

JSP – JavaBEAN – MySQL 1.

1. Adatbázis tábla importálás

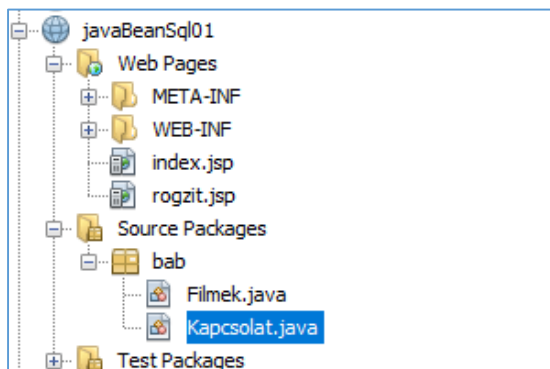
Adott egy MySQL-tábla, a **filmek**:

| Név | Típus | Illesztés | Tulajdonságok | Nulla | Alapértelmezett | Megjegyzések | Extra |
|------------|--------------|-------------------|---------------|-------|-----------------|--------------|----------------|
| id | int(11) | | | Nem | Nincs | | AUTO_INCREMENT |
| cim | varchar(200) | utf8_hungarian_ci | | Igen | NULL | | |
| ev | int(11) | | | Igen | NULL | | |
| mufaj | varchar(30) | utf8_hungarian_ci | | Igen | NULL | | |
| nemzetiseg | varchar(10) | utf8_hungarian_ci | | Igen | NULL | | |
| ido | int(11) | | | Igen | NULL | | |
| jegyjar | int(11) | | | Igen | NULL | | |

A tábla tartalmaz néhány adatot, importáljuk a szerkezetet és a tartalmat a **filmek.sql** állományból!

2. Hozzunk létre egy Java Web projektet!

3. Alkossuk meg az osztályt **Kapcsolat néven** az adatbázishoz kapcsolódás végrehajtásához és ellenőrzéséhez!



```
package bab;

import java.sql.Statement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class Kapcsolat {

    private Connection con;
    private Statement stm;
    private String uzenet;

    public Kapcsolat() {
    }

    public Statement getStm() {
        return stm;
    }
}
```

```

public boolean connOK()
{
    try
    {
        Class.forName("com.mysql.jdbc.Driver");
        con = DriverManager.getConnection("jdbc:mysql://localhost:3306/filmek","root","root");
        stm = con.createStatement();
        return true;
    }
    catch (ClassNotFoundException | SQLException e)
    {
        uzenet = e.toString();
        return false;
    }
}

public String uzenet()
{
    if(connOK())
        return "Kapcsolódás rendben";
    else
        return "Kapcsolódási HIBA <br />" + uzenet;
}
}

```

- a projekthez hozzá kell adni a MySQL-Library-t, és a Class.forName az ebben lévő Driver osztályra hivatkozik!
- a kapcsolódás menete a Java SE-ben megszokott, csak összevontam egy sorba
- a **connOK()** metódus kapcsolódik, ha tud, ha nem tud, akkor hibaüzenettel tér vissza, amúgy logikai típusú, hogy később ellenőrzésre tudjuk használni.
- az **uzenet()** metódus ellenőrzi az előbbi visszatérési értékét és adja az üzenet szövegét.

4. Próbáljuk ki, az **index.jsp**-ben csatlakozunk a kiszolgálóhoz!

Az ábrán a forráskód első fele látható, a vége a szokásos – egyelőre...

```

<%@ page contentType="text/html" pageEncoding="UTF-8" language="java" import="java.sql.*"%>
<%@ page import = "bab.Kapcsolat" %>
<%@ page import = "bab.Filmek" %>

<jsp:useBean id="dbKapcs" class="bab.Kapcsolat" scope="session"/>
<jsp:useBean id="filmes" class="bab.Filmek" scope="session"/>
<!--jsp:setProperty name="teglalap" property="**"/-->

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
<title>JSP Page</title>
</head>
<body>
<div class="container">
<div>
<h1>Hello MySQL!</h1>
<%=dbKapcs.uzenet() %>
<br />
<a href="index.jsp">Adatok listázása</a> ||
<a href="rogzit.jsp">Adatok rögzítése</a> ||
</div>

```

A korábban megtanult bab-kapcsolat, a filmek-re hivatkozó még nem kell!

