Day - 12
Servlet Config and Context

Previously we had seen how to display all the drives which the student was eligible for, but we can optimize the approach which we had previously followed.
Every servlet, if it wants to connect to the database, then it should have configuration information(URL, username, and password) inside it and these configurations are present in all of our 3 servlets i.e., **Validation.java, Eligible.java, and Drives.java**
But if we change anything in the configuration, then we have to update it in all 3 servlets which are not so efficient and maintainability becomes an issue.
So how do we resolve this?

The solution lies in the **Deployment Descriptor(web.xml)**.

We can add the configuration in the servlet inside the servlet tag in web.xml by making use of the ***<init-param>*** tag
The <init-param> tag has two child tags-

- ➔ ***<param-name>***
- ➔ ***<param-value>***

Which is used to save the initialization parameters

Type in the following inside web.xml

```
<servlet>
	<servlet-name>Login</servlet-name>
	<servlet-class>com.tap.student.Validation</servlet-class>

	<init-param>
		<param-name>url</param-name>
		<param-value>jdbc:mysql://localhost:3306/tapacademy</param-value>
	</init-param>

	<init-param>
		<param-name>username</param-name>
		<param-value>root</param-value>
	</init-param>

	<init-param>
		<param-name>password</param-name>
		<param-value>root</param-value>
	</init-param>

</servlet>

<servlet-mapping>
	<servlet-name>Login</servlet-name>
	<url-pattern>/login</url-pattern>
</servlet-mapping>
```

```
<servlet>
	<servlet-name>Drive</servlet-name>
	<servlet-class>com.tap.student.Drive</servlet-class>

	<init-param>
		<param-name>url</param-name>
		<param-value>jdbc:mysql://localhost:3306/tapacademy</param-value>
	</init-param>

	<init-param>
		<param-name>username</param-name>
		<param-value>root</param-value>
	</init-param>

	<init-param>
		<param-name>password</param-name>
		<param-value>root</param-value>
	</init-param>

</servlet>

<servlet-mapping>
	<servlet-name>Drive</servlet-name>
	<url-pattern>/drive</url-pattern>
</servlet-mapping>

<servlet>
	<servlet-name>Eligible</servlet-name>
	<servlet-class>com.tap.student.Eligible</servlet-class>

	<init-param>
		<param-name>url</param-name>
		<param-value>jdbc:mysql://localhost:3306/tapacademy</param-value>
	</init-param>

	<init-param>
		<param-name>username</param-name>
		<param-value>root</param-value>
	</init-param>

	<init-param>
		<param-name>password</param-name>
		<param-value>root</param-value>
	</init-param>

</servlet>
```

```
<servlet-mapping>
      <servlet-name>Eligible</servlet-name>
      <url-pattern>/eligible</url-pattern>
</servlet-mapping>
```

Whenever a servlet is loaded, a **ServletConfig object** is created and all the initialization parameters which we have mentioned inside the **web.xml** will be initialized as instance variables inside the object and these instance variables can be accessed by using the overloaded init( ), which is **init(ServletConfig sc)**.
To access the contents inside the object we can use the following methods-

- ➔ **getInitParameter(String name)**
- ➔ **getInitParameterNames( )**

Now let us see how to access the instance variables present inside the **ServletConfig** object.
Type in the following-

Validation.java

```
public class Validation extends HttpServlet {
      Connection con = null;
      PreparedStatement pstmt = null;
      ResultSet res = null;
      String url = null;
      String un = null;
      String pwd = null;

      @Override
      public void init(ServletConfig sc) throws ServletException {

            url = sc.getInitParameter("url");
            un = sc.getInitParameter("username");
            pwd = sc.getInitParameter("password");
```

Eligible.java

```
public class Eligible extends HttpServlet {
      Connection con = null;
      String url = null;
      String un = null;
      String pwd = null;

      @Override
      public void init(ServletConfig sc) throws ServletException {

            url = sc.getInitParameter("url");
            un = sc.getInitParameter("username");
            pwd = sc.getInitParameter("password");
```

