

# Kurulum

Yazılom bile!enlerinde sYrYm farklaröndan dolayı(sYrYmlerin eski veya yeni olmasö) sorun ya!anabiliyor.

## Java SYrYmYnYn KontrolY

**pom.xml** dosyasönda belirtilen Java sYrYmY **17** oldu"undan;

Y Sistem ortam de"!keni olan **JAVA\_HOME** da aynö sYrYmde bir Java kurulum klasr yolunu gštermelidir.

```
java -version
```

Y IntelliJ IDEA Community Edition i•inden •alö!törolacaksa, a!a"ödaki menY yolu takip edilerek, sYrYmY **17** olan bir Java se•ilmelidir:

*menY yolu*

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

## Dosyalarö indirmek



Kaynak kodlar linkte herkese a•ök olmayan depoda bulunuyor olabilir. Dolayösöyle erilim yetkiniz yok ise bu belgede anlatölan kurulum i!lemlerini yapmanöz mYmkYn olmayabilir.

Y Dosyalarö depodan klonlamak i•in **git** kurulu olmalödör. Kuruluysa bo! bir klasrde konsoldan a!a"ödaki komut •alö!törolör.

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

Y Veya dosya internet gezgini aracölö"öyle <https://github.com/turkerozturk/SweetCherrySync/archive/refs/heads/main.zip> adresinden ar!iv dosyasö olarak indirilip i•indeki dosyalar ar!ivden •ökarölör.

## Kaynak koddan derlemek

**SweetCherrySync** klasrYnde konsol penceresi a•ölör. A!a"ödaki komut •alö!törolör.

windows

```
mvnw.cmd clean install
```

linux

```
./mvnw clean install
```

!!

Bu komut, projenin `.mvn/wrapper/maven-wrapper.properties` dosyasında yazılmış olan Maven sürümünü (sistemde bulamazsa, `KULLANICI_NIN_EV_DIZINI/.m2/wrapper/dist` 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 oldu"unu belirtmek gerekir.

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

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

## , çalışmadan ...nce Port Numarası Kontrolü

!

Sonraki bölümden devam edebilirsiniz. Uygulamanız çalışmazsa, bu bölümü geri döndürüp port kontrolü ve gerekirse uygulama ayarları yapmayı 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"
```

Komutların 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
-----	---------------	-----------	-----------	--------------

görsünüyorsa, bu durumda port doludur ve uygulamamızın 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 çalışacağını anlamına gelir.

## Farklı Port Numaraları Kullanılmak İstenirse

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

¥ Proje, kaynak kodundan çalıştırılacaksa dosyanın konumu `PROJEKLER\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 "
```

" HTTP port numarası

## Projenin , çalıştırılması

### Kaynak Koddan , çalıştırmak

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

*windows cmd konsolunda*

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

linux terminalinde veya windows powershell konsolunda

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

## Derlenmi! Dosyayõ , alõ!tõrmak

Derlenmi! proje a!a"õdaki komutla •alõ!tõrõlõr.

windows cmd

```
Êjava -jar `SweetCherrySync-1.0.0.jar`
```

#

Derlenmi! JAR dosyasõnõ •alõ!tõrõrken dikkat edilmesi gereken, dosya hangi klasõrõn i•indeyse konsolda da o klasõrdeyken •alõ!tõrmaktõr. Aksi halde uygulama •alõ!maya ba!ladõktan sonra kullanõm esnasõnda ihtiya• duydu"u dosya ve klasõr konumlarõnda sorun ya!ayacaktır.

## Web Gezgini #le Giri! Yapmak

Giri! <https://localhost:8081> sayfasõndan yapõlõr.

Websayfasõ sadece istatistik bilgi i•erir, senkronizasyon i•in gerekli de"ildir, webserver õzelli"i devredõ!õ bõrakõlabilir.

*application.yml dosyasõ i•inde olmasõ gereken kãsõm (enable disable embedded tomcat)*

```
spring:
  Ê main:
  Ê web-application-type: servlet "
```

" Varsayõlan de"er **servlet**, devredõ!õ bõrakõlmak istenirse **none**.

## Kaynaklar

¥ [Cloning a repository](#)

¥ [A Quick Guide to Maven Wrapper](#)

¥ [Determine Which Program Uses or Blocks a Port](#)

Author: [TÕrker ...ztÕrk](#)