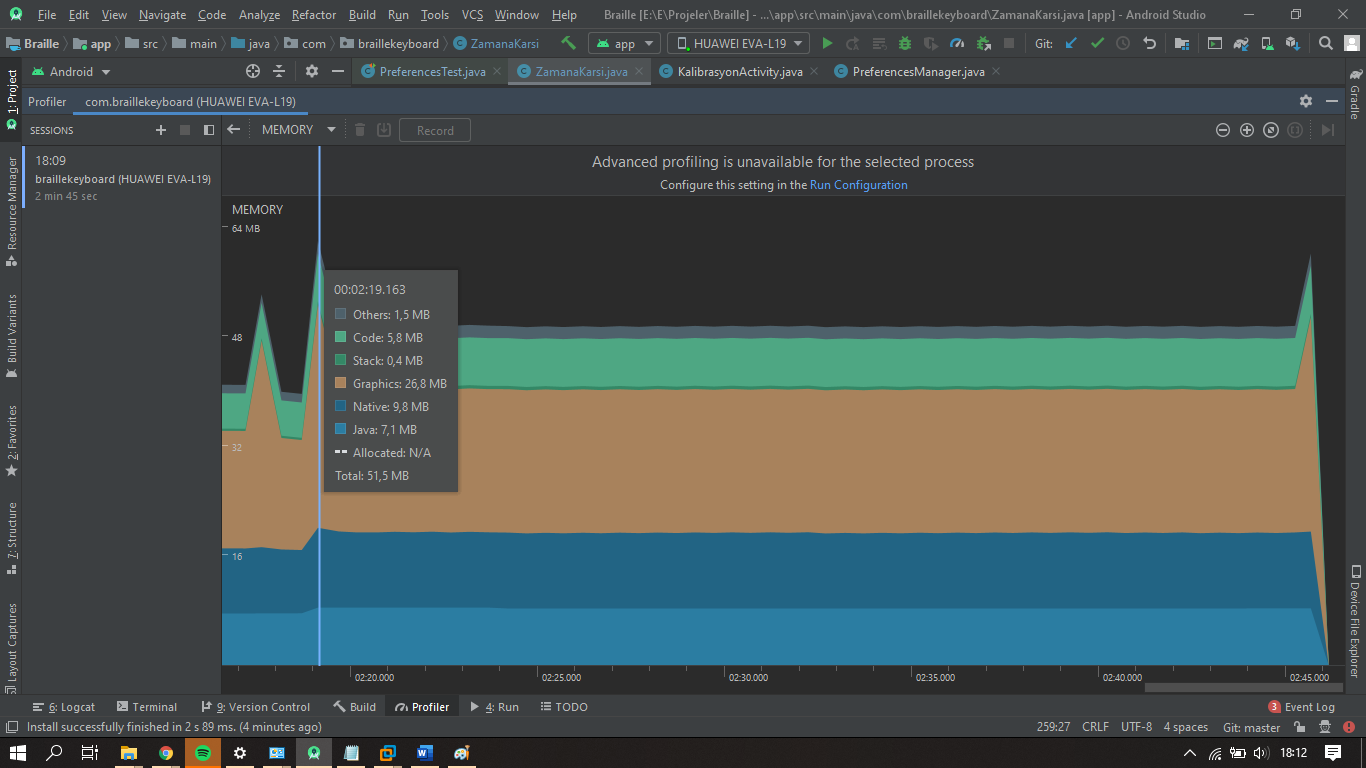
## Network Testi

Uygulama içerisindeki network işlemleri çok fazla yük olmamaktadır. Uygulamanın en çok veri çeken bölmelerinden birisi zamana karşı yarış kısmıdır. Bu kısımda çekilen değerler sadece 16 KB’lık boyuttadır ve oldukça düşük olduğu görülmektedir.



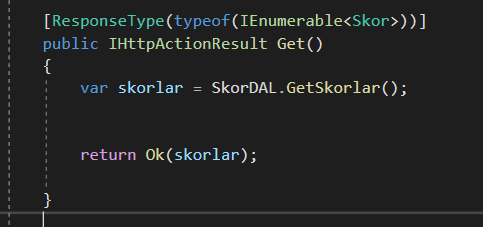
## RAM Testi

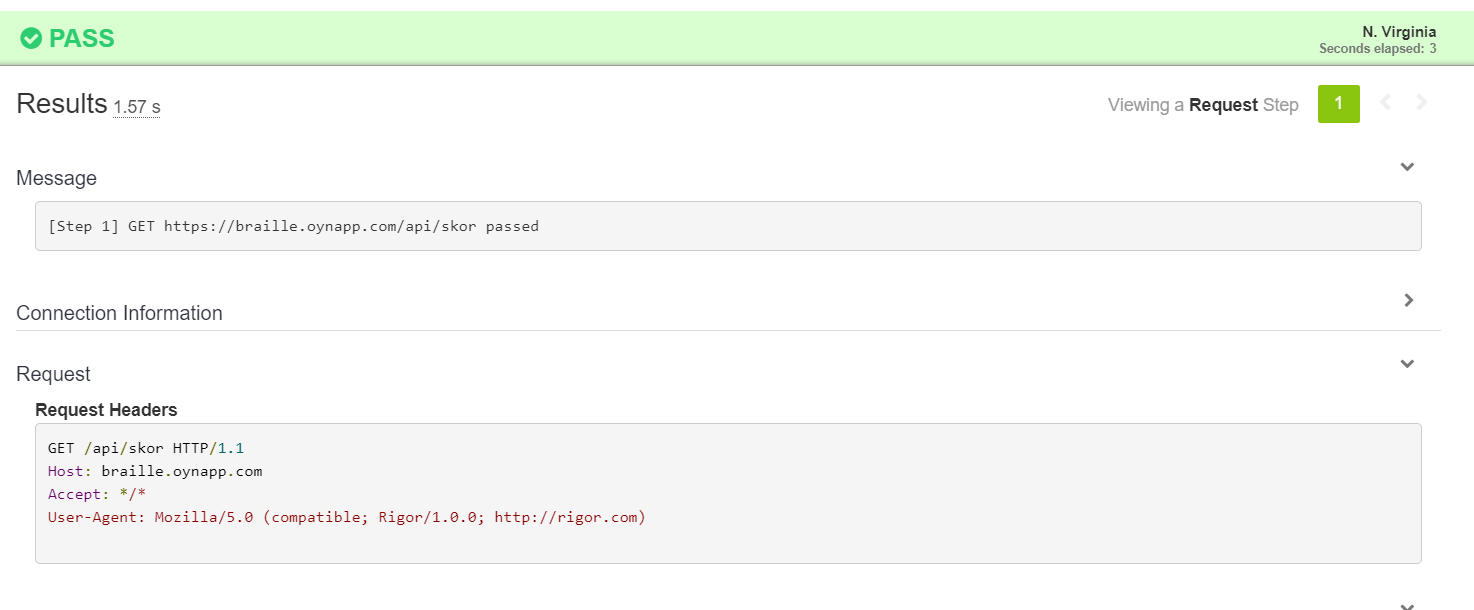
RAM testi yapıldığında, uygulamanın toplamda 51 MB kullandığı görüntülenmektedir. Gününümüzde Android cihazların RAM miktarı oldukça yüksektir. Yüksek olan değerler gözetildiğinde bu değerin oldukça küçük olduğu görülmektedir.



