

# Kurulum

Yazılım bileşenlerinde sürüm farklarından dolayı(sürümlerin eski veya yeni olması) sorun yaşanabiliyor.

## Java Sürümünün Kontrolü

`pom.xml` dosyasında belirtilen Java sürümü **17** olduğundan;

- Sistem ortam değişkeni olan `JAVA_HOME` da aynı sürümde bir Java kurulum klasör yolunu göstermelidir.

```
java -version
```

- IntelliJ IDEA Community Edition içinden çalıştırılacaksa, aşağıdaki menü yolu takip edilerek, sürümü **17** olan bir Java seçilmelidir:

*menü yolu*

```
IntelliJ IDEA
File
  Project Structure
    Project Settings
      Project
        SDK
```

## Dosyaları indirmek



Kaynak kodlar linkte herkese açık olmayan depoda bulunuyor olabilir. Dolayısıyla erişim yetkiniz yok ise bu belgede anlatılan kurulum işlemlerini yapmanız mümkün olmayabilir.

- Dosyaları depodan klonlamak için [git](#) kurulu olmalıdır. Kuruluysa boş bir klasörde konsoldan aşağıdaki komut çalıştırılır.

```
git clone https://github.com/turkerozturk/SweetCherrySync.git
```

- Veya dosya internet gezgini aracılığıyla <https://github.com/turkerozturk/SweetCherrySync/archive/refs/heads/main.zip> adresinden arşiv dosyası olarak indirilip içindeki dosyalar arşivden çıkarılır.

## Kaynak koddan derlemek

`SweetCherrySync` klasöründe konsol penceresi açılır. Aşağıdaki komut çalıştırılır.

windows

```
mvnw.cmd clean install
```

linux

```
./mvnw clean install
```



Bu komut, projenin `.mvn/wrapper/maven-wrapper.properties` dosyasında yazılı olan Maven sürümünü (sistemde bulamazsa, `KULLANICININ EV DİZİNİ/.m2/wrapper/dists` klasörüne indirdikten sonra temiz bir derleme işlemi yapar(varsa önceki derleme klasörünü kaldırır).



Linux işletim sisteminde `mvnw` komutunun çalışabilmesi için, bir kereliğine `chmod +x mvnw` komutu verilerek dosyanın çalıştırılabilir olduğunu belirtmek gerekir.

İşlem sonucunda derlenen proje, proje dizininde oluşan `target` dizininde JAR dosyası halinde bulunmaktadır.

JAR dosyası `SweetCherrySync-1.0.0.jar` adıyla oluşur.

## Çalıştırmadan Önce Port Numarası Kontrolü



Sonraki başlıktan devam edebilirsiniz. Uygulamanız çalışmazsa, bu başlığa geri dönüp port kontrolü ve gerekirse uygulama ayarı yapıp tekrar deneyebilirsiniz.

### Eğer port numarasını kullanan başka uygulama varsa

Kullanılmayan port numarası deneme yanılma metodu ile veya konsol komutu ile bulunabilir. Uygulamanın varsayılan olarak kullandığı portların boş olup olmadığını kontrol etmek için:

windows cmd

```
netstat -ano -p tcp |find "8081"
```

Komutlarının çıktısında,

TCP	0.0.0.0:{httpsportnumber}	0.0.0.0:0	LISTENING	PID
numarası				

ve

TCP	0.0.0.0:8081	0.0.0.0:0	LISTENING	PID	numarası
-----	--------------	-----------	-----------	-----	----------

görünüyorsa, bu durumda port doludur ve uygulamamızı çalıştırmak için ya o portu kullanan uygulamayı bulup sonlandırmak, veya uygulamamızın port numarasını değiştirmek gereklidir.



Linux için aynı işi gören komutlar;

```
sudo netstat -ano -p tcp
```

ve

```
ps -ef | grep PID numarası
```

## PID'den işlemi bulma

Eğer belirli bir PID (Process ID) üzerinden hangi uygulamanın bu portu kullandığını öğrenmek isterseniz, şu komutu kullanabilirsiniz:

```
tasklist | find "PID numarası"
```

Bu komut, ilgili işlemin adını gösterecektir.

Hiçbir çıktı göstermediyse port boşta demektir ve bu da uygulamamızın bu portu kullanarak sorunsuz çalışacağı anlamına gelir.

## Farklı Port Numaraları Kullanılmak İstenirse

Sistemde kullanılmayan farklı bir port numarasını `application.yml` dosyasında belirtmek gerekir.

- Proje, kaynak kodundan çalıştırılacaksa dosyanın konumu `PROJEKLASÖRÜ/src/main/resources/application.yml` olur.
- Proje, JAR dosyası olarak çalıştırılacaksa, JAR dosyası ile aynı klasörde bir `application.yml` dosyası oluşturulmalıdır. Satır başlarından itibaren boşluk karakterleri aynı olmalıdır.

*application.yml dosyası içinde olması gereken kısım*

```
server:  
  port: 8081 ①
```

① HTTP port numarası

## Projenin Çalıştırılması

### Kaynak Koddan Çalıştırmak

Derlenmemiş proje, proje klasörünün içinde bulunan `mvnw` dosyasının bulunduğu yerde konsol penceresinde aşağıdaki komutla çalıştırılır.

*windows cmd konsolunda*

```
mvnw.cmd spring-boot:run
```

linux terminalinde veya windows powershell konsolunda

```
./mvnw spring-boot:run
```

## Derlenmiş Dosyayı Çalıştırmak

Derlenmiş proje aşağıdaki komutla çalıştırılır.

windows cmd

```
java -jar `SweetCherrySync-1.0.0.jar`
```



Derlenmiş JAR dosyasını çalıştırırken dikkat edilmesi gereken, dosya hangi klasörün içindeyse konsolda da o klasördeyken çalıştırmaktır. Aksi halde uygulama çalışmaya başladıktan sonra kullanım esnasında ihtiyaç duyduğu dosya ve klasör konumlarında sorun yaşayacaktır.

## Web Gezgini İle Giriş Yapmak

Giriş <https://localhost:8081> sayfasından yapılır.

Websayfası sadece istatistik bilgi içerir, senkronizasyon için gerekli değildir, webserver özelliği devredışı bırakılabilir.

*application.yml dosyası içinde olması gereken kısım (enable disable embedded tomcat)*

```
spring:
  main:
    web-application-type: servlet ①
```

① Varsayılan değer **servlet**, devredışı bırakılmak istenirse **none**.

## Kaynaklar

- [Cloning a repository](#)
- [A Quick Guide to Maven Wrapper](#)
- [Determine Which Program Uses or Blocks a Port](#)

Author: [Türker Öztürk](#)