Connecting to a Database



Kevin Jones

@kevinrjones



What's in This Module



JDBC URL

Initial connection to database



String url = "jdbc:mysql://localhost:3306/loboticket"

Connection conn = DriverManager.getConnection(url);

Getting the Connection

To get a connection need to use the DriverManager

The URL identifies the database to connect to and is database specific

JDBC URLs

jdbc:mysql://localhost:3306/loboticket

JDBC URLs

jdbc:mysql://localhost:3306/loboticket

JDBC URLs

Database specific connection string

jdbc:mysql://localhost:3306/loboticket

Subname

Example JDBC URLs

jdbc:postgresql://localhost/loboticket

jdbc:mysql://localhost:3306/loboticket

jdbc:oracle:thin:@123.123.123.123:5321:loboticket

jdbc:derby:loboticket



Connecting to a Database



Connecting with a User Name and Password

Running the Code

Jar file containing driver

java -cp ::mysql-connector-java-8.0.23.jar com.app.Application

Output

com.mysql.cj.jdbc.ConnectionImpl@2ea41516

Notice that it prints the driver specific class name



Demo



DriverManager.getConnection



Database Connections

These are native resources

Must be closed by the application after use

Use 'try with resources' to close connection

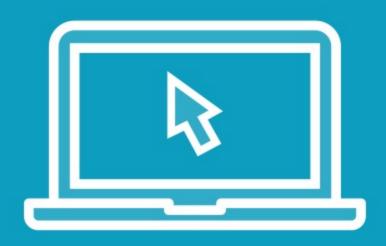
This will also close the statement and the result set



Using Try With Resources

```
public static void main(String[] args) throws SQLException {
  try (Connection conn = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/loboticket")) {
     PreparedStatement stmt = conn.prepareStatement("");
     ResultSet rs = stmt.executeQuery();
    while(rs.next()) {
       System.out.println(rs.getString("Name"));
```

Demo



try..with resources



Summary



JDBC URL has a specific format

Use the URL to load the driver

Driver connects to the database

Use try with resources to close connection, prepared statement and result set



Up Next: Using PreparedStatements