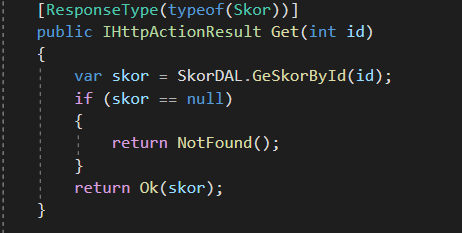
# API Testi

## Skor Get Testi



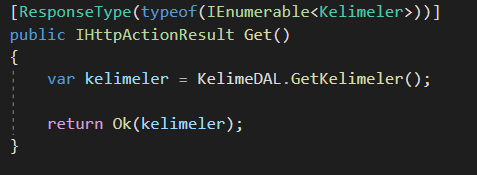


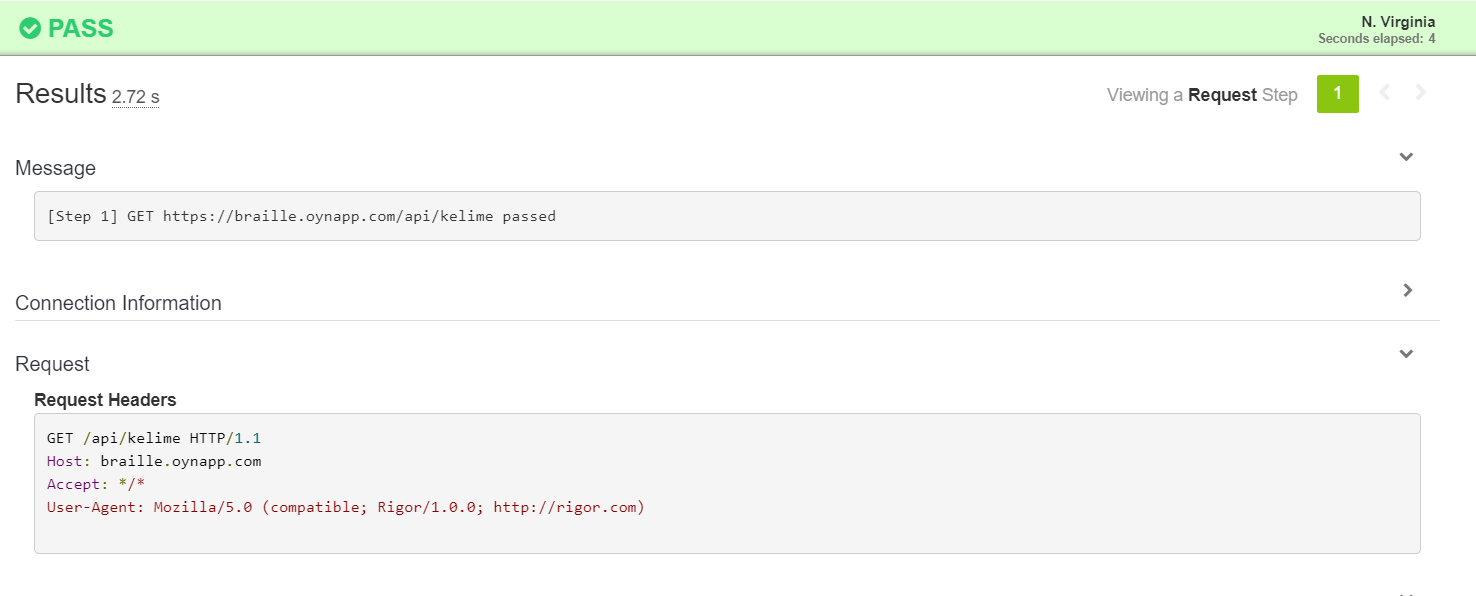
## Skor Id ile Get Testi



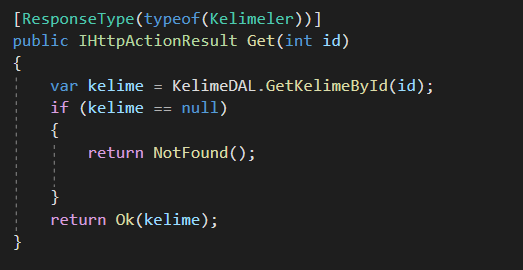


## Kelime Get Testi



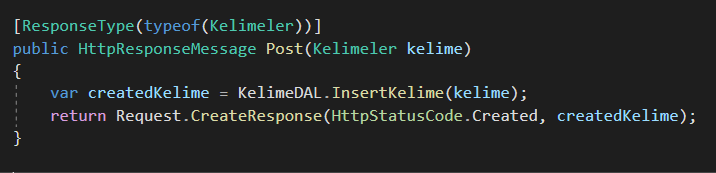


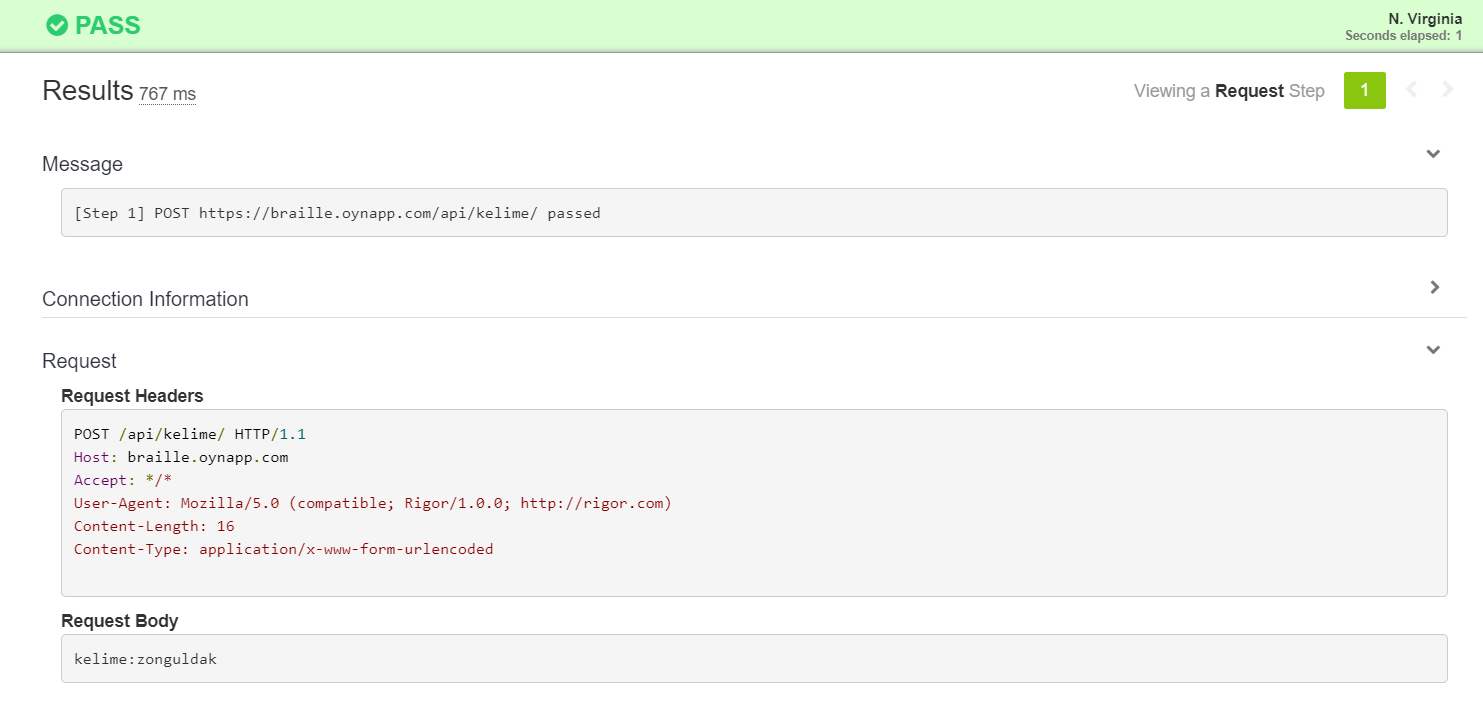