Hello MySQL!

Kapcsolódás rendben

[Adatok listázása](#) || [Adatok rögzítése](#) ||

5. Hozzuk létre a filmek tábla kezelésére alkalmas osztályt **Filmek** néven!

(a használatához majd kell az `index.jsp` elején a hivatkozás!!!)

```
package bab;

import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Filmek {

    private int id;
    private String cim;
    private int ev;
    private String mufaj;

    private final Kapcsolat kapcs;
    private String uzenet;
    private Statement st;
    private ResultSet rs;

    public Filmek() {
        this.kapcs = new Kapcsolat();
    }
}
```

Csak 4 mezőt kezelek, bővítsük ki a többivel!

```
public int getId() { ...3 lines }
```

```
public void setId(int id) { ...3 lines }
```

```
public String getCim() { ...3 lines }
```

```
public void setCim(String cim) { ...3 lines }
```

```
public int getEv() { ...3 lines }
```

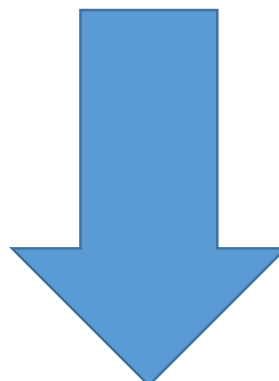
```
public void setEv(int ev) { ...3 lines }
```

```
public String getMufaj() { ...3 lines }
```

```
public void setMufaj(String mufaj) { ...3 lines }
```

*Getterek és Setterek
összezsukva!*

Írjunk egy olyan metódust, ami az osztályon belül szedi össze az adatokat és egy Stringként kérdezhető le! A végeredmény tekintetében nem a legjobb most, de nem baj, ha látunk ilyet is...



```

public String listaz()
{
    String kiir = "";
    try
    {
        if(kapcs.connOK())
        {
            st = kapcs.getStm();
            rs = st.executeQuery("select * from filmek");
            while(rs.next())
            {
                kiir += rs.getInt("id") + " ";
                kiir += rs.getString("cim") + " ";
                kiir += rs.getInt("ev") + " ";
                kiir += rs.getString("mufaj");
                kiir += "<br />";
            }
        }
    }
    catch (SQLException e)
    {
        kiir += "HIBA a kiolvasásakor!" + e;
    }
    return kiir;
}

```

Teljesen megegyezik a Java SE-ben tanultakkal!

Folytatva az **index.jsp** állományt, hívjuk meg a bean listaz() metódusát:

```

<div>
    <h1>Listázás a Bean-ből</h1>
    <%=filmes.listaz() %>
</div>

```

Listázás a Bean-ből

- 1 Az ördög maga 1998 Akció
- 2 Hyppolit a lakáj 1936 Vígjáték
- 3 Titanic 1998 Dráma
- 4 Piedone 1978 Vígjáték
- 5 Csínibaba 1995 Vígjáték
- 6 Lesz ez még így se! 1997 Vígjáték

NEM SZÉP, NEM FORMÁZOTT, DE OTT VAN MINDEN ADAT... ☺

6. Készítsük el más módszerrel! Hozzunk létre 2 új metódust a **Filmek.java**-ban!

```
public void listaz2()
{
    try
    {
        rs = st.executeQuery("select * from filmek");
    }
    catch (SQLException e)
    {
        System.out.println(e);
    }
}

public ResultSet listaLeker() {
    return rs;
}
```

