# SPRİNG BOOT DERS NOTLARI

# <https://start.spring.io/>

# Yukarıdaki siteye girince projemize eklemek istediğimiz özellikleri tek seferde seçip projeyi kolaylık yapmamıza olanak tanır.Mesela projemizde Jpa,Web,Security kullanacağız,üstteki siteye girip ‘add dependencies’ e tıklayıp web security jpa ekliyoruz.Yine buradan projenin adını artifact ID sini,group ID sini belirleyip hangi programlama dili ve maven mi gradle mi olduğunu seçebiliyoruz.Daha sonra ‘generate’ye tıklayıp,indiriyoruz.Mavenin jpanın securitynin webin hazır olduğu projeyi import ediyoruz.(Normal import değil,importa basınca oradan existing maven project’i bulmalıyız.Daha sonra projemizi açıyoruz.Ve artık hibernate,web gibi bağımlılıkları projemize eklememize gerek kalmıyor.Herşey pom.xml’de hazır.

# ÖNEMLİ OLAYLAR 1=Diyelim ki projede hibernate kullanmicaz ama yinede spring boot’ta jpa eklemişsek eğer aşağıdaki hatayı alırız o yüzden kullanmicamız bağımlılıkları eklemek yok.!

# HATA

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**\****

***APPLICATION FAILED TO START***

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

***Description:***

***Failed to configure a DataSource: 'url' attribute is not specified and no embedded datasource could be configured.***

***Reason: Failed to determine a suitable driver class***

***Action:***

***Consider the following:***

***If you want an embedded database (H2, HSQL or Derby), please put it on the classpath.***

***If you have database settings to be loaded from a particular profile you may need to activate it (no profiles are currently active).***

# ÖNEMLİ OLAYLAR 2=Spring Boot’la beraber Tomcat bağlantısı otomatik iniyor.Eğer daha önceden tomcat pc’de yüklü ise onu durdurmalıyız.Oda şöyle öncelikle aşağıda ‘Console’ nin yanında ‘Servers’ diye bi menü olmalı eğer yoksada kendin bul.Oradan tomcati kaldırıyoruz.Daha sonra başlat menüsüne hizmetler yada services.msc yazıyoruz oradan da Apache Tomcati bulup durduruyoruz.

# ÖNEMLİ OLAYLAR 3=Spring Boot projemizi yukardan falan çalıştırmıyoruz.Sol tarafta SpringBootDemoApplication’nun içinde olduğu paketi sağ tıklıyoruz oradan Run’a basıp çalıştırıyoruz.

# ÖNEMLİ OLAYLAR 4=Yaptığımız uygulamayı çalıştırdık diyelim daha sonra kodlara gelip bir şey değiştirdikten sonra bir daha çalıştırıcağız bunun için ilk önce consolu durdurmalıyız.

# ÖNEMLİ OLAYLAR 5=Pom.xml’e aşağıdaki kodlarla DevTools eklemeliyiz.Nedir bu DevTools? Bu eklentiyi projeye ekledikten sonra kodlara geldiğimiz ctrl+s yapınca yani kaydedince direk bir daha pakete gelip run yapmamıza gerek kalmıyor.Sen kodları değiştirip ctrl+s yapınca otomatik sitede değişiyor.Zaten kaydedince console’da çalışacak görürsün.Ayrıca bu DevTools sayesinde madde 4’e ve madde 3’e artık gerek yok.

# ÖNEMLİ OLAYLAR 6=Eğer uygulamayı çalıştırınca localhost:8080 e gidince Whitelabel Error Page diye bir hata alırsak bu şu anlama geliyor.Her localhostu sadece birkez kullanabiliriz ve bunu daha önce kullanmışız artık silmeliyiz oradaki verileri.Peki nasıl silicez? Şöyle:

# 1-İlk önce başlattan cmd diğer adıyla komut istemini çalıştırıyoruz.Oraya şunu yazıyoruz.

netstat -aon | findstr 8080

# 2-Daha sonra bir menü gibi bir şey çıkacak onun sağında listening var ve karşısında bir port numarası var.O port numarasıyla beraber aşağıdaki kodu yazıyoruz.

taskkill /PID 1 /F -> 1’yerine o gördüğümüz listening port numarasını yazıyoruz.

# 3-Success diye yanıt almadıysak şu adrese git

# <https://stackoverflow.com/questions/39632667/how-do-i-kill-the-process-currently-using-a-port-on-localhost-in-windows>

# ÖNEMLİ OLAYLAR 7= Actuator: Spring boot actuator'ı projemizin pom.xml’ine dependency olarak ekleyerek, projemizin çalışma durumu(ayakta/ayakta değil), trafik durumu, son yüz http isteği, veritabanımızın durumu vs gibi bilgilere çok kolay bir şekilde endpointler üzerinden erişebiliriz.Dependency ekledikten sonra localhost:8080/actuator/health vs gibi aşağıdaki gördüğün kodları yazarak bilgilere ulaşabilirsin.

Bu actuatorü güvenlik açısından kapatman gerekebilir.Kapatma işlemini ise src/main/resources pathin altına ekleyeceğin application.properties dosyasına şu kodu yazarak yapabilirsin. = management.endpoints.enabled-by-default=false

Actuator/kodları

/ autoconfig : tüm auto-configuration tanımlamalar

/ beans : Spring tarafından yönetilen tüm nesneler

/ configprops : @ConfigurationProperties anotasyonuna sahip tüm sınıflar

/ dump : thread dump almayı sağlar

/ env : Spring’ in ConfigurableEnvironment değerlerini listeler

/ flyway : uygulanan herhangi bir Flyway veritabanı taşıma işlemini gösterir

/ health : uygulamanın durumu hakkında genel bilgi verir (disk veya database erişim durumu vs.)

/ info : tanımlı uygulama bilgilerini gösterir

/ loggers : tanımlı sınıfların log seviyeleri

/ liquibase : uygulanan herhangi bir Liquibase veritabanı taşıma işlemini gösterir

/ metrics : mevcut uyguluma için ölçüm verilerini gösterir

/ mappings : @RequestMapping tanımlamalarını gösterir

/ shutdown : uygulamayı kapatabilmeyi sağlar (varsayılan değeri false)

/ trace : en son 100 HTTP istek detayını listeler

MVC projeleri için ekstra;

/ docs : classpath‘ te spring-boot-actuator-docs tanımlı ise Actuator lerle alakalı döküman ve örnek istek-sonuçları gösterir

/ heapdump : sıkıştırılmış olarak hprof heap dump dosyası döner

/ jolokia : Jolokia classpath‘ te tanımlı ise JMX bean lerini HTTP üzerinden gösterir

/ logfile : logging.file veya logging.path değerleri tanımlı ise mevcut log dosyası içeriğini gösterir

# SPRİNG BOOT KODLARI