## Kelime id ile Get Testi



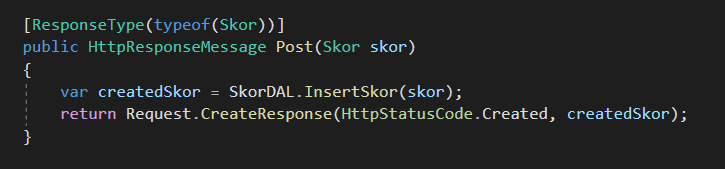


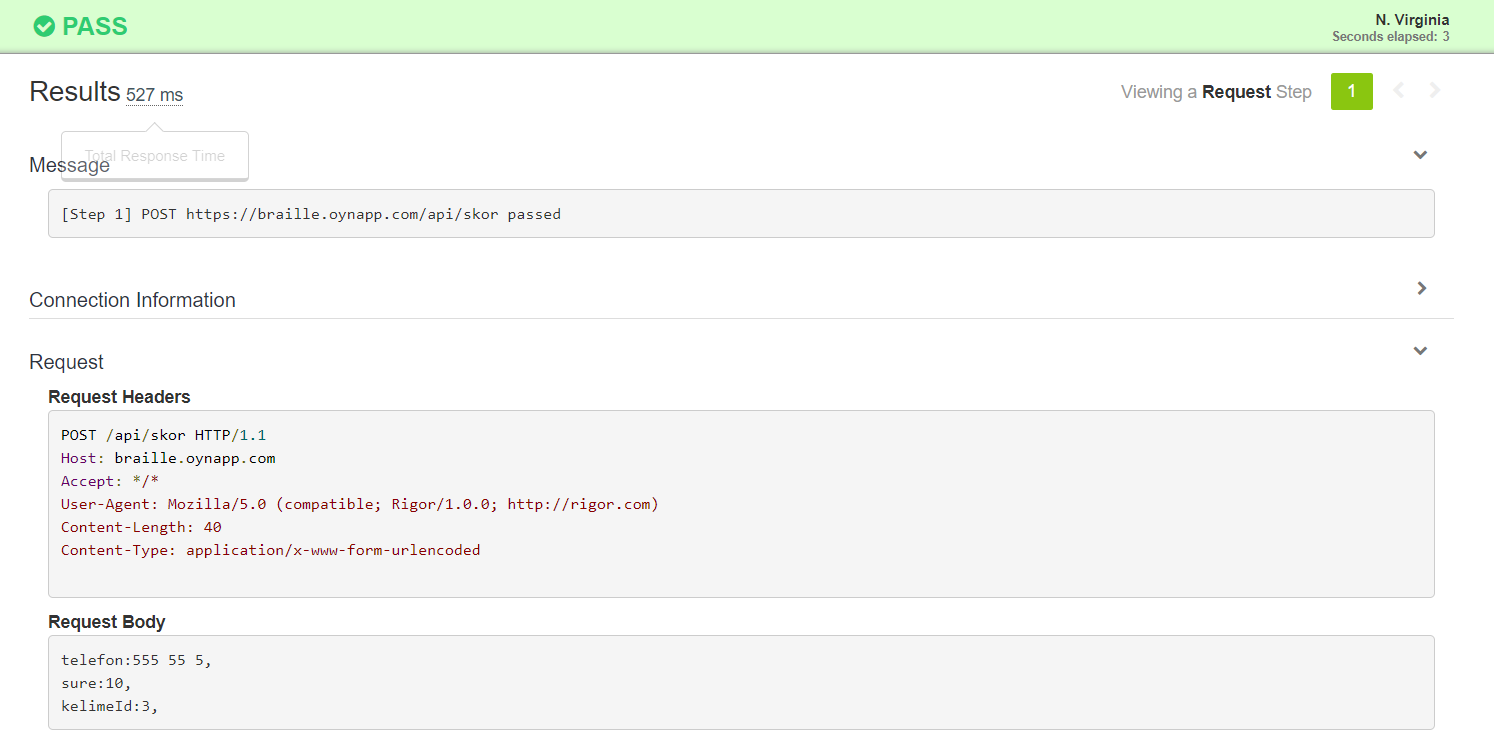
## Kelime Post/ Insert Testi



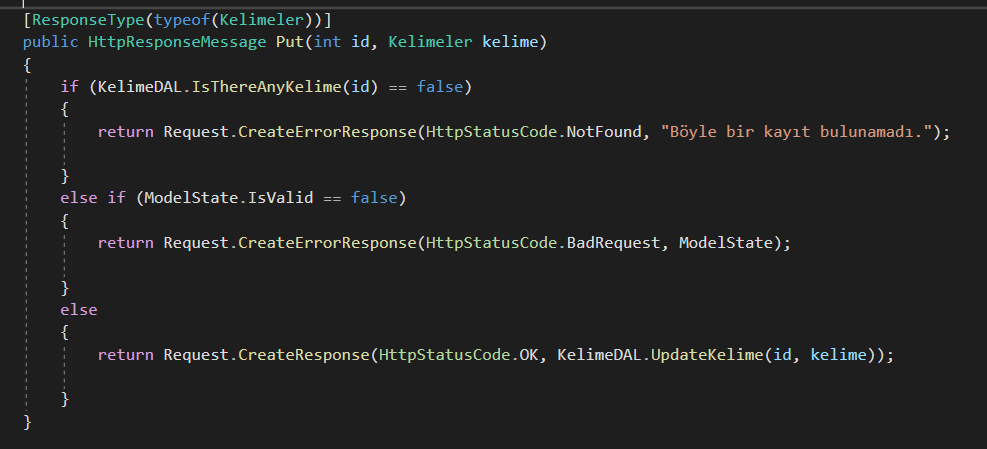


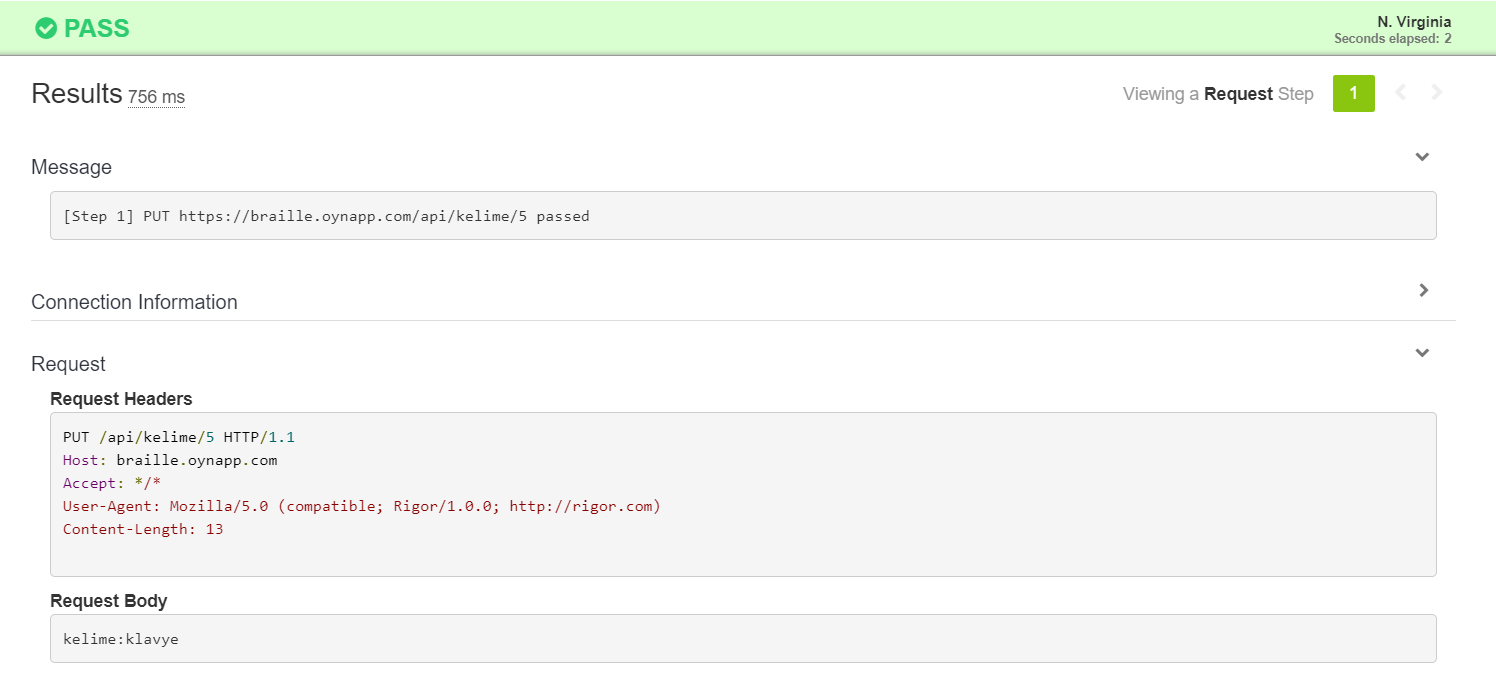
## Skor Post/Insert Testi



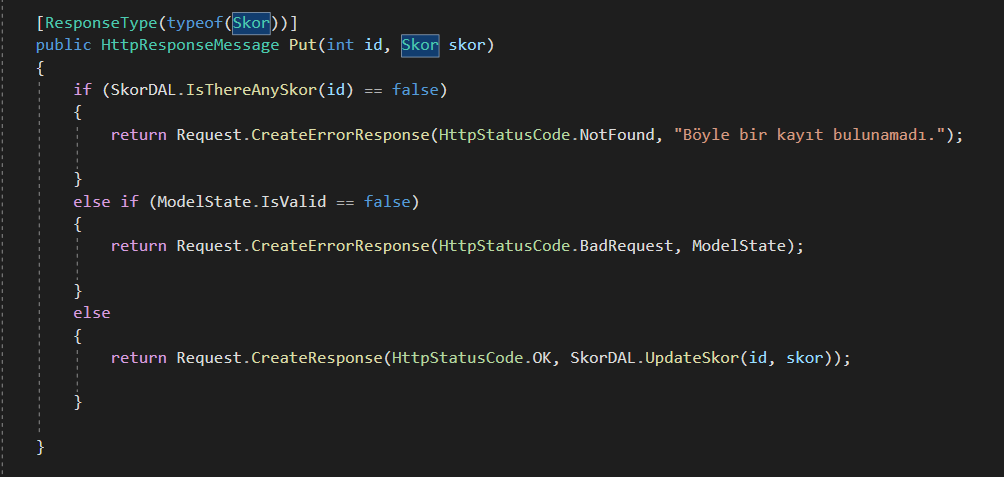


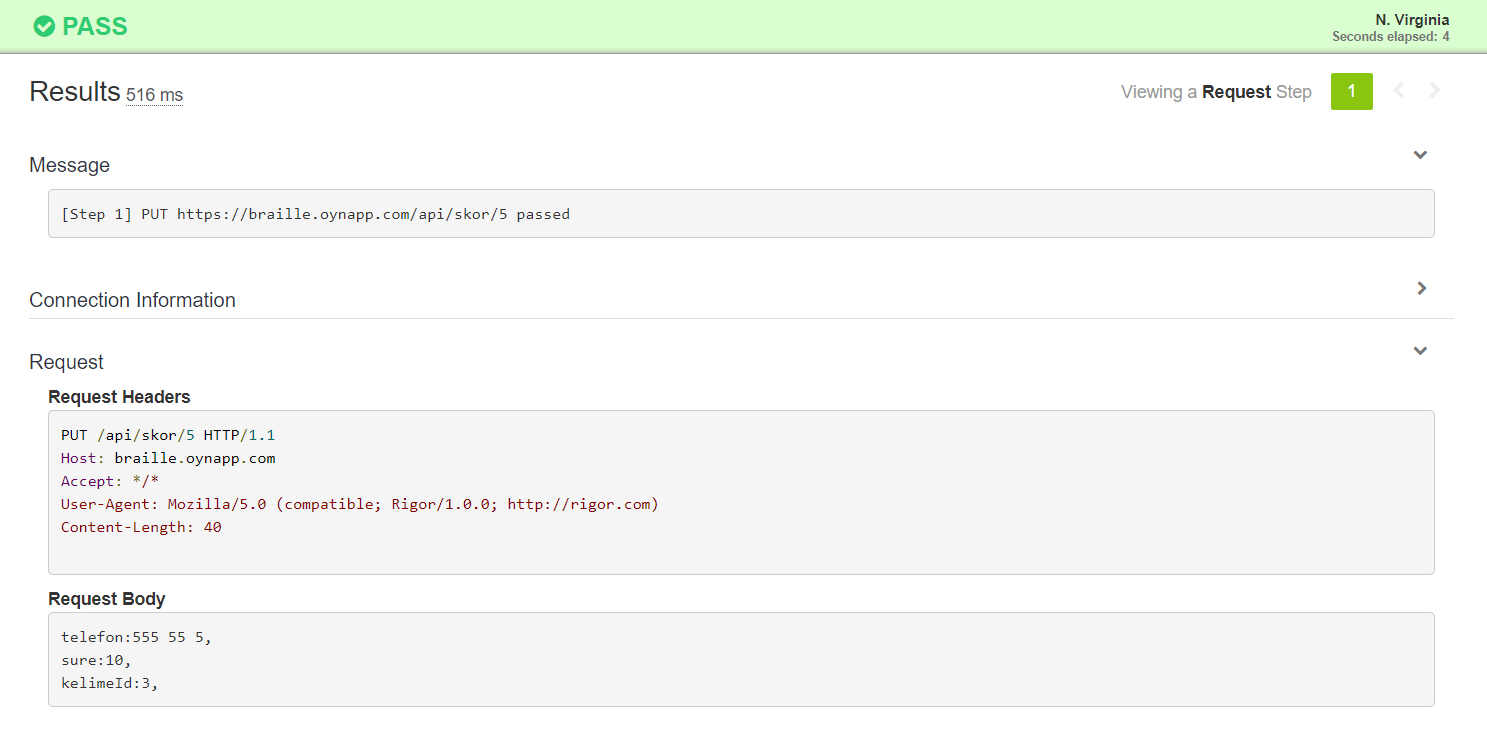
