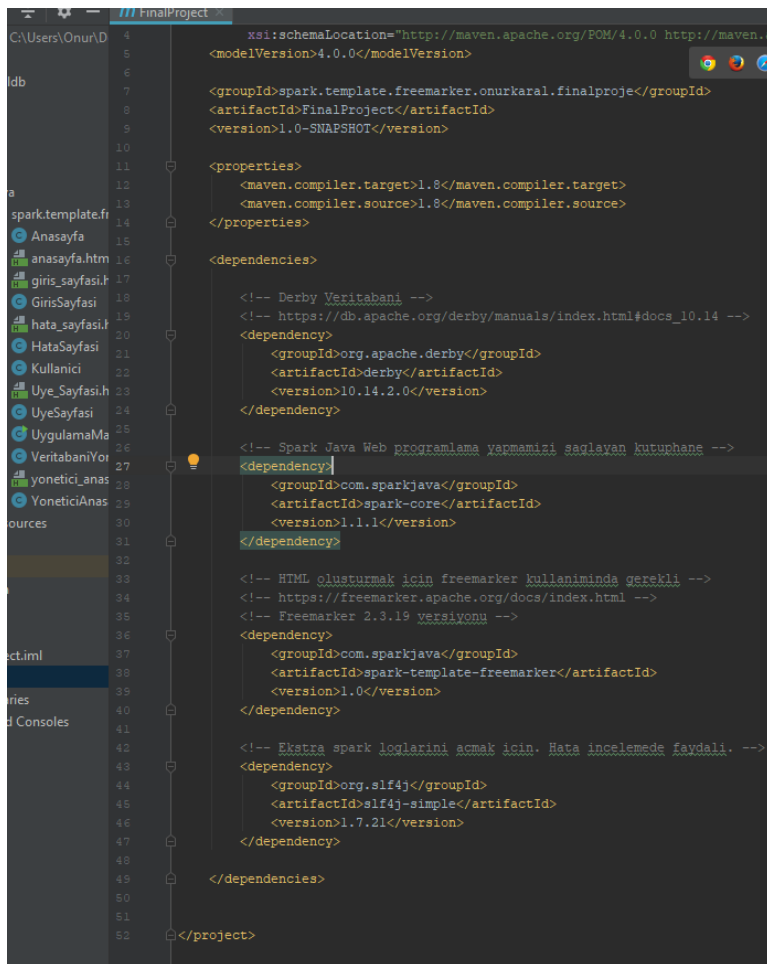


ILERI JAVA UYGULAMALARI - 2018 2019 GUZ (Ekim 2018 - Ocak 2019)

Onur Karal

H5170067

İlk olarak Maven projesindeki pom.xml dosyasına
“apache derby, spark.java,
spark.template.freemarker” kütüphanelerini
tanıttık.



```
4  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.a
5  <modelVersion>4.0.0</modelVersion>
6
7  <groupId>spark.template.freemarker.onurkaral.finalproje</groupId>
8  <artifactId>FinalProject</artifactId>
9  <version>1.0-SNAPSHOT</version>
10
11  <properties>
12    <maven.compiler.target>1.8</maven.compiler.target>
13    <maven.compiler.source>1.8</maven.compiler.source>
14  </properties>
15
16  <dependencies>
17    <!-- Derby Veritabani -->
18    <!-- https://db.apache.org/derby/manuals/index.html#docs_10.14 -->
19    <dependency>
20      <groupId>org.apache.derby</groupId>
21      <artifactId>derby</artifactId>
22      <version>10.14.2.0</version>
23    </dependency>
24
25    <!-- Spark Java Web programlama yapmamizi saglayan kutuphane -->
26    <dependency>
27      <groupId>com.sparkjava</groupId>
28      <artifactId>spark-core</artifactId>
29      <version>1.1.1</version>
30    </dependency>
31
32    <!-- HTML glusturmak icin freemarker kullaniminda gerekli -->
33    <!-- https://freemarker.apache.org/docs/index.html -->
34    <!-- Freemarker 2.3.19 versiyonu -->
35    <dependency>
36      <groupId>com.sparkjava</groupId>
37      <artifactId>spark-template-freemarker</artifactId>
38      <version>1.0</version>
39    </dependency>
40
41    <!-- Ekstra spark loglarini acmak icin. Hata incelemede faydali. -->
42    <dependency>
43      <groupId>org.slf4j</groupId>
44      <artifactId>slf4j-simple</artifactId>
45      <version>1.7.21</version>
46    </dependency>
47  </dependencies>
48
49  </project>
```

