

## ARAYÜZ İLE KOD BAĞLANTISI (2)

- Proje Oluşturma
- Arayüz Elemanları Oluşturma
  - View
  - Label
  - Textfield
  - Button
- Arayüz ile Kod Bağlantısı
  - Label Bağlantısı
  - TextField Bağlantısı
- Buton Eylemi Tanımlama
  - Button Bağlantısı

Bu dersimizde arayüz ile kaynak kod bağlantısı sağlayacağız. Storyboard'daki nesneler kaynak kod ile ilişkilidir.

**AutoLayout:** Arayüz tasarımında kullanılan nesnelerin ekran boyutuna göre dinamik yerleşimini sağlar.

**Scene:** Storyboard'daki bir içerik ekranını ve bir görünüm denetleyicisini (view controller) temsil eder.

**Nesne Kütüphanesi (Object Library):** Arayüz tasarımında kullanılacak nesneleri barındıran kütüphanedir.

**Özellik Denetçisi (Attributes Inspector):** Nesnelerin özelliklerinin ayarlandığı bölüm.

**Boyut Denetçisi (Size Inspector):** Nesnelerin boyutlarının ayarlandığı bölüm.

### Proje Oluşturma

1. Proje teması olarak **iOS** ve **Single View Application** seçin.
2. Proje bilgileri şu şekilde girin:
  - a. Product Name: **ConnectUltoCode**.
  - b. Team: Boş bırakın.
  - c. Organization Name: **Ktü**.
  - d. Organization Identifier: **ktu**.
  - e. Bundle Identifier: Bu kimlik otomatik olarak oluşturulacaktır.
  - f. Language: **Swift**
  - g. Devices: **Universal**
  - h. Use Core Data: Seçili değil.
  - i. Include Unit Tests: Seçili değil.
  - j. Include UI Tests: Seçili değil.

3. Bir sonraki adımda çalışma alanı belirlenir. **Create Git repository on** seçeneğini, seçili değil olarak belirleyin ve **Create** butonuna tıklayın.

Projeyi oluşturduk. Şimdi arayüz tasarımını yapalım. Arayüzde kullanılacak nesneler:

- View
- Button
- TextField
- Label

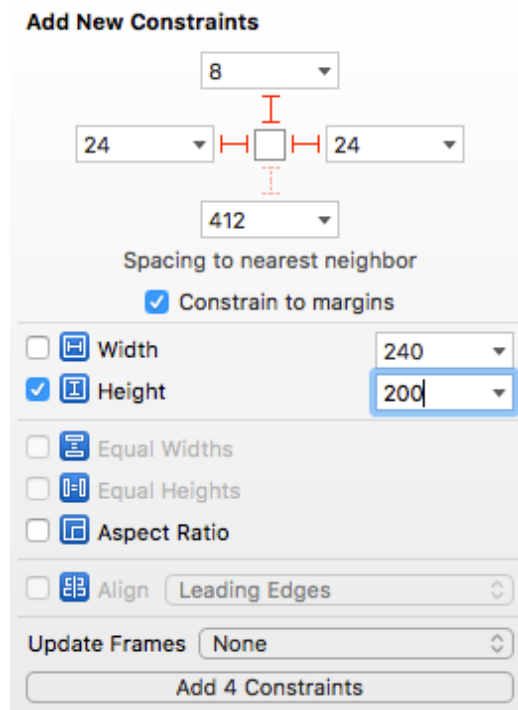
Arayüz elemanları AutoLayout özelliğini kullanılarak yerleştirilecektir.

### Arayüz Elemanları Oluşturma

Projede **iOS Simulator** ayarını **iPhone SE** olarak seçiniz.