## Kelime Post/Update Testi





## Skor Post/Update Testi



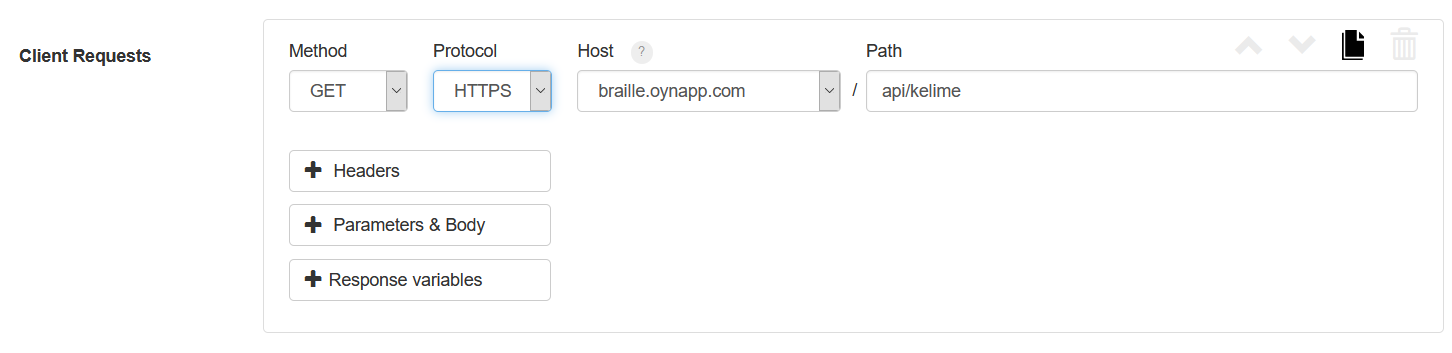


# Stres Testleri

API stres testleri **loader.io** yardımıyla yapılmıştır. Ücretsiz planda 10.000 kişiye kadar test clientlarına izin vermektedir.

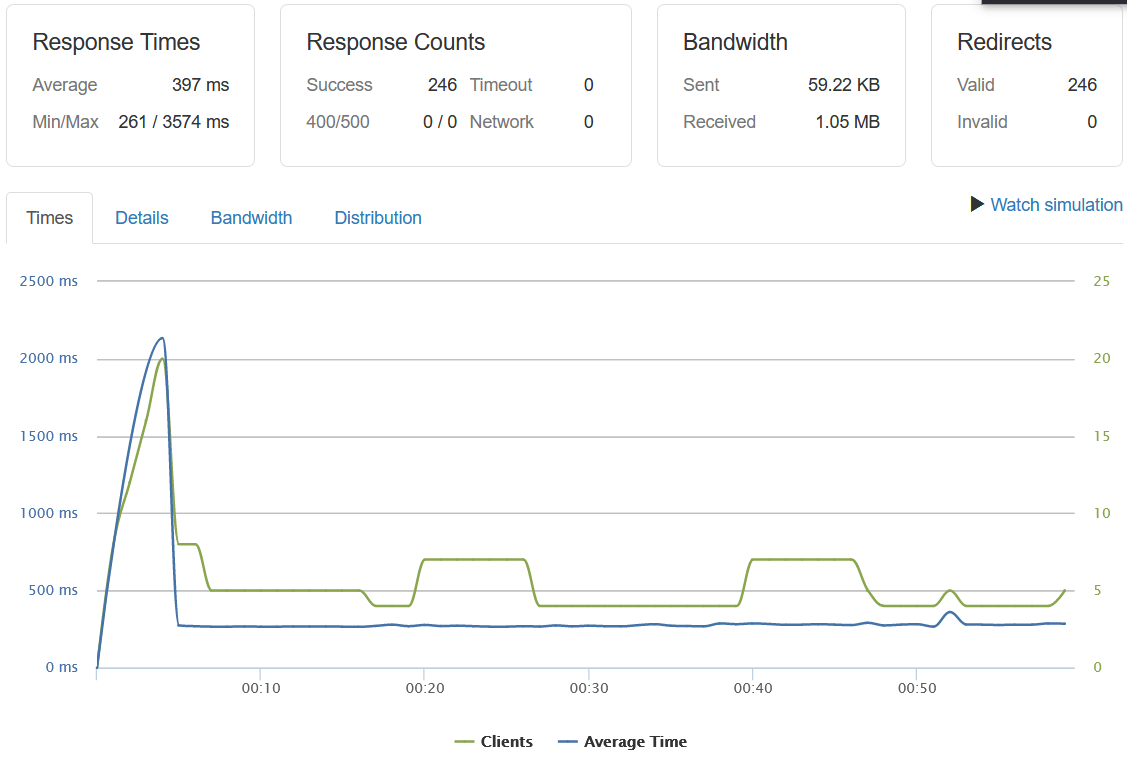
## 250 Client ile GET Request Testi

GET request ilgili adrese gönderilmiştir.