Daha sonrasında GirişSayfasi.java ile giris_sayfasi.html dosyalarını oluşturduk. Ve giriş sayfası tanımla fonksiyonu oluşturup Uygulama maininde bunu gösterdik.

```
private static void GirişSayfasiTanımla() {
    FreeMarkerRoute girisSayfasi = new FreeMarkerRoute( path: "/onurkaral/giris_sayfasi/" ) {
        @Override
        public Object handle(Request istek, Response cevap) {
            return new ModelAndView( model: null, viewName: "/onurkaral/finalproje/giris_sayfasi.html");
        }
    };
    get(girisSayfasi);
}
```

gir_sayfasi.html

GirisSayfasi

```
package spark.template.freemarker.onurkaral.finalproje;

public class UygulamaMain {

    public static Kullanici girisYapmisKullanici = null;

    public static void main(String[] args) {

        VeritabaniYonetimi.veritabaniBaglantisiKur();
        VeritabaniYonetimi.tablolariOlustur();
        VeritabaniYonetimi.kullaniciolariOlustur();
        GirişSayfasi.sayfalarıTanımla();
        UyeSayfasi.sayfalarıTanımla();
        Anasayfa.sayfalarıTanımla();
        YoneticAnasayfa.sayfalarıTanımla();
        HataSayfasi.sayfalarıTanımla();

    }
}
```

Kullanıcının Giriş sayfasında giriş formunu doldurduktan sonra anasayfaya geçiş için olan kısım yazıldı.

```
Route girisFormPostAction = new Route( path: "/onurkaral/giris_sayfasi/login") {
    @Override
    public Object handle(Request istek, Response cevap) {
        String kullaniciaadi = istek.queryParams("kullaniciaadi");
        Kullanici kullanici = VeritabaniYonetimi.kullaniciSorgula(kullaniciaadi);
        Kullanici ADMIN = VeritabaniYonetimi.kullaniciSorgula(kullaniciaadi);
        if(kullanici == null) {
            System.out.println("Verilen Kullanici Bulunamadi.! " + kullaniciaadi);
            // TODO hatayi sayfada goster
            cevap.redirect( location: "/onurkaral/giris_sayfasi/");
        } else {
            String sifre = istek.queryParams("sifre");
            if(kullanici.getKullaniciTipi().equals(kullaniciaadi)){
                UygulamaMain.girisYapmisKullanici = ADMIN;
                cevap.redirect( location: "/onurkaral/yonetici_anasayfa/");
            } else if(kullanici.getSifre().equals(sifre)) {
                System.out.println("Sifre Dogru. Giris Basarili.");
                UygulamaMain.girisYapmisKullanici = kullanici;
                cevap.redirect( location: "/onurkaral/");
            } else {
                System.out.println("Sifre Hatali.!");
                cevap.redirect( location: "/onurkaral/giris_sayfasi/");
            }
        }
        return null;
    }
};
post(girisFormPostAction);
}
```

