



جامعة طرابلس
University of Tripoli



Java Persistence API



Part-II

- هي عملية حفظ entity object في قاعدة البيانات.

```
Employee emp = new Employee(158);  
em.persist(emp);
```

Finding an Entity

- عند وجود entity في قاعدة البيانات ونرغب في ارجاعها نستخدم:

```
Employee emp = em.find(Employee.class, 158);
```

Removing an Entity

- عند حذف entity يجب ايجادها اولاً ومن ثم استخدام الامر `.remove`.

```
Employee emp = em.find(Employee.class, 158);  
em.remove(emp);
```

Updating an Entity

- عند حصولنا على entity object ويكون في الحالة managed أي تعديل يتم ايضاً في قاعدة البيانات..

```
Employee emp = em.find(Employee.class, 158);  
emp.setSalary(emp.getSalary() + 1000);
```

- تمر Entity بأربع مراحل خلال دورة حياتها وهي:

- **New**

- عند انشاء entity object يكون في الحالة new وفي هذه الحالة هو غير مرتبط بعد ب EntityManager ولم يتواجد في قاعدة البيانات ايضاً

- **Managed**

- Managed entity تكون مرتبطة بقاعدة البيانات وتتم ادارتها من خلال Persistence Context واي تغيير فيها تتأثر به قاعدة البيانات، وتصل لهذه الحالة من خلال الدوال:

- EntityManager.persist(Entity object);

- EntityManager.find(Long id);

Removed •

- تنتقل entity إلى الحالة Removed عند حذفها باستخدام الدالة:

```
EntityManager.remove(book); •
```

Detached •

- تصل entity للحالة detached عند استعمال الدالة:

```
EntityManager.remove(book); •
```

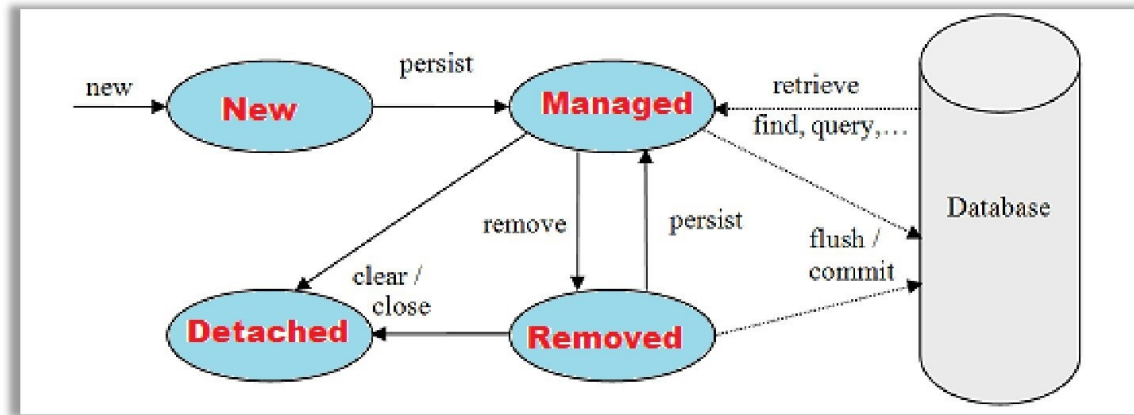
```
Entitymanger.close() •
```

```
Entitymanger.clear •
```

- للتعامل مع detach entity يجب عمل merge لها لتصبح في الحالة manged.

Entity Life Cycle

- الشكل التالي يبين المراحل التي تمر بها entity خلال دورة حياتها.