Elde edilen sonuçlarda, 59.22 KB veri gönderilmiş olup 1.05 MB veri çekilmiştir. Bu işlem 250 adet client ile aynı anda yapılmıştır. 250 request gönderilmiştir fakat sadece 4 request down olmuştur.

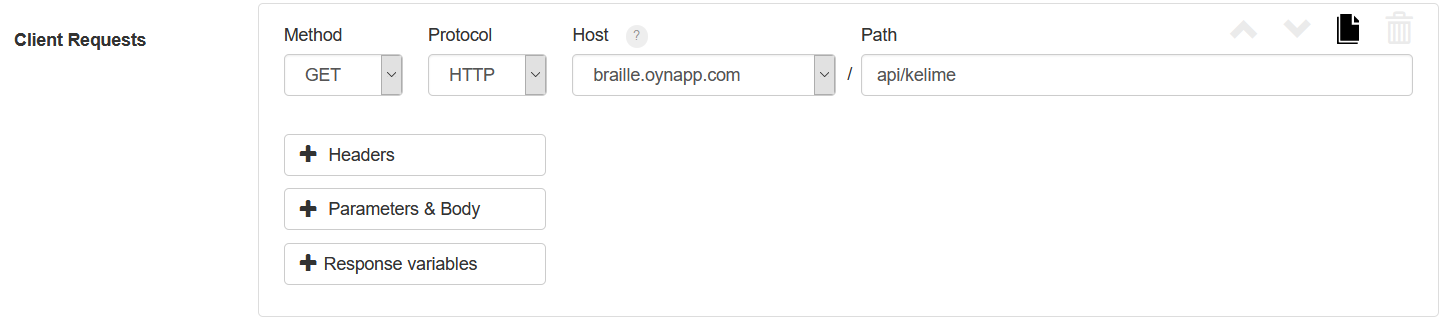
**Test Başarı Oranı: %98.4 (Ulaşmayan paket sayısına göre)**



Yaptığımız testin detaylarına ilgili adresten ulaşılabilir: <https://bit.ly/3fXsfM2>