Admin girişi olduğunda kullanıcı yönetici sayfasına yönlendirildi.

```
public class Anasayfa {
    public static void sayfalarıTanimla() { anaSayfaGET(); }

    private static void anaSayfaGET() {
        FreeMarkerRoute anaSayfa = new FreeMarkerRoute( path: "/onurkaral/" ) {
            @Override
            public Object handle(Request istek, Response cevap) {
                Kullanici kullanici = UygulamaMain.girisYapmisKullanici;
                if (kullanici == null) {
                    System.out.println("Once Giris Yapilmali");
                    cevap.redirect( location: "/onurkaral/giris_sayfasi/");
                }

                Map<String, Object> sayfaVerisi = new HashMap<>();
                sayfaVerisi.put("kullaniciaadi", kullanici.getKullaniciAdi());
                return new ModelAndView(sayfaVerisi, viewName: "/onurkaral/finalproje/anasayfa.html");
            }
        };
    }
    get(anaSayfa);
}
```

Anasayfa spark.java da tanıtıldı ve giriş yapılmamış ise tekrar giriş sayfasına yönlendirdi.

Eğer giriş yapmadan admin sayfasına giriş yapmak istenirse hata sayfasına yönlendirildi.

```
package spark.template.freemarker.onurkara1.finalproje;

import ...

public class HataSayfasi {
    public static void sayfalarıTanımla() { HataSayfaGET(); }

    private static void HataSayfaGET() {
        FreeMarkerRoute HataSayfa= new FreeMarkerRoute( path: "/onurkara1/hata_sayfasi/" ) {
            @Override
            public Object handle(Request istek, Response cevap) {

                return new ModelAndView( model: null, viewName: "/onurkara1/finalproje/hata_sayfasi.html");
            }
        };
        get(HataSayfa);
    }
}
```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Hata Sayfasi</title>
</head>
<body>
/ - - - - - /
<br></br>
HATA
<br></br>
Admin Paneline Giriş Yapmaya Çalıştınız.Lütfen Geri Donunuz.!
<br></br>
/ - - - - - /
</body>
</html>
```

```
private static void YoneticisiSayfaGET() {
    FreeMarkerRoute YoneticisiSayfa = new FreeMarkerRoute( path: "/onurkara1/yoneticisi_anasayfa/" ) {
        @Override
        public Object handle(Request istek, Response cevap) {

            Kullanici ADMIN = UygulamaMain.girisYapmisKullanici;

            if(ADMIN == null ) {

                cevap.redirect( location: "/onurkara1/hata_sayfasi/");
            }
        }
    };
}
```

Eğer admin ise bu sayfaya yönlendirildi:

```
package spark.template.freemarker.onurkaral.finalproje;
import static spark.Spark.get;
import java.util.HashMap;
import java.util.Map;

import spark.ModelAndView;
import spark.Request;
import spark.Response;
import spark.template.freemarker.FreeMarkerRoute;

public class YoneticisiAnasayfa {
    public static void sayfalarıTanımla() { YoneticisiSayfaGET(); }

    private static void YoneticisiSayfaGET() {
        FreeMarkerRoute YoneticisiSayfa = new FreeMarkerRoute( path: "/onurkaral/yoneticisi_anasayfa/" ) {
            @Override
            public Object handle(Request istek, Response cevap) {

                Kullanici ADMIN = UygulamaMain.girisYapmisKullanici;

                if(ADMIN == null ) {

                    cevap.redirect( location: "/onurkaral/hata_sayfasi/" );

                }

                Map<String, Object> sayfaVerisil = new HashMap<>();
                sayfaVerisil.put("kullanicitipi", ADMIN.getKullaniciTipi());
                return new ModelAndView(sayfaVerisil, viewName: "/onurkaral/finalproje/yoneticisi_anasayfa.html");
            }
        };
        get(YoneticisiSayfa);
    }
}
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Admin Page</title>
</head>
<body>
    <br></br>
    <h1>HOSGELDINIZ</h1>
    <h2>ADMIN PAGE</h2>
    Giris Yapan Kullanici: ${kullanicitipi}
    <h3>Siparisler</h3>
</body>
</html>
```

VeritabaniYonetimi.java doaysında derby veritabanı bağlantısı yapıldı.