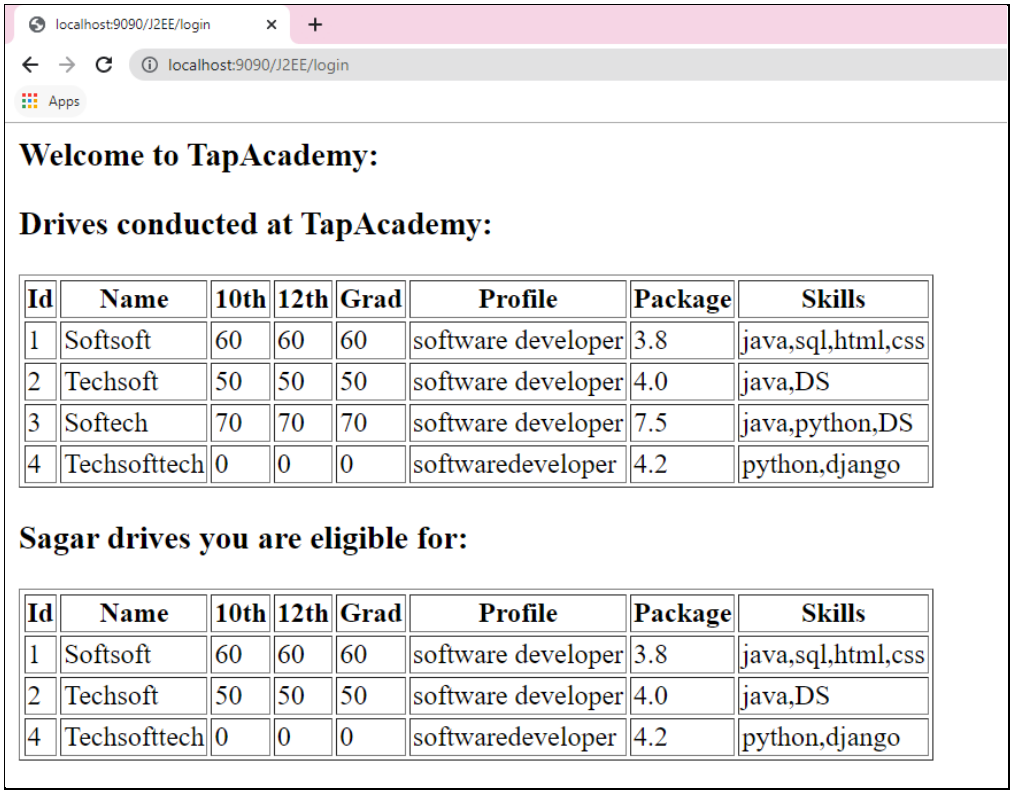
Drive.java

```
public class Drive extends HttpServlet {
	Connection con = null;
	PreparedStatement pstmt = null;
	ResultSet res = null;
	String url = null;
	String un = null;
	String pwd = null;

	@Override
	public void init(ServletConfig sc) throws ServletException {

		url = sc.getInitParameter("url");
		un = sc.getInitParameter("username");
		pwd = sc.getInitParameter("password");
```

And upon execution, we will get the following output-



Id	Name	10th	12th	Grad	Profile	Package	Skills
1	Softsoft	60	60	60	software developer	3.8	java,sql,html,css
2	Techsoft	50	50	50	software developer	4.0	java,DS
3	Softech	70	70	70	software developer	7.5	java,python,DS
4	Techsofttech	0	0	0	softwaredeveloper	4.2	python,django

Id	Name	10th	12th	Grad	Profile	Package	Skills
1	Softsoft	60	60	60	software developer	3.8	java,sql,html,css
2	Techsoft	50	50	50	software developer	4.0	java,DS
4	Techsofttech	0	0	0	softwaredeveloper	4.2	python,django

As we can see, we get the exact same output but we have optimized the code in such a way that configuration information is present inside the web.xml and it is now easy to maintain.