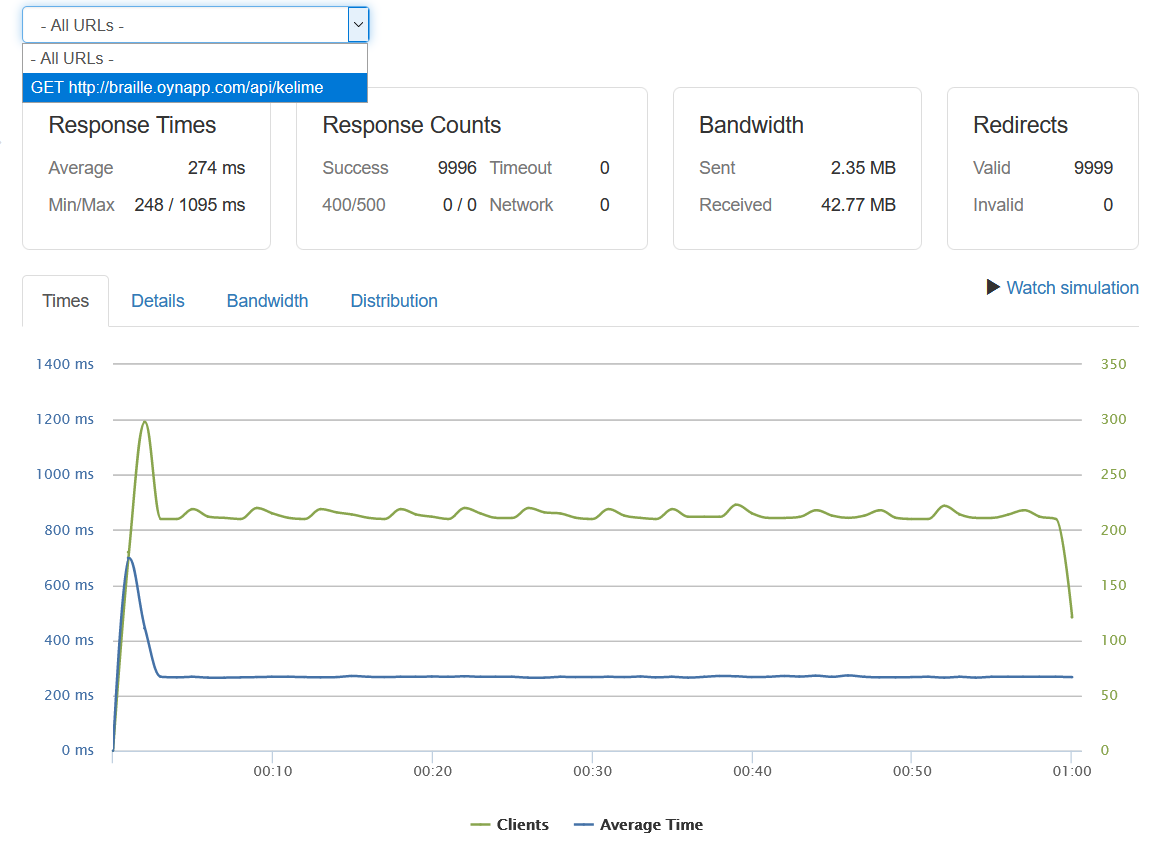
## 10.000 Client ile GET Request Testi

250 clienta benzer olarak sadece client sayısı artırılmıştır. Test sonuçları 250 client testinden farklı sonuçlar vermemiştir.



Test sonuçlarına göre, 4 request down olmuştur ve 2.35 MB gönderilmiş olup 42.77 MB alınmıştır. Sunucu bu istekleri rahatlıkla karşılamıştır. Hedef kitlemiş olan 5.000 kişi olması sebebiyle başarılı bir test olarak nitelendirilebilmektedir.

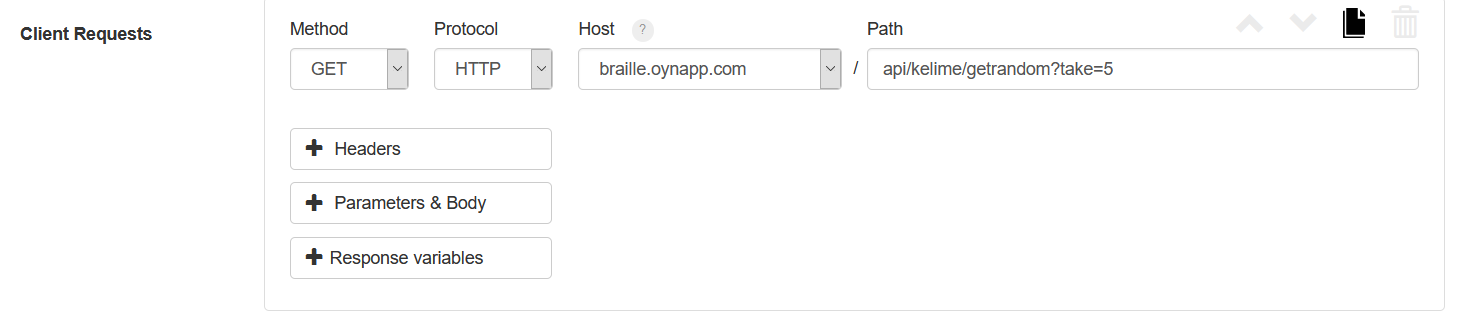
**Test Başarı Oranı: %99.96 (Ulaşmayan paket sayısına göre)**



Yaptığımız testin detaylarına ilgili adresten ulaşılabilir: <https://bit.ly/2ACU3oM>

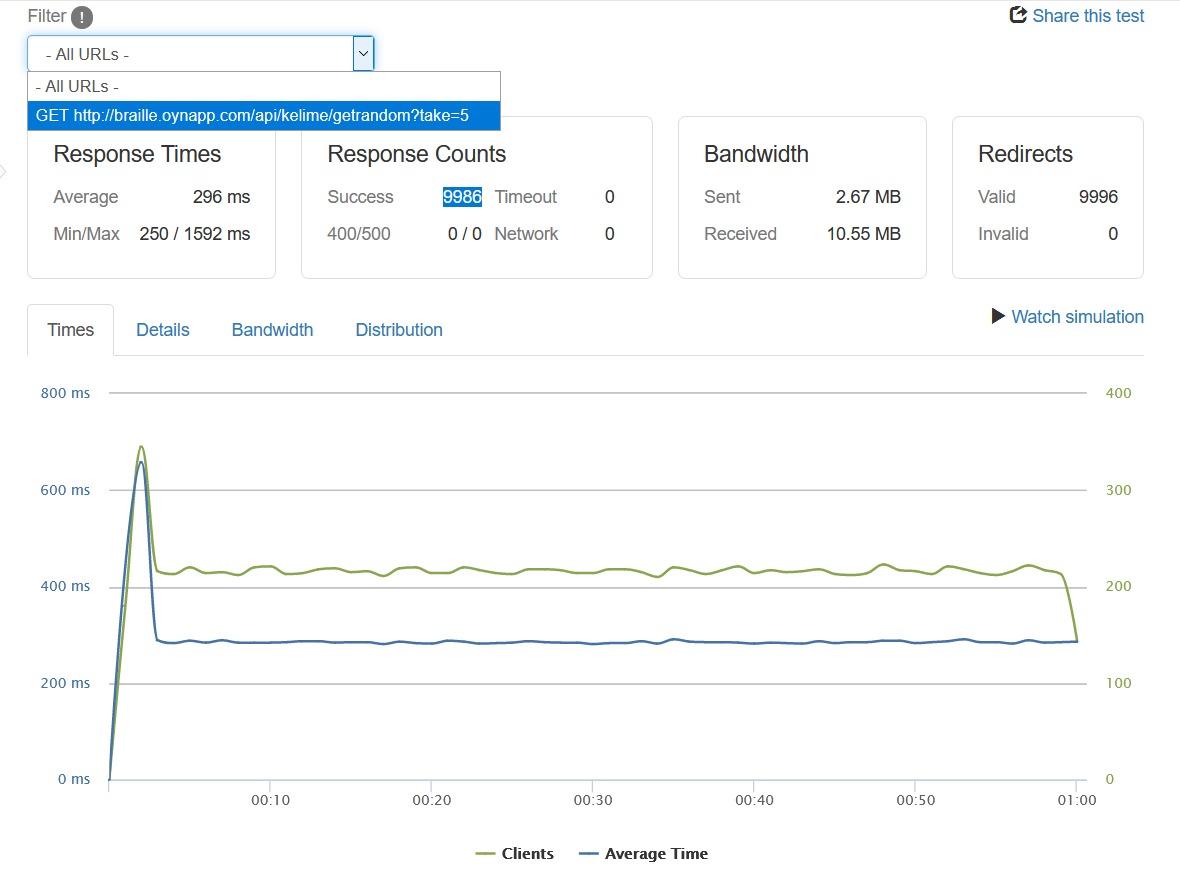
## 10.000 Client ile Random 5 Kelime GET Request Testi