```
package spark.template.freemarker.onurkaral.finalproje;
import java.awt.List;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;

public class VeritabaniYonetimi {
    private static Statement veritabaniStatement;
    public static Statement veritabaniBaglantisiKur() {
        try {
            Class.forName("org.apache.derby.jdbc.EmbeddedDriver").newInstance();
            Connection veritabaniBaglantisi = DriverManager.getConnection(
                urk: "jdbc:derby:"
                    + ".\\onurkaraldb\\"
                    + ";create=true;");
            veritabaniStatement = veritabaniBaglantisi.createStatement();
            return veritabaniStatement;
        } catch (Exception e) {
            System.out.println("Veritabanina Baglanirken Hata Olustu");
        }
        return null;
    }
}
```

Sonrasında Bir kere oluşturulacak olan “kullanıcılar, ürünler, siparişler “ tabloları oluşturuldu.

VeritabaniYonetimi.java kısmında 2 tür kullanıcı oluşturuldu: Guest ve admin.

```
public static Kullanici kullaniciSorgula(String kullaniciAdi) {
    try {
        ResultSet sonuclar = veritabaniStatement.executeQuery(
            sql: "SELECT * FROM kullanicilar WHERE kullanic_iadi = " + kullaniciAdi + ";"
        );
        while (sonuclar.next()) {
            String kullaniciTipi = sonuclar.getString( columnLabel: "kullanic_iadi");
            int yas = sonuclar.getInt( columnLabel: "yas");
            String cinsiyet = sonuclar.getString( columnLabel: "cinsiyet");
            String sifre = sonuclar.getString( columnLabel: "sifre");
            return new Kullanici(kullaniciAdi, kullaniciTipi, yas, cinsiyet, sifre);
        }
    } catch (Exception e) {
        System.out.println("Kullanici Sorgu Hatasi.!" + e.getMessage());
    }
    return null;
}

public static void kullanicilariOlustur() {
    if (kullaniciSorgula( kullaniciAdi: "guest") != null) {
        System.out.println("Sistem kullanicilari mevcut");
        return;
    }
    try {
        // "kullanicilar" tablosu: kullanic_iadi, kullanic_iadi, yas, cinsiyet, sifre
        veritabaniStatement.executeUpdate( sql: "INSERT INTO kullanicilar VALUES ( "
            + " 'guest', 'NORMAL', 22, 'E', 'abcd'"
            + ")");
        veritabaniStatement.executeUpdate( sql: "INSERT INTO kullanicilar VALUES ( "
            + " 'ADMIN', 'ADMIN', 22, 'K', 'xyz'"
            + ")");

        System.out.println("Sistem kullanicilari olusturuldu.");
    } catch (Exception e) {
        System.out.println("Ayni id'li kayıt olusturulamaz : " + e.getMessage());
    }
}
```

Daha sonrasında ise bunların sorgusu yapıldı.

Kullanıcıların temel bilgileri kullanıcı.java dosyasına kaydedildi.

```
package spark.template.freemarker.onurkaral.finalproje;

public class Kullanici {
    private String kullaniciAdi;
    private String kullaniciTipi;
    private int yas;
    private String cinsiyet;
    private String sifre;

    public Kullanici(String kullaniciAdi, String kullaniciTipi, int yas, String cinsiyet, String sifre) {
        this.kullaniciAdi = kullaniciAdi;
        this.kullaniciTipi = kullaniciTipi;
        this.yas = yas;
        this.cinsiyet = cinsiyet;
        this.sifre = sifre;
    }

    public String getKullaniciAdi() { return kullaniciAdi; }

    public String getKullaniciTipi() { return kullaniciTipi; }

    public int getYas() { return yas; }

    public String getCinsiyet() { return cinsiyet; }

    public String getSifre() { return sifre; }
}
```

Kullanıcıların üye oluşturabileceği üye sayfası.java uye_sayfası.html dosyaları oluşturuldu.