Whenever different servlets are connecting to different databases, then their configuration settings will be different, but in our case, all 3 servlets are connecting to the same database **tapacademy**, instead of using

ServletConfig, we can use **ServletContext object** and This ServletContext object is common to all the servlets present inside the JEE project and this object is created by the web container **Catalina**.
Let us now see how to create a **ServletContext object**
Modify the web.xml as shown below-

1. Delete all the <init-param> tags previously added in web.xml
2. Add the following outside of servlet tags-

```
<context-param>
	<param-name>url</param-name>
	<param-value>jdbc:mysql://localhost:3306/tapacademy</param-value>
</context-param>

<context-param>
	<param-name>username</param-name>
	<param-value>root</param-value>
</context-param>

<context-param>
	<param-name>password</param-name>
	<param-value>root</param-value>
</context-param>
```

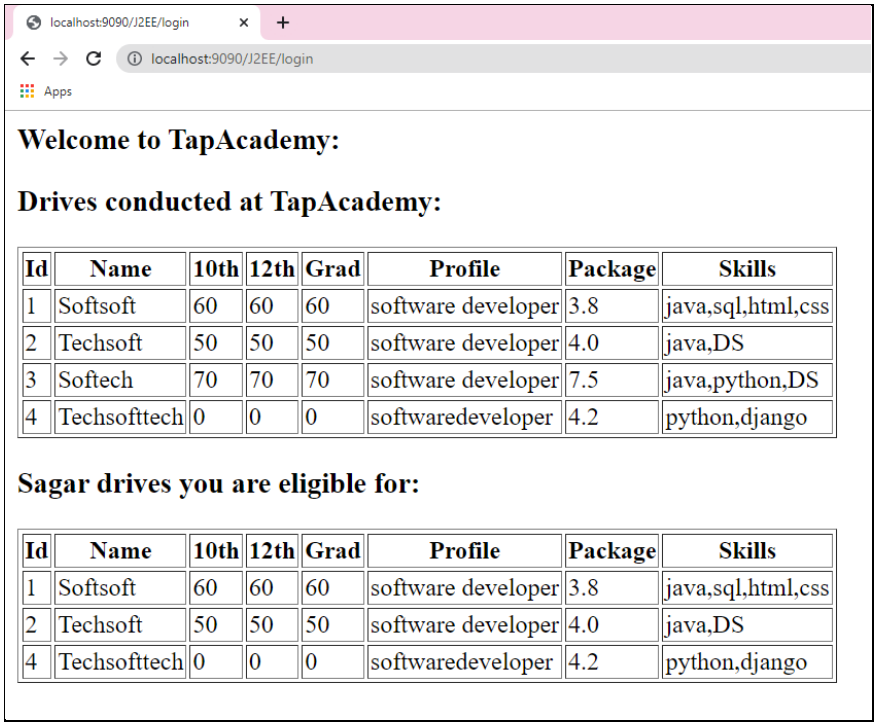
The ServletContext object is automatically created and we must just access it and now let us see how to access the instance variables inside the ServletContext object.
Make the following changes in all 3 servlets-

```
@Override
	public void init(ServletConfig sc) throws ServletException {

		ServletContext sCon = sc.getServletContext();

		url = sCon.getInitParameter("url");
		un = sCon.getInitParameter("username");
		pwd = sCon.getInitParameter("password");
```

Upon execution, we will get the following output-



Welcome to TapAcademy:

Drives conducted at TapAcademy:

Id	Name	10th	12th	Grad	Profile	Package	Skills
1	Softsoft	60	60	60	software developer	3.8	java,sql,html,css
2	Techsoft	50	50	50	software developer	4.0	java,DS
3	Softech	70	70	70	software developer	7.5	java,python,DS
4	Techsofttech	0	0	0	softwaredeveloper	4.2	python,django

Sagar drives you are eligible for:

Id	Name	10th	12th	Grad	Profile	Package	Skills
1	Softsoft	60	60	60	software developer	3.8	java,sql,html,css
2	Techsoft	50	50	50	software developer	4.0	java,DS
4	Techsofttech	0	0	0	softwaredeveloper	4.2	python,django