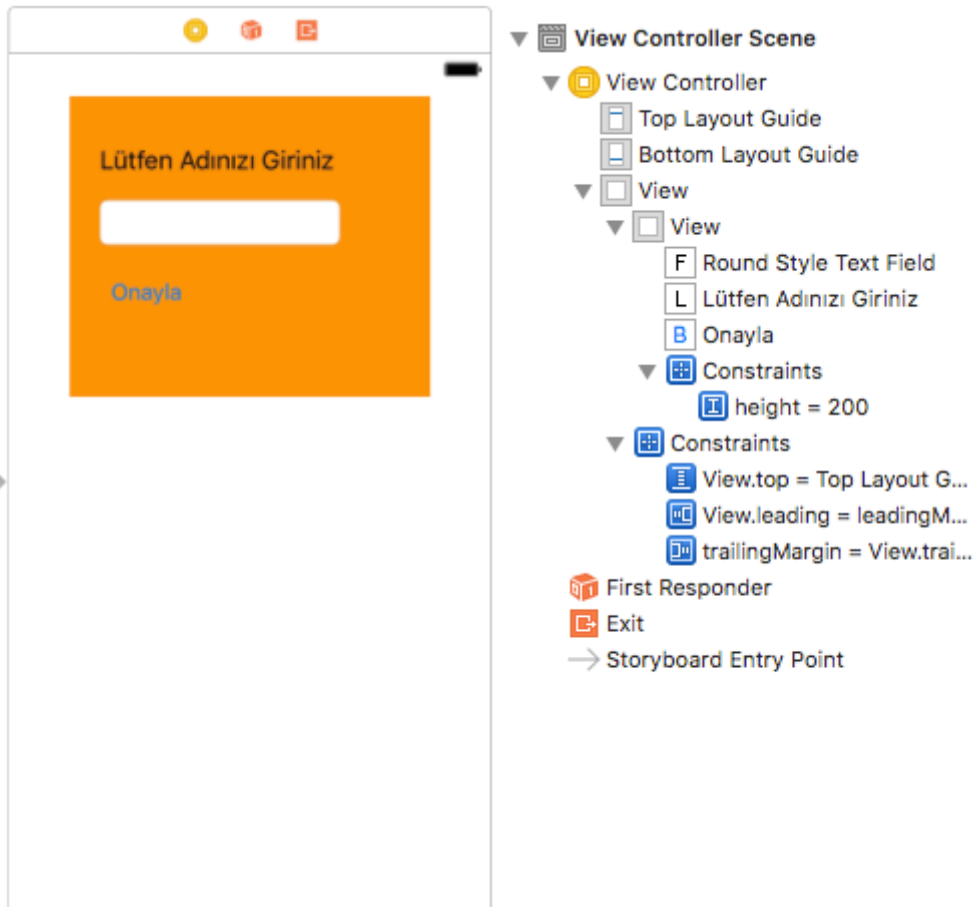
Main.storyboard dosyasında alt kısımda bulunan View as: iPhone SE olarak belirlemeyi unutmayınız.

1. Projede **Main.storyboard** dosyasını açın.
2. Nesne Kütüphanesinden bir **View** nesnesini sahneye sürükleyin.
3. **View** seçili haldeyken Özellik Denetçisinden arka plan rengini değiştirin. **Background -> Other -> Colors**.
4. **View** seçili iken Boyut Denetçisinden **X:40, Y:28; Width:240, Height:200** olarak girin.
5. **View** seçili iken alt kısımda bulunan **Add New Constraints** butonuna tıklayın ve şekildeki gibi değerleri girin.



6. Alt kısımda **View as** ile gösterilen diğer cihazları da seçerek arayüzün dinamik olduğunu test edin.
7. Nesne Kütüphanesinden bir **Label** nesnesini sahneye sürükleyin.
8. **Label** nesnesine çift tıklayarak “**Lütfen Adınızı Giriniz:**” yazın.
9. **Label** seçili iken Boyut Denetçisinden **X:20, Y:32, Width:160, Height:21** olarak girin.
10. Nesne Kütüphanesinden bir **TextField** nesnesini sahneye sürükleyin.
11. **TextField** seçili iken Boyut Denetçisinden **X:20, Y:69, Width:160, Height:30** olarak girin.
12. Nesne Kütüphanesinden bir **Button** nesnesini sahneye sürükleyin.
13. **Button** nesnesine çift tıklayarak “**Onayla**” yazın.
14. **Button** seçili iken Boyut Denetçisinden **X:20, Y:69, Width:64, Height:30** olarak girin.

