PreparedStatements
+ eficiencia + seguridad

ÍNDICE DE CONTENIDOS

Clases de la API

Clase	Descripción
DriverManager	Para cargar un driver
Connection	Para establecer conexiones con las bases de datos
Statement	Para ejecutar sentencias SQL y enviarlas a las BBDD
PreparedStatement	La ruta de ejecución está predeterminada en el servidor de base de datos que le permite ser ejecutado varias veces
ResultSet	Para almacenar el resultado de la consulta

PREPARED STATEMENTS

- Otra forma de realizar consultas
- Compilación previa (consume recursos inicialmente)
- Posteriormente es más eficiente
- Parametrizable
- Evita ataques SQL injection

PREPARED STATEMENTS

1. Defino la sentencia en una cadena

```
String sql = “SELECT * FROM customers
WHERE customername=? and country=?”;
```

2. Creo la PreparedStatement

```
Sentencia = connection.prepareStatement(sql);
```

PREPARED STATEMENTS

1. **Establezco los parámetros (Nombre y País)**

```
Sentencia.setString(1, “La Rochelle Gifts”);
Sentencia.setString(2, “France”);
```

2. **Ejecuto la sentencia**

```
ResultSet rs = sentencia.executeQuery();
o también .executeUpdate() <- Insert/Update/Delete
```

SQL Injection

> Inyección SQL es un método de infiltración de código intruso que se vale de una vulnerabilidad informática presente en una aplicación en el nivel de validación de las entradas para realizar operaciones sobre una base de datos

SQL Injection

```
where ..... or 1=1;
sql_command1;
sql_command2;
```

El intérprete ejecuta las dos órdenes seguidas. Jugamos con eso a la hora de mandar datos a la aplicación poniendo en segundo lugar alguna orden “maliciosa”.

AVISO

USAD PreparedStatements

Te evitará problemas posteriormente

Fin

jgardur081@g.educaand.es