

Java Uygulamasından MSSQL Server'a Bağlanmak

Class.forName Metodu

```
Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
```

Class bir Class ismidir. `forName()` metodunun bu şekilde kullanılmasının nedeni Class isimli class'tan **object** yaratılmamasıdır. Çünkü Class class'ı **private** 'dir.

Java da bir driver yüklenirken, **Class** isimli **class** kullanılmaktadır. Dediğimiz gibi bu Class'tan bir **object** yaratılmaz. Bu nedenle ilgili metodu yani `forName()` metodunu **static** metod şeklinde çağırıp ilgili driver ismini parametre olarak yazacağız.

Class.forName(Class) bir class'ın **başlatılmasını** sağlar. Bunu bir applet'in veya application'un yüklenmesi gibi düşünebiliriz. Yani jvm'nin yaptığı işi biz yapıyoruz. Örneğin database'e bağlantı kuran bir sürücü class'ı yükliyoruz ve Driver hafızaya yükleniyor. Bir class'ın nesnesini yaratmaktan farklıdır.

SQL Server Driver Neden Gereklidir ?

Veritabanı ile Java arasında bir nevi iletişim köprüsü oluşturur. Bu köprü'nün kurulabilmesi için öncelikle Driver yüklemek olarak tabir edilen işlemi gerçekleştirmek gerekir.

Bazı Örnekler :

```
com.microsoft.sqlserver.jdbc.SQLServerDriver
```

```
oracle.jdbc.driver.OracleDriver
```

Connection Class'ı

```
1 String connectionUrl = "jdbc:sqlserver://localhost:1433;databaseName=OnlineBookSale;user=john;password=123;";
2 Connection con = DriverManager.getConnection(connectionUrl);
```

Connection class'ından con isimli bir **object** yaratılırken **DriverManager** class'ının `getConnection()` isimli **static** metodu kullanılır.

Statement Class'ı ve ResultSet Interface'i

```
1 String query="select * from tblUsers where UserName like '%a%'";
2 Statement statement = con.createStatement();
3 ResultSet resultset = statement.executeQuery(query);
```

ResultSet interface'i türüne sahip olan resultset object'i Statement class'ından yaratılan statement object'inden türetilir.

ResultSet Türündeki resultset Object'inin Metodları

```
1 resultset.next();//Sql de dönen tablonun ilk satırındaki verileri dönderir. Yani Sql tablosundan satır satır okumayı sağlar.
2 while(resultset.next()){ info=resultset.getString(1);//info bir String objecti
3 }
4 //Metodu 1,2,3.. gibi sutun numaraların parametre olarak kullanır ve ilgili sütun numarasına göre string dönderir.
5 resultset.getString();
```

Örnek Proje

```
1 String queryStatement="select * from tblUsers where UserName like '%a%'";
2 int columnNumber=1;
3 String info=" ";
4 Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");//SqlServerDriver
5 String connectionUrl = "jdbc:sqlserver://localhost:1433;databaseName=OnlineBookSale;user=john;password=123;";
6 Connection con = DriverManager.getConnection(connectionUrl);
7 Statement statement = con.createStatement();//bir tane statement yaratılır.
8 ResultSet resultset = statement.executeQuery(queryStatement);
9 while(resultset.next()){
10     info=resultset.getString(1);
11 }
12 System.out.println(info);
13 con.close();//Asla unutma
```

Maven Projesine SQL JDBC Kütüphanesini Dahil Etmek

Adım 1: İlk önce <http://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=11774> adresinden Microsoft **JDBC Driver** indirilir

Adım 2: İndirilen dosya çıkartılır

Adım 3: Çıkartılan dosyanın bulunduğu dizine giderek **sqljdbc4.jar** isimli jar dosyası **local repository**'e **import** edilir. Bu işlem için aşağıdaki kodu Windows konsol(command prompt) ekranına açıp yapıştırınız:

```
1 mvn install:install-file -Dfile=sqljdbc4.jar -DgroupId=com.microsoft.sqlserver -DartifactId=sqljdbc4 -Dversion=3.0 -Dpackagin
```

Not: Windows Konsol Ekranı sqljdbc4.jar dosyasının bulunduğu dizinde açılmalıdır

Adım 4: pom.xml dosyasına aşağıdaki **dependency** ekleyiniz:

```
1 <dependency>
2 <groupId>com.microsoft.sqlserver</groupId>
3 <artifactId>sqljdbc4</artifactId>
4 <version>3.0</version>
5 </dependency>
```