



# Chương 4: Ứng dụng kết nối CSDL NoSQL

**Biên soạn: Bộ môn HTTT**

# Nội dung



**1. Giới thiệu**

**2. Ứng dụng kết nối Mongodb**

**3. Ứng dụng kết nối Neo4J**

# Giới thiệu

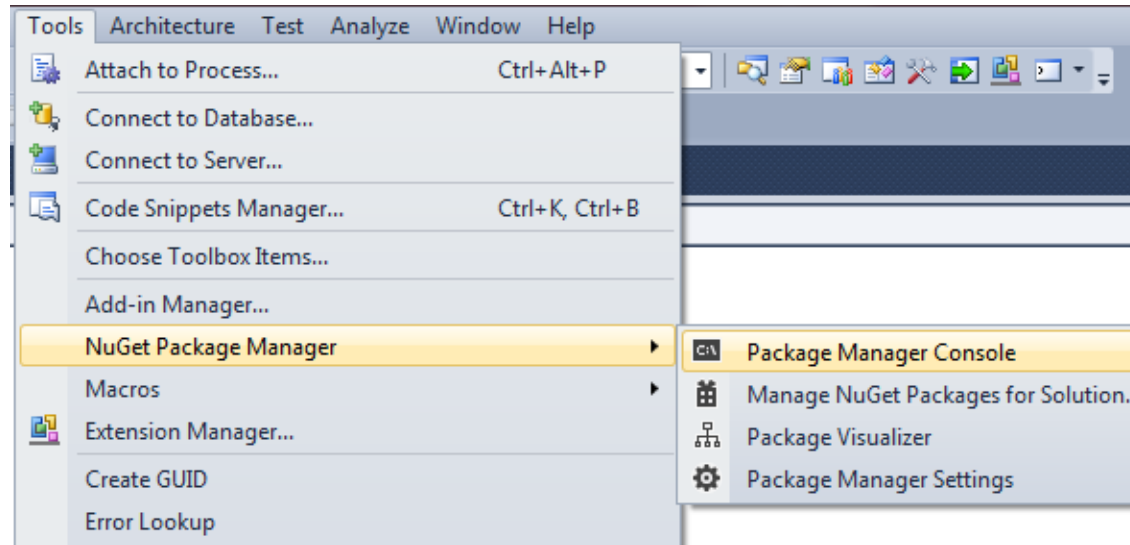
- ❖ Việc ứng dụng cơ sở dữ liệu MongoDB, Neo4J cũng tương tự như các hệ quản trị khác, cần xây dựng các giao diện giao tiếp người dùng bằng một ngôn ngữ lập trình cụ thể và kết nối đến MongoDB, Neo4J.
- ❖ MongoDB và Neo4J hỗ trợ rất nhiều ngôn ngữ lập trình ứng dụng như: C#, Java, Python, PHP,... Mỗi ngôn ngữ được xây dựng gói chương trình gọi là driver tương ứng để kết nối thao tác trên cơ sở dữ liệu.

# Ứng dụng C# kết nối MongoDB

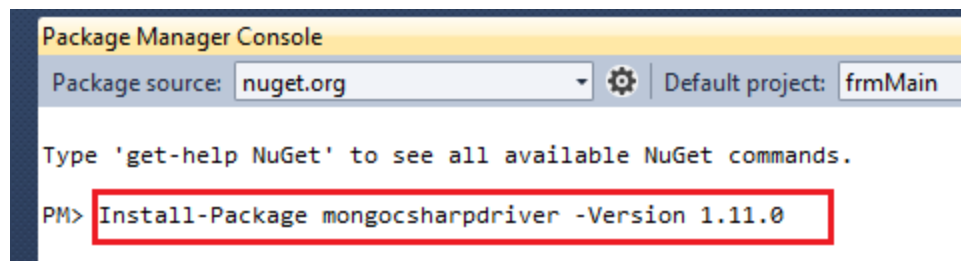
- Tên gói chương trình: **mongo-csharp-driver**
- Cài đặt:
  - Cài đặt online bằng *Nuget Package Manager*:  
<https://www.nuget.org/packages/mongocsharpdriver>
  - Hoặc add References (nếu đã có file .DLL):
    1. MongoDB.Bson.dll
    2. MongoDB.Driver.dll
- **Lưu ý:** *mongo-csharp-driver* phiên bản 2.0 trở lên sẽ tương thích với bộ .Net Framework 4.5 trở lên.