Bu işlem adımları ile birlikte arayüz tasarımı tamamlandı. Arayüz ve View Controller Scene şu şekilde görünmelidir:

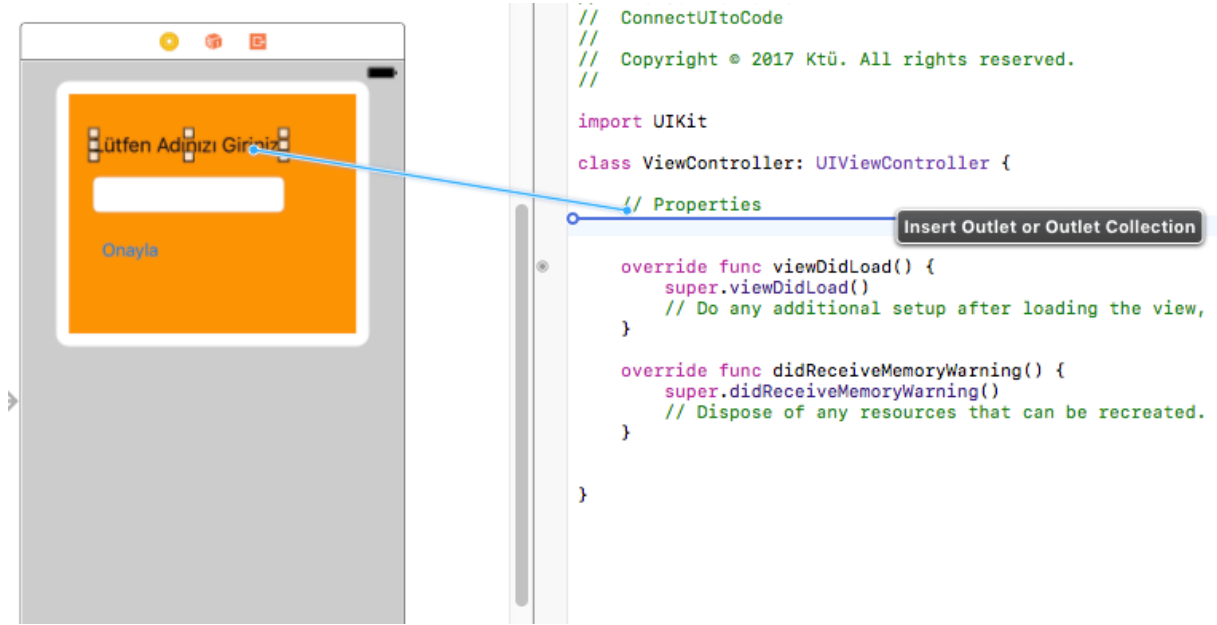


Simulator ile projenizi çalıştırıp tasarladığınız arayüzün gerçek bir cihazda nasıl görüneceğine bakabilirsiniz.

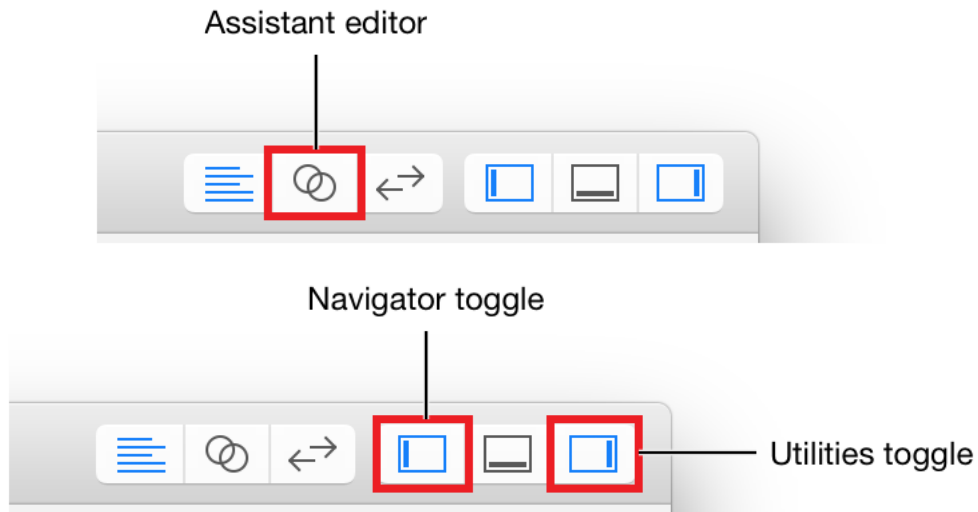
### Arayüz ile Kod Bağlantısı

**ViewController** sınıfı arayüz nesnelerine kaynak kod ile ulaşmayı sağlar. Arayüz nesnelerine kaynak kodu dosyalarından referans vermek için çıkış (**Outlet**) oluşturulur. Bir arayüz

