



JAVA

JDBC : Traitement des requêtes SQL :
L'interface CallableStatement

L'interface CallableStatement : les procédures stockées

L'objet **CallableStatement** représente une instruction de traitement SQL stockée sur la base de données. C'est une sous-interface de **PreparedStatement**. Tout comme l'interface **PreparedStatement**, il est possible de mettre des « ? » pour réutiliser la requête avec des paramètres différents. Il est créé par l'intermédiaire d'une instance de **Connection** par la méthode :

```
public CallableStatement prepareCall(String sql) throws SQLException;
```

L'interface CallableStatement :

```
try {  
    CallableStatement prepState = connect.prepareCall(sql);  
}  
catch (SQLException e) { ... }
```

Les syntaxes disponibles pour l'appel de procédures stockées sont les suivantes :

- ✓ {call procedure_name[(?, ?, ...)]}
- ✓ {? = call procedure_name[(?, ?, ...)]}

La seconde permettant d'avoir un paramètre de retour.

L'interface CallableStatement :

Cet objet est pourvu de méthodes permettant de :

- Exécuter la requête SQL, retournant les résultats de celle-ci :

```
public ResultSet executeQuery() throws SQLException;  
public int executeUpdate() throws SQLException;
```

- Spécifier le type et la valeur de tel ou tel argument :

```
public void setX(int indiceParameter, X value) throws SQLException;  
où indiceParameter est le rang du paramètre où X est le type de la valeur  
setString(int indiceParameter, String value);  
setInt(int indiceParameter, int value);
```

...

- Vider la liste des paramètres :

```
public void clearParameters() throws SQLException;
```

- Spécifier le type du paramètre renvoyé par la procédure stockée :

```
public void registerOutParameter(int numParam, int typeSQL) throws SQLException;  
où numParam est le rang du paramètre  
où typeSQL est le type SQL du paramètre (constantes disponibles dans java.sql.Types)
```

Exemple

```
String sql = "{call liste_vendeur_par_ville(?)}"; try {  
    CallableStatement callState =  
    connect.prepareCall(sql);  
    callState.setString(1, "Paris");  
    ResultSet rs = callState.executeQuery();  
    ...  
}  
catch (SQLException e) { ... }
```

Tester