# Cài đặt bằng Nuget

**Bước 1:** Trên project hiện hành, vào menu Tool > NuGet Package Manager > Package Manager Console

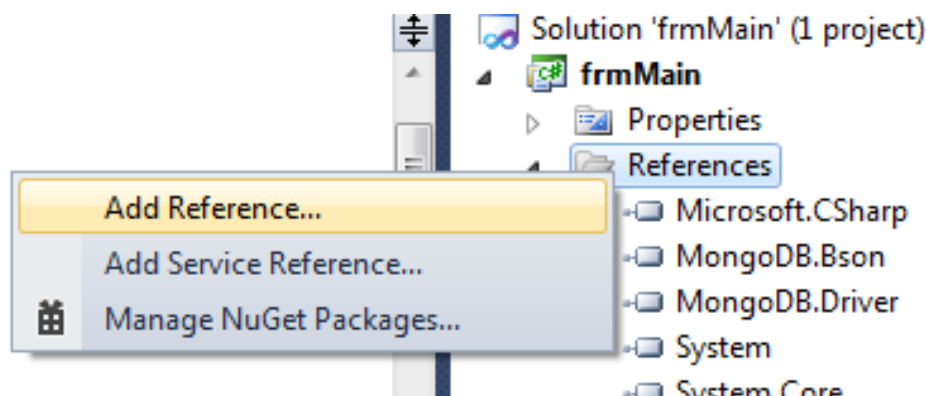


**Bước 2:** Trên cửa sổ Package Manager Console, gõ vào lệnh cài đặt, nhấn Enter

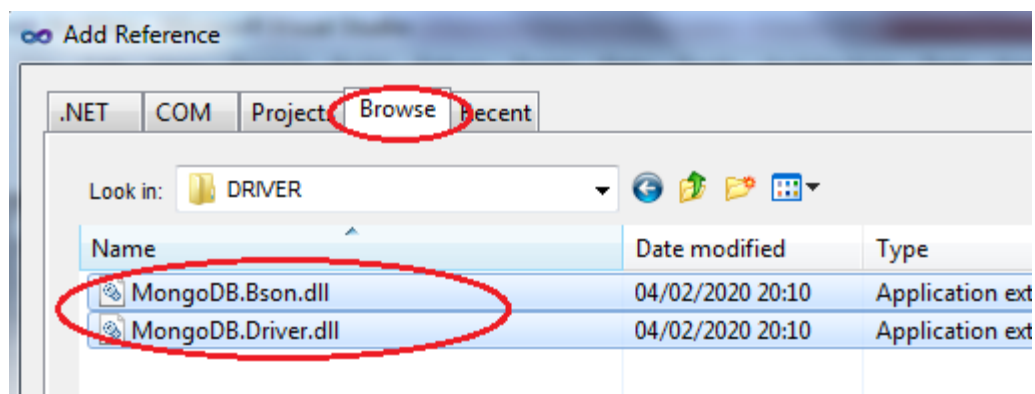


# Cài đặt bằng file .DLL

**Bước 1:** Mở rộng danh mục project > Click phải vào References > chọn Add References



**Bước 2:** Chọn Tab Browse > chọn 2 tập tin .dll > OK



# Ứng dụng C# kết nối MongoDB

Sau khi cài đặt *mongo-csharp-driver*, ngoài các namespace mặc định, sử dụng thêm các namespace sau để kết nối mongodb:

```
6  using System.Linq;
7  using System.Text;
8  using System.Windows.Forms;
9  using MongoDB.Driver;
10 using MongoDB.Bson;
11
```

# Kết nối database

- ❖ Sử dụng chuỗi connectionString kết nối cơ sở dữ liệu MongoDB:

```
var connectionString="mongodb://localhost:27017/admin";  
var client = new MongoClient(connectionString);  
var server = client.GetServer();
```

- ❖ Lấy đối tượng database:

```
var db = server.GetDatabase("mydb");
```

- ❖ Lấy đối tượng collection:

```
var collection=db.GetCollection<BsonDocument>(collectionName);
```

**collectionName:** Tên của collection trong database. Ví dụ: NHANVIEN



# Thao tác trên collection

- ❖ Duyệt các Document trong collection
- ❖ Thêm Document vào collection
- ❖ Xóa Document khỏi collection
- ❖ Cập nhật collection

# Duyệt các Document

```
foreach (BsonDocument document in collection.FindAll())
{
    string value =
        document.GetElement(fieldname).Value.ToString();
}
```

## Trong đó:

- fieldname: là tên của