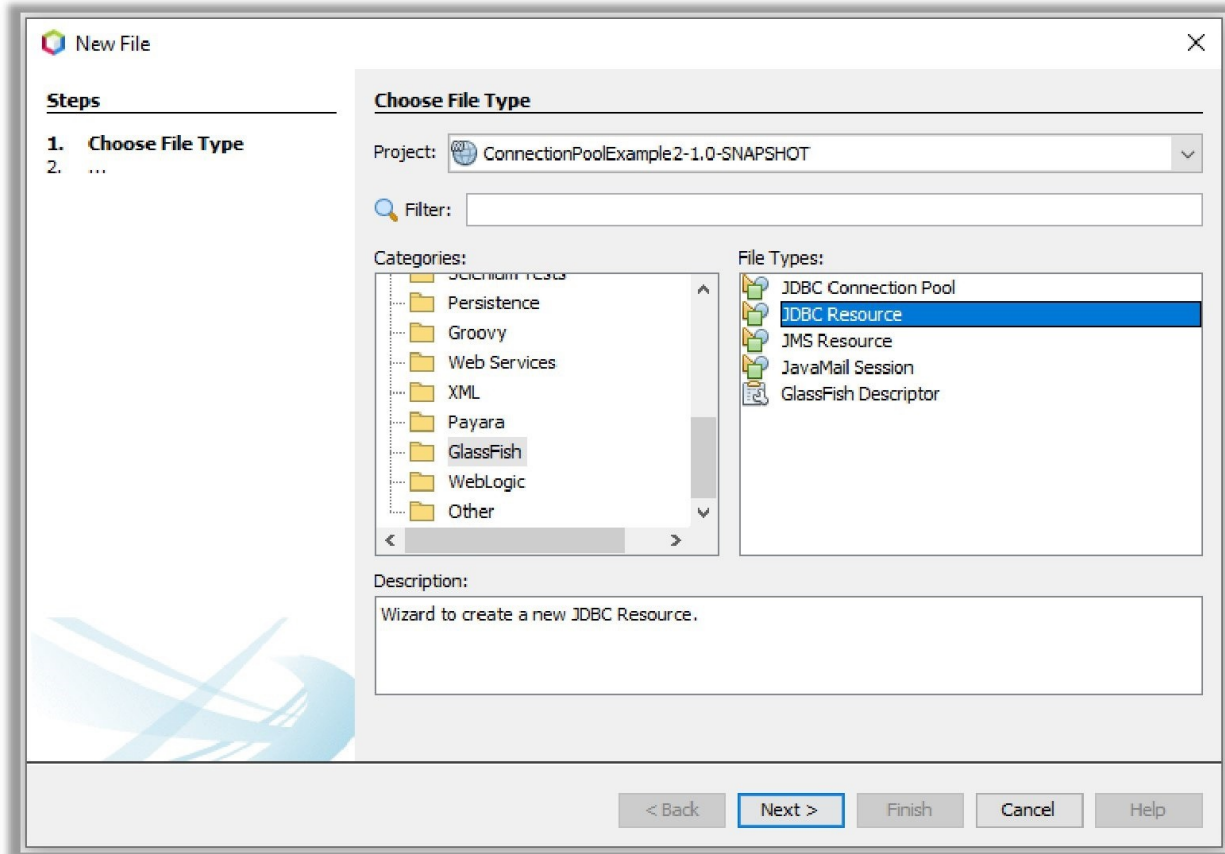


Creating Connection Pool and Data Source



- Connection Pool عبارة عن مجموعة من reusable connections خاصة بقاعدة البيانات.
- Data Source يتم استخدامه للتواصل مع قاعدة البيانات.
- لأنشاء connection pool and data source توجد طريقتان:
- عبر انشاء الملف glassfish-resources.xml وعند عمل deployment يقوم server باستخدامه لأنشاء كل من connection pool and data source .
- استخدام server مباشرة لانشاء كل من connection pool and data source .

- يمكن انشاء ملف glassfish-resources باستخدام netbeans عبر الخيار JDBC Resource من خلال الخطوات التالية.



General Attributes

New JDBC Resource

Steps

1. Choose ...
- 2. General Attributes - JDBC Resource**
3. Properties
4. Choose Database Connection
5. Add Connection Pool Properties
6. Add Connection Pool Optional Properties

General Attributes

Provide configuration information for the JDBC Resource.
Either choose an existing JDBC Connection Pool, or create a new JDBC Connection Pool.
Fields with an * mark are required.

Use Existing JDBC Connection Pool

< No JDBC Connection Pool >

Create New JDBC Connection Pool

JNDI Name: *

Object Type:

Enabled:

Description:

< Back **Next >** Finish Cancel Help

New JDBC Resource

Steps

1. Choose ...
2. General Attributes - JDBC Resource
- 3. Properties**
4. Choose Database Connection
5. Add Connection Pool Properties
6. Add Connection Pool Optional Properties

Additional Properties

Add additional configuration information for the resource jdbc/myDatasource.
Hit the Enter key to save values in the Properties table.

Properties:

Name	Value
------	-------

Connection Pool Properties

New JDBC Resource

Steps

1. Choose ...
2. General Attributes - JDBC Resource
3. Properties
4. Choose Database Connection
- 5. Add Connection Pool Properties**
6. Add Connection Pool Optional Properties

Add Connection Pool Properties

Enter the Datasource Classname, URL, and User to continue.
Hit the Enter key to save values in the Properties table.

Datasource Classname:

Resource Type:

Description:

Properties:

Name	Value
User	root
Password	12345
allowPublicKeyRetrieval	true
UseSSL	false

< Back Finish Cancel Help

Connection Pool Additional Properties

New JDBC Resource X

Steps

1. Choose ...
2. General Attributes - JDBC Resource
3. Properties
4. Choose Database Connection
5. Add Connection Pool Properties
- 6. Add Connection Pool Optional Properties**

Specify Optional Properties for Connection Pool

Pool Settings

Steady Pool Size:

Max Pool Size:

Max Wait Time:

Pool Resize Quantity:

Idle Timeout (secs):

Transaction Isolation

Transaction Isolation:

Guarantee Isolation Level:

Connection Validation

Connection Validation Required:

Validation Method:

Table Name:

Fail All Connections:

Non Transactional Connections:

Allow Non Component Callers:

- بعد إتمام عملية الانشاء يجب التأكد من امتداد الملف xml .
- مراجعة المحتوى ويجب ان تكون مطابقة لما تم إدخاله.
- سيقوم server بتكوين connection pool and data source عند اجراء اول عملية deployment .
- يجب حذف الملف بعد نجاح عملية الانشاء لاحتواه على أسم المستخدم وكلمة السر.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE resources PUBLIC "-//GlassFish.org//DTD GlassFish Application Server 3.1 Resource Definitions//EN"
"http://glassfish.org/dtds/glassfish-resources_1_5.dtd">
<resources>
  <jdbc-resource enabled="true" jndi-name="jdbc/connectionPoolExample2" object-type="user" pool-name="connectionPoolExample2">
    <description/>
  </jdbc-resource>
  <jdbc-connection-pool allow-non-component-callers="false" associate-with-thread="false" connection-creation-retry-attempts="0"
    <property name="URL" value="jdbc:mysql://localhost:3308/jpalecture?zeroDateTimeBehavior=CONVERT_TO_NULL"/>
    <property name="User" value="root"/>
    <property name="Password" value="12345"/>
    <property name="allowPublicKeyRetrieval" value="true"/>
    <property name="UseSSL" value="false"/>
  </jdbc-connection-pool>
</resources>
```

- استخدام server مباشرة لإنشاء كل من connection pool and data source .
- يتم ذلك من خلال admin console والخيار JDBC Connection Pools لإنشاء connection pool .
- الخيار JDBC Resources لإنشاء data source

JDBC Connection Pools

To store, organize, and retrieve data, most applications use relational databases. Java EE applications access relational databases through the JDBC API. Before an application can access a database, it must get a connection.

Pools (8)



New...

Delete

Select	Pool Name	Resource Type	Classname	Description
<input type="checkbox"/>	SamplePool	javax.sql.DataSource	org.apache.derby.jdbc.ClientDataSource	
<input type="checkbox"/>	__TimerPool	javax.sql.XADataSource	org.h2.jdbcx.JdbcDataSource	
<input type="checkbox"/>	connectionPoolExample	javax.sql.DataSource	com.mysql.cj.jdbc.MySQLDataSource	
<input type="checkbox"/>	connectionPoolExample2	javax.sql.DataSource	com.mysql.cj.jdbc.MySQLDataSource	

New JDBC Connection Pool (Step 1 of 2)

Identify the general settings for the connection pool.

* Indicates required field

General Settings

Pool Name: *

Resource Type:

Must be specified if the datasource class implements more than 1 of the interface.

Database Driver Vendor:

Select or enter a database driver vendor

Introspect:

Enabled

If enabled, data source or driver implementation class names will enable introspection.

Additional Properties (5)



Add Property

Delete Properties

Select	Name	Value	Description
<input type="checkbox"/>	User	root	
<input type="checkbox"/>	Password	12345	
<input type="checkbox"/>	UseSSL	false	
<input type="checkbox"/>	allowPublicKeyRetrieval	true	
<input type="checkbox"/>	url	jdbc:mysql://localhost:3308/jpalecture?zeroDateTi	

General

Advanced

Additional Properties

 Ping Succeeded

Edit JDBC Connection Pool

Save

Cancel

Modify an existing JDBC connection pool. A JDBC connection pool is a group of reusable connections for a particular database.

Load Defaults

Flush

Ping

يستخدم الخيار JDBC Resources لإنشاء data source .

JDBC Resources

JDBC resources provide applications with a means to connect to a database.

Resources (7)



New...

Delete

Enable

Disable

Select	JNDI Name	Logical JNDI Name	Enabled	Connection Pool	Description
<input type="checkbox"/>	jdbc/_TimerPool		✓	_TimerPool	
<input type="checkbox"/>	jdbc/_default	java.comp/DefaultDataSource	✓	H2Pool	

New JDBC Resource

OK

Cancel

Specify a unique JNDI name that identifies the JDBC resource you want to create. The name must contain only alphanumeric, underscore, dash, or dot characters.

JNDI Name: *

Pool Name:

Use the [JDBC Connection Pools](#) page to create new pools

Description:

Status: Enabled

Additional Properties (0)

Add Property

Delete Properties

Select	Name	Value	Description
No items found.			

- يتم استخدام data source الذي تم إنشاؤه في ملف persistence.xml ليتم من خلاله تحديد قاعدة البيانات التي ستستخدمها persistence unit.

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.2" xmlns="http://xmlns.jcp.org/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence/persistence_2_2.xsd">
  <!-- Define Persistence Unit -->
  <persistence-unit name="my_persistence_unit">
    <jta-data-source>jdbc/connectionPoolExample2</jta-data-source>
    <properties>
      <property name="javax.persistence.schema-generation.database.action" value="drop-and-create"/>
    </properties>
  </persistence-unit>
</persistence>
```