```
package spark.template.freemarker.onurkaral.finalproje;
import ...

public class UyeSayfasi {
    public static void sayfalarıTanimla() {
        UyeSayfasiTanimla();
        UyeFormTanimla();
    }

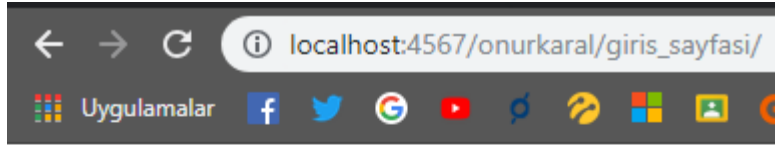
    private static void UyeFormTanimla() {
        Route girisFormPostAction = new Route( path: "/onurkaral/Uye_Sayfasi/create" ) {
            @Override
            public Object handle(Request istek, Response cevap) {
                String kullaniciadi = istek.queryParams("kullaniciadi");
                String sifre = istek.queryParams("sifre");

                return null;
            }
        };
        post(girisFormPostAction);
    }

    private static void UyeSayfasiTanimla(){
        FreeMarkerRoute UyeSayfasi = new FreeMarkerRoute( path: "/onurkaral/Uye_Sayfasi/" ) {
            @Override
            public Object handle(Request istek1, Response cevap1) {
                return new ModelAndView( model: null, viewName: "/onurkaral/finalproje/Uye_Sayfasi.html");
            }
        };
        get(UyeSayfasi);
    }
}
```

Bunlarda hepsi gibi sayfa Tanımla fonksiyonu ile Main dosyasına atandı.

Giriş Sayfası



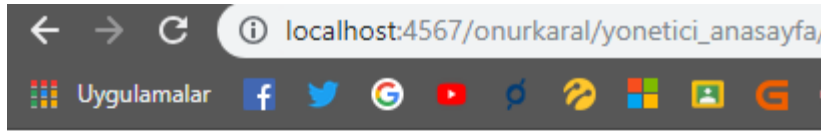
Hosgeldiniz

Kullanici adi:

Sifre:

[Kullanici Olustur](#)

Admin Paneli



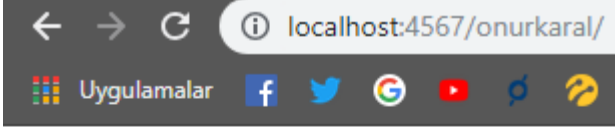
HOSGELDINIZ

ADMIN PAGE

Giris Yapan Kullanici: ADMIN

Siparisler

Normal Kullanıcı Paneli

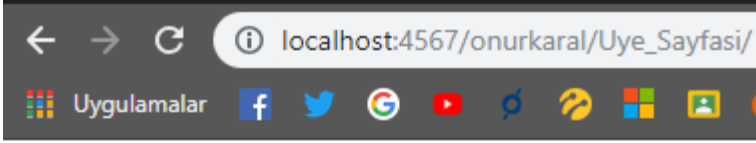


HOSGELDINIZ

ANASAYFA

Giris Yapan Kullanici: guest

Yeni Üyelik Sayfası



Uyelik

Kullanici adi:

Sifre:

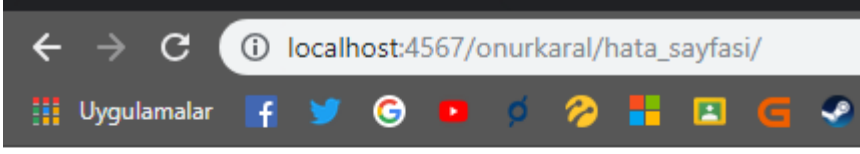
Yas:

Cinsiyet:

- ☐ Male
☐ Female
☐ Other

[Geri Don](#)

HATA SAYFASI



/-----/

HATA

Admin Paneline Giris Yapmaya Calistiniz.Lutfen Geri Donunuz.!

/-----/