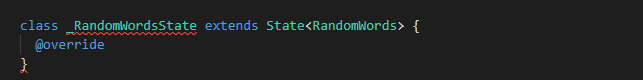
**V- STATEFUL WİDGET EKLEME**

Stateless Widget’lar immutable’dır. Yani özellikleri değiştirilemez, durum değişikliği söz konusu değildir.Tüm değerler sabittir(final). Stateful Widget’lar da ise widget var olduğu süre boyunca state’ deki değişikliklerle kendilerini yeniden oluşturan widgetlardır ve stateful widget kullanabilmemiz için en az iki classa ihtiyaç duyarız:

* [Stateful Widget class](https://docs.flutter.io/flutter/widgets/StatefulWidget-class.html)
* [State class](https://docs.flutter.io/flutter/widgets/State-class.html)

Bu adımda ise uyulamamıza bir statuful widget ekleyeceğiz.Widgetımızın adı “ RandomWords” ve state classımızın adı ise “ RandomWordsState” olacak şekilde classlarımızı yazalım…

Sınıflarımızı yazdıktan sonra IDE bize bir şeylerin eksik ya da yanlış olduğunu söyleyecek.



Bu hatanın sebebi build fonksiyonunun bulunmamasıdır. Build fonksiyonunu ekleyip içini ise yapmak istediğimiz işlemlerle dolduracağız.

Daha sonra ilk yazdığımız Stateless widgetı da düzenleyerek uygulamamızı geliştirmeye devam edelim.

class RandomWordsState extends State<RandomWords> {  
 @override *//... bu satırdan*  
 Widget build(BuildContext context) {  
 final WordPair wordPair = WordPair.random();  
 return Text(wordPair.asPascalCase);  
 } *// buraya kadar ekledik.*  
}

İpucu: Eğer widget\_test.dart sayfasında hata alıyorsanız class ismini kontrol ediniz.

class MyApp extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 // final WordPair wordPair = WordPair.random(); *// bu satırı* // siliyoruz.  
   
 return MaterialApp(  
 title: 'Welcome to Flutter',  
 home: Scaffold(  
 appBar: AppBar(  
 title: Text('Welcome to Flutter'),  
 ),  
 body: Center(  
 *//child: Text(wordPair.asPascalCase), // bu satırı silip*  
 child: RandomWords(), *// ... bu satır ekliyoruz.*  
 ),  
 ),  
 );  
 }  
}

**VI- LISTVIEW OLUŞTURMA**

Bu adımda ise random olarak oluşan kelimeleri bir listview ile düzenli ve sıralı bir şekilde kaydırarak görebileceğimiz şekilde oluşturalım.

**>>** İlk olarak \_suggestions adında bir List tanımlayalım. Ayrıca yazı tipi boyutunu büyütmek için \_biggerFont değişkeni tanımlayalım.

Daha sonra ise \_buildSuggestions() isimli yeni bir fonksiyon yazıyoruz. Bu fonksiyonumuz kelimelerin doğru bir şekilde eşleşmesini sağlayacaktır. Ayrıca Listview classımız itembuilder — iki parametre alır. Bunlar BuildContext, ve bir iterator’dür(sayac). İterator 0'dan başlar ve fonksiyon her çağrıldığında 1 artar. — Bu şekilde kullanıcı listview’i her kaydırdığında iterator 1 artarak sonsuza kadar devam eder.

class RandomWordsState extends State<RandomWords> {  
 *// Bu iki satırı ekleyelim.*  
 final List<WordPair> \_suggestions = <WordPair>[];  
 final TextStyle \_biggerFont = const TextStyle(fontSize: 18);   
 ...  
}

**>>** Şimdi ise aşağıda verilen \_buildSuggestions fonksiyonunu RandomWordsState classına yerleşirelim.\_buildSuggestionsfonksiyonu her kelime çifti için 1 kez \_buildRowfonksiyonun çağırır.

Widget \_buildSuggestions() {  
 return ListView.builder(  
 padding: const EdgeInsets.all(16),  
 itemBuilder: (BuildContext \_context, int i) {  
 if (i.isOdd) {  
 return Divider();  
 }  
  
 final int index = i ~/ 2;  
 if (index >= \_suggestions.length) {  
 \_suggestions.addAll(generateWordPairs().take(10));  
 }  
 return \_buildRow(\_suggestions[index]);  
 }  
 );  
 }

**>>** RandomWordsState’e bir \_buildRowişlevi ekleyelim.

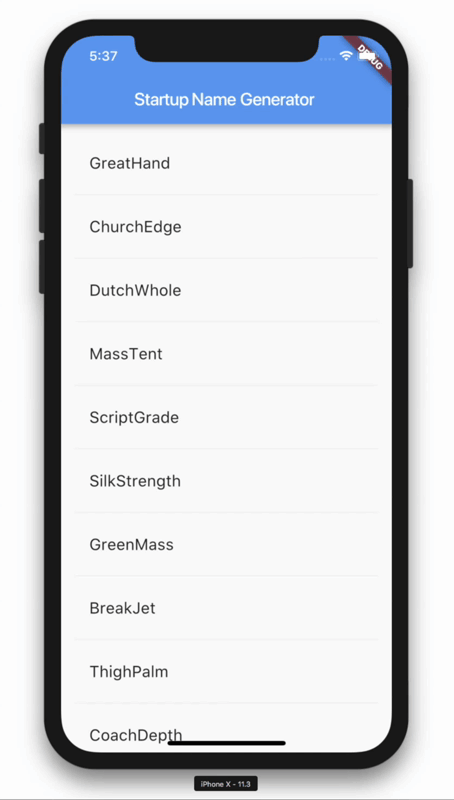
Widget \_buildRow(WordPair pair) {  
 return ListTile(  
 title: Text(  
 pair.asPascalCase,  
 style: \_biggerFont,  
 ),  
 );  
 }

**>>**Scaffold, temel Material dizayn gibi görsel tasarımı ayarlayalım.

@override  
 Widget build(BuildContext context) {  
 *//final wordPair = WordPair.random(); // bu iki satırı...*   
 *//return Text(wordPair.asPascalCase); // ... siliyoruz.*  
  
 return Scaffold ( *// buradan...*   
 appBar: AppBar(  
 title: Text('Startup Name Generator'),  
 ),  
 body: \_buildSuggestions(),  
 ); *// ... buraya kadar ekliyoruz.*  
 }

**>>** Son olarak ise başlığı ve home’u RandomWords çağıracak şekilde değişiklikleri yapalım.

@override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 title: 'Startup Name Generator',  
 home: RandomWords(),  
 );  
 }



Part 1’ in sonuna geldik :) Part 2 de ise mevcut uygulamamızın üstüne yeni şeyler ekleyerek devam edeceğiz.

https://medium.com/flutter-t%C3%BCrkiye/flutter-ile-i%CC%87lk-uygulaman%C4%B1-yazmaya-haz%C4%B1r-m%C4%B1s%C4%B1n-part-1-f2833af78685