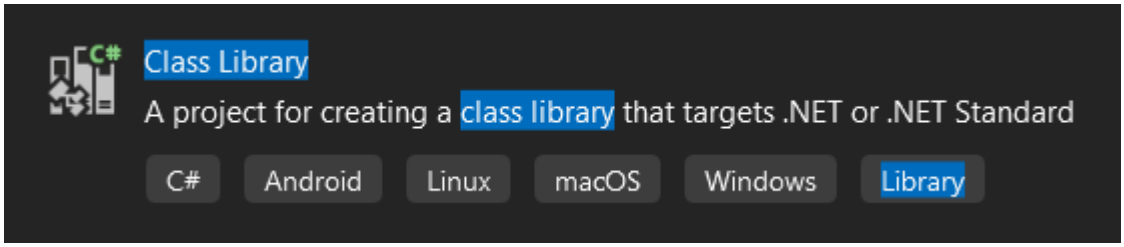
1. Class Library oluşturacağız.

Solution sağ click -> Add Project tıklayıp aşağıdaki projeyi seçiyoruz.



Bu projenin ismini "AppUser.Management.Service" yapalım.

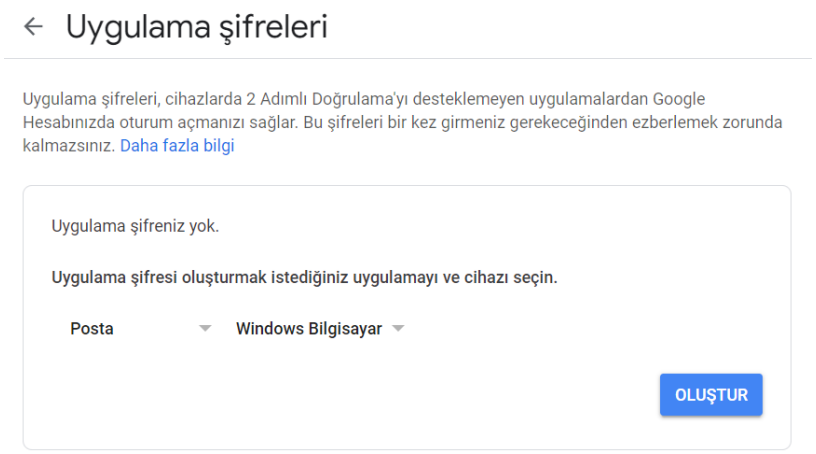
2. AUTHENTICATION projemizdeki appsettings.json dosyası içerisine "Email Ayarları" ekleyeceğiz. Bu ayarlar email gönderirken kullanılacaktır.

Google Hesabınızı Yönetin

Güvenlik

2 Adımlı Doğrulama kısmına giriniz.

En altta uygulama şifreleri var. Bir uygulama şifresi oluşturduktan sonra bu şifreyi kullanacağız. GOOGLE güvenlik önlemi aldığı için gmail şifresini kullanmamızı engellediğini teyit etmiştik. Uygulama şifresini kendisi oluşturarak var olan şifremizin ayarlar dosyasında vs. kullanılmasını engellemek amacıyla bu şekilde bir çözüm sunduğunu görmüş oluyoruz. "Uygulama Şifreleri" seçeneğine tıklayınız. Ve aşağıdaki seçenekleri seçtikten sonra "Oluştur" butonuna tıklayınız. Oluşturulan şifre "EmailConfig" içerisindeki "Password" özelliğine yazılacaktır.




```
"EmailConfig": {
      "From": "fatihudemy@gmail.com",
      "SmtpServer": "smtp.gmail.com",
      "Port": 465,
      "Username": "fatihudemy@gmail.com",
          // Buraya sizin oluşturmuş olduğunuz şifre gelecek.
      "Password": "xxxxxxxxxxxxxxxx"
}
```

3. AppUser.Management.Service kütüphanemize "NETCore.MailKit" kütüphanesini Nuget aracılığıyla ekleyelim.
4. Bu sınıf kütüphanesine "Configurations" klasörü ekleyelim ve içerisinde "EmailConfig.cs" sınıfı oluşturalım. Ve projedeki gibi içerisini dolduralım.

5. AppUser.Management.Service kütüphanesine "Models" klasörü ekleyip altına "MailMessage.cs" adında dosyamızı oluşturuyoruz ve içerisini projedeki gibi dolduruyoruz. MailboxAddress tipi MimeKit kütüphanesinde olduğu için ve MimeKit MailKit kütüphanesi tarafından kullanıldığın için tekrar referans etmemize gerek kalmıyor. Direkt kullanabiliyoruz.

6. AppUser.Management.Service kütüphanesine Services adında klasör ekliyoruz. İçerisinde IEmailService.cs adında interface oluşturuyoruz. Bu servis implemente edilen sınıf tarafından mail gönderme kurallarının belirlenmesini zorunlu kılacaktır.

7. IEmailService.cs dosyasının bulunduğu klasörde EmailService.cs adında bir dosya oluşturuyoruz. Bu dosyayı projedeki gibi dolduruyoruz.

8. EmailConfig.cs sınıfını Singleton (tek bir kullanım) olarak servislere dahil ediyoruz. Dahil ederken appsettings.json içerisinde kullanacağımız ayarları da belirtiyoruz. Tek instance'ımızın property'lerinin value'ları appsettings.json içerisinde belirtmiş olduğumuz değerler neyse artık onlar olacak.

9. IEmailService objesi kullanılacağı zaman otomatik olarak EmailService nesnesi yaratılması için scoped olarak servislere ekliyoruz.
(dependency injection – bağımlılık enjeksiyonu) (nesneyi kendimiz sınıf içerisinde oluşturmak yerine dışarıda oluşturmuş ve RAM'de hali hazırda bulunan nesnenin kullanılması diyebiliriz)

10. AUTHENTICATION projesinde AuthController.cs dosyasında AuthController sınıfına IEMailService field'ı ekliyoruz. Ve constructor'da parametre olarak alıyoruz. Projeden bakabilirsiniz.

11. AuthController içerisinde SendMail adında metot oluşturup içerisini projedeki gibi dolduruyoruz.

Artık bu aksiyonu çalıştırdığında belirlemiş olduğunuz email adreslerine emailleriniz gönderilecektir.

LÜTFEN BÜTÜN KODLARIN MANTIĞINI ANLAMAYA ÇALIŞINIZ. DİREKT KOPYALAYIP YAPIŞTIRIP GEÇMEYİNİZ!

Kolay gelsin.
Fatih Kaan AÇIKGÖZ