Normal GET requesti yapıldığında tüm kelimeler gelmektedir. Bu tip GET requestinde ise **take** parametresi yardımıyla 5 kelime alınmıştır. İlk GET testini rahatlıkla geçtiği göz önüne alınırsa bu test ona göre çok daha başarılı geçmelidir.



Görüldüğü üzere, 2.67 MB gönderilen ve 10.55 MB alınan veri bulunmaktadır. 14(14/10000 oranında) request down olmuştur ancak sunucu neredeyse tüm isteklere rahatlıkla cevap verebilmiştir.

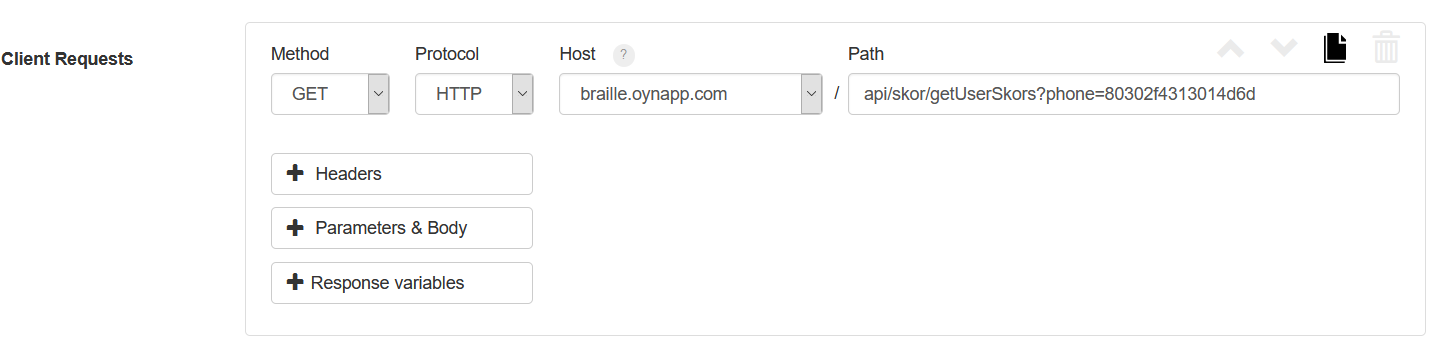
**Test Başarı Oranı: %99.86 (Ulaşmayan paket sayısına göre)**



Yaptığımız testin detaylarına ilgili adresten ulaşılabilir: <https://bit.ly/2WGEtkx>

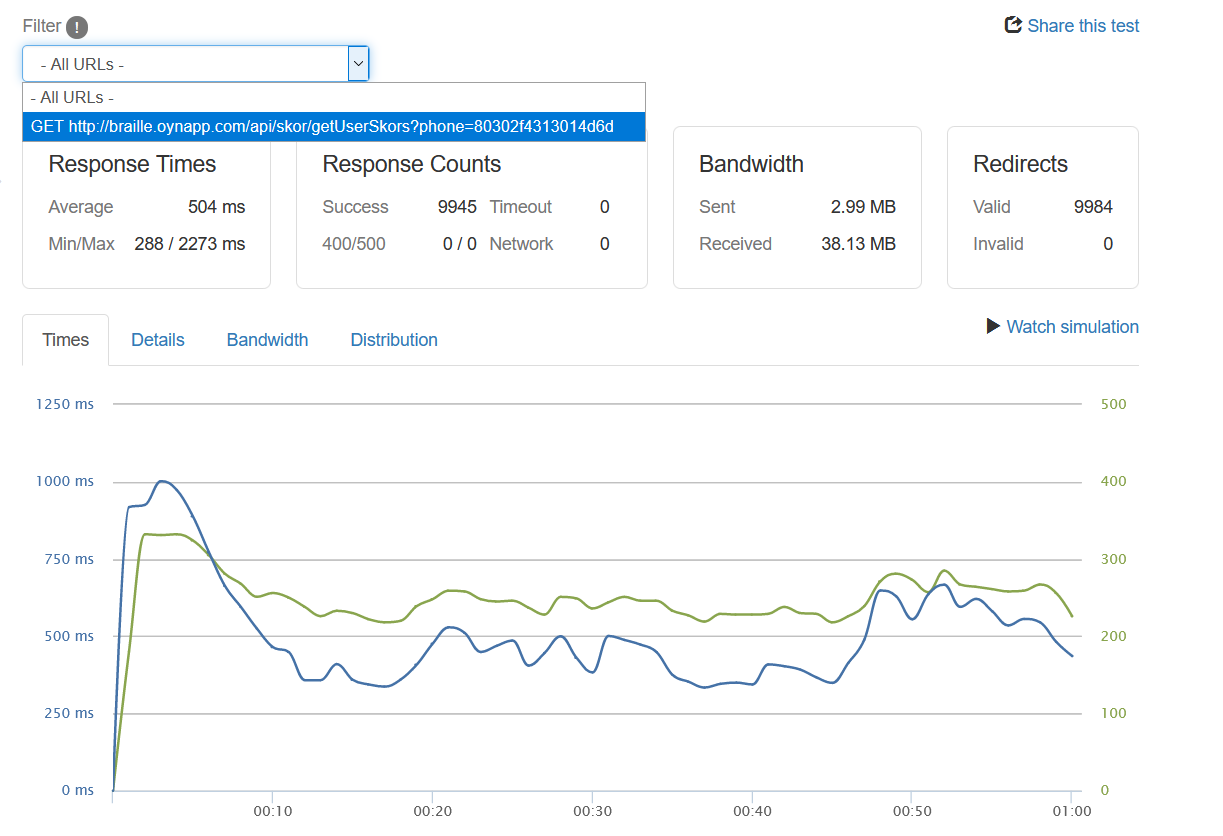
## 10.000 Client ile Skor GET Request Testi

Kelime testinden farklı olarak kullanıcıya göre farklı boyutta veriler alınmaktadır. Kullanıcının kaydettiği veriler kendi ID’sine göre response oluşturmaktadır.