Az **index.jsp** pedig fel fogja használni:

```
<div>
    <h1>A listázás másként</h1>
    <table class="table table-hover">
        <tr>
            <td>id</td><td>cim</td><td>ev</td><td>mufaj</td>
        </tr>
        <%
            try
            {
                filmes.listaz2();
                ResultSet temp = filmes.listaLeker(); //lsd. 1. sor import!
                while (temp.next())
                {
                    <tr>
                        <td><%out.print(temp.getInt("id")); %></td>
                        <td><%out.print(temp.getString("cim")); %></td>
                        <td><%out.print(temp.getInt("ev")); %></td>
                        <td><%out.print(temp.getString("mufaj")); %></td>
                    </tr>
                }
            }
            catch (Exception e)
            {
                out.print("HIBA " + e);
            }
        %>
    </table>
</div>
```

A listázás másként

| id | cim | ev | mufaj |
|----|------------------|------|----------|
| 1 | Az ördög maga | 1998 | Akció |
| 2 | Hyppolit a lakáj | 1936 | Vígjáték |

7. Készítsünk új metódust az adatok rögzítéséhez! Egészítsük ki a **Filmek.java**-t!

```
public void beszur()
{
    try
    {
        String query = "insert into filmek(cim,ev,mufaj) values('"
            +cim+"', '"+ev+"', '"+mufaj+"')";
        st.executeUpdate(query);
        System.out.println("A beszúrás sikeres!");
    }
    catch (SQLException e)
    {
        System.out.println("Error in inserting: " + e.getMessage());
    }
}
```

Az új adatok rögzítéséhez készítsünk egy új jsp állományt **rogzit.jsp** néven!

```
<%@ page contentType="text/html" pageEncoding="UTF-8" language="java" import="java.sql.*"%>
<%@ page import = "bab.Kapcsolat" %>
<%@ page import = "bab.Filmek" %>

<jsp:useBean id="dbKaps" class="bab.Kapcsolat" scope="session"/>
<jsp:useBean id="filmes" class="bab.Filmek" scope="session"/>
<jsp:setProperty name="filmes" property="*" />

<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
    <title>Adatrögzítés</title>
</head>

<body>
    <div class="container"><div>
        <h1>Új filmek felvétele</h1>
        <a href="index.jsp">Adatok listázása</a> ||
        <a href="rogzit.jsp">Adatok rögzítése</a> ||
    </div>
    <div class="col-md-6">
        <form name="rogzit" class="form-row" method="POST" action="rogzit.jsp">
            <div class="form-group col-md-12">
                <label for="cim">Cim: </label>
                <input type="text" name="cim" class="form-control">
            </div>
            <div class="form-group col-md-12">
                <label for="ev">Év: </label>
                <input type="text" name="ev" class="form-control">
            </div>
            <div class="form-group col-md-12">
                <label for="mufaj">Műfaj: </label>
                <input type="text" name="mufaj" class="form-control">
            </div>
            <div class="form-group col-md-12">
                <input type="submit" name="kuld" value="Adatok rögzítése" class="btn btn-success">
            </div>
        </form>
    </div>
```

```

<div class="col-md-12">
    <%
        try
        {
            if(request.getParameter("kuld") != null)
            {
                filmes.beszur();
                out.println("Sikeres rögzítés...");
            }
        }
        catch (Exception e)
        {
            out.println("Be kéne írni valamit, Haver...");
        }
    %>
</div>
</div>
</body>
</html>

```

Új filmek felvétele

[Adatok listázása](#) || [Adatok rögzítése](#) ||

Cím:

Év:

Műfaj:

[Adatok rögzítése](#)

FELADATOK!

- Minden feladatrészt alkalmazzunk a teljes táblára!
- Formázzuk meg a weboldalt tetszőlegesen (de bármennyire is tetszik így, ne hagyjuk a leírás kinézetében ☺)
- Készítsünk menüsort, ami nem ismétlődjön minden állomány elején, alkalmazzuk az include lehetőségét!