nesnesinin çıkışı oluşturmak için **Ctrl** tuşuna basarak veya sağ tık ile kod ekranına sürüklenir. Bu işlem arayüz nesnesinin kaynak kodunda referansını oluşturur.



Yardımcı editörü (**Assistant Editor**) açmak için, **Xcode** araç çubuğunda sağ üst köşedeki Yardımcı (**Assistant**) butonuna tıklayın. Yeni açılan pencerede yardımcı editör butonundan **ViewController** sınıfını seçin.

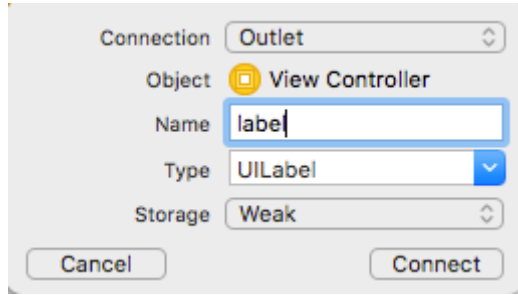


Daha fazla çalışma alanı için, **Xcode** araç çubuğundaki butonlar ile kendinize daha fazla çalışma alanı açabilirsiniz.

Kaynak koda **Label** referansı bağlamak için:

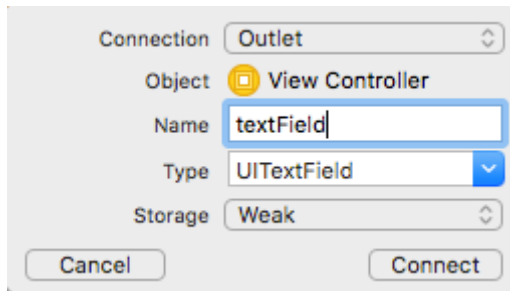
1. Projede **Main.storyboard** dosyasını açın.
2. İkinci ekrana **ViewController.swift** dosyası açıldığından emin olun.
3. **ViewController** sınıfının hemen altına **// Properties** yazarak kodun daha düzenli görünmesini sağlayabilirsiniz.
4. **Main.storyboard** dosyasında **Label** nesnesi seçin.

5. **Label** nesnesini **Ctrl** tuşuna basılı bir şekilde sürükleyip **ViewController** sınıfına bırakın.
6. Açılan pencerede **Label** nesnesine **label** adını verin ve **Connect** butonuna tıklayın.



Kaynak koda **TextField** referansı bağlamak için:

1. Projede **Main.storyboard** dosyasını açın.
2. İkinci ekrana **ViewController.swift** dosyası açıldığından emin olun.
3. **Main.storyboard** dosyasında **TextField** nesnesi seçin.
4. **TextField** nesnesini **Ctrl** tuşuna basılı bir şekilde sürükleyip **ViewController** sınıfına bırakın.
5. Açılan pencerede **TextField** nesnesine **textField** adını verin ve **Connect** butonuna tıklayın.



```
@IBOutlet weak var label: UILabel!
```

```
@IBOutlet weak var textField: UILabel!
```

İşlemleri adım adım tamamladıktan sonra yukarıdaki kod parçacığını eklemiş olduk.

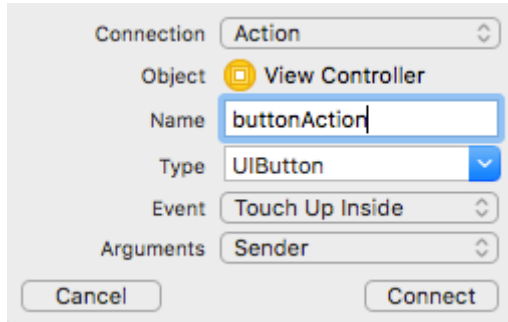
### Buton Eylemi Tanımlama

Arayüz nesnelerini koda bağladığımız gibi butonları da koda bağlarsak arayüz referansı değil de buton eylemi oluşturmuş oluruz. Aynı arayüz nesnelere çıkış verdiğimiz gibi eylemlere de çıkış veririz.

Kaynak koda **Button** eylemi bağlamak için:

1. Projede **Main.storyboard** dosyasını açın.
2. İkinci ekrana **ViewController.swift** dosyası açıldığından emin olun.
3. **Main.storyboard** dosyasında **Button** nesnesi seçin.
4. **Button** nesnesini **Ctrl** tuşuna basılı bir şekilde sürükleyip **ViewController** sınıfına bırakın.

5. Açılan pencerede **Connection** tipini **Action**, **Type** **UIButton** olarak seçin.
6. **Button** eylemine **buttonAction** adını verin ve Connect butonuna tıklayın.

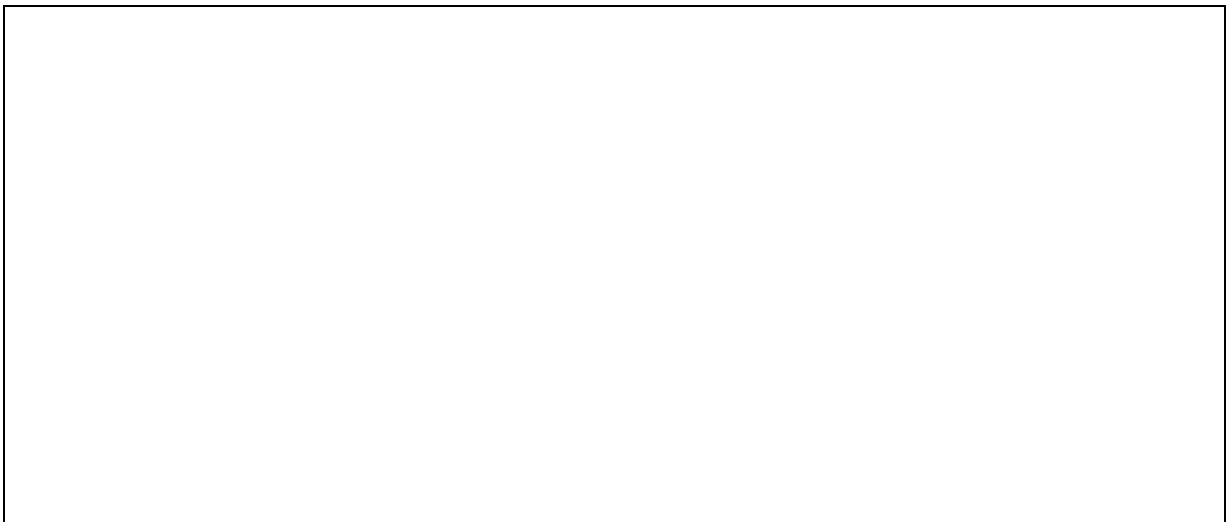


Yeni oluşturulan **buttonAction** fonksiyonu şu şekilde hazırlanacaktır.

```
// Action
@IBAction func buttonAction(_ sender: UIButton) {
    let value = "Merhaba " + textField.text!
    let alert = UIAlertController(title: "Mesaj",
                                message: value, preferredStyle: UIAlertControllerStyle.alert)
    alert.addAction(UIAlertAction(title: "Tamam",
                                style: UIAlertActionStyle.cancel, handler: nil))
    self.present(alert, animated: true, completion: nil)
}
```

Yazılan fonksiyon ile birlikte **TextFiled** nesnesine yazılan metin alınır. Metnin öncesine **"Merhaba"** metni eklenir. Daha sonra bu metin mesaj olarak ekrana yansıtılır. Bu işlem **UIAlertController** sınıfı ile gerçekleştirilir.

**Ne Öğrendik?**



Uygulamanızı çalıştırıp test edebilirsiniz. Resimdeki gibi bir sonuçla karşılaşmalısınız.

Lütfen Adınızı Giriniz

Onayla

Lütfen Adınızı Giriniz

Sefa

Onayla

**Mesaj**

Merhaba Sefa

**Tamam**