Test sonuçlarında gönderilne 2.99 MB iken alınan veri 38.13 MB olmuştur. Yapılan diğer testleri göre daha fazla paket down olmuştur ancak 55/10000 paket göz önüne alındığında oldukça küçük bir oran olduğu gözlemlenmektedir.

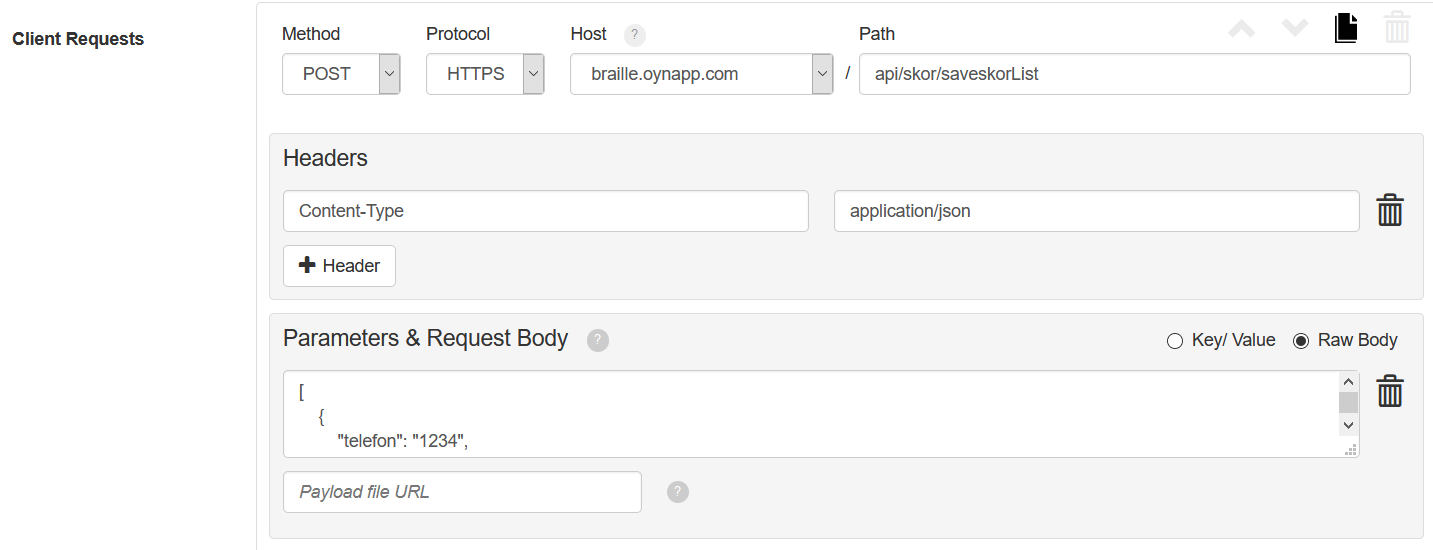
**Test Başarı Oranı: %99.45 (Ulaşmayan paket sayısına göre)**



Yaptığımız testin detaylarına ilgili adresten ulaşılabilir: <https://bit.ly/2TuK2Rh>

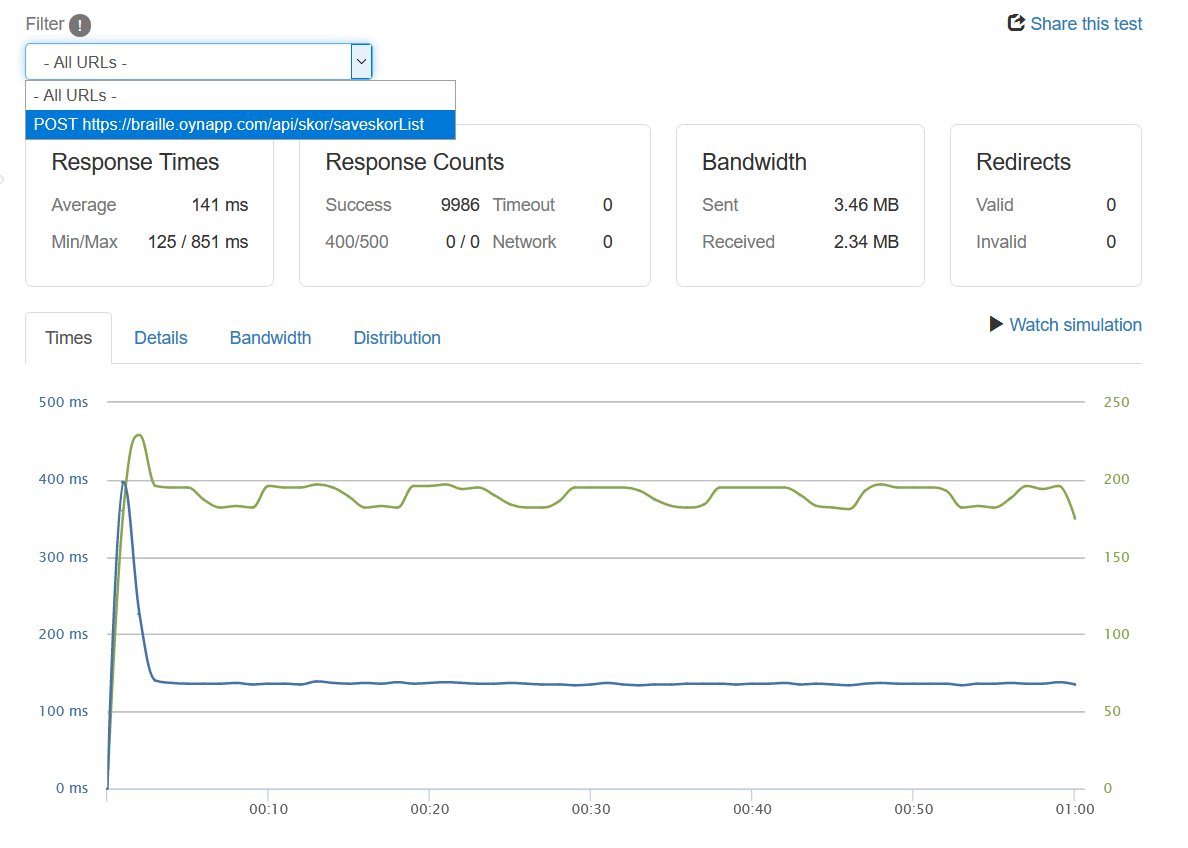
## 10.000 Client ile Skor POST Testi

Zamana karşı yarış sonucunda veriler sunucuya POST edilmektedir. POST edilen değerler JSON formatında gönderilmektedir. Bu nedenle gönderilen veri boyutunun alınandan fazla olması beklenmektedir.



Test sonuçları incelendiğinde, 14 paketin down olduğu gözlemlenebilmektedir. Gönderilen veri 3.46 MB iken alınan veri beklendiği üzere gönderilenden daha düşüktür ve 2.34 MB olarak görüntülenmiştir. 14/10000 oranında down olan paket bulunmaktadır.

**Test Başarı Oranı: %99.86 (Ulaşmayan paket sayısına göre)**



Yaptığımız testin detaylarına ilgili adresten ulaşılabilir: <https://bit.ly/3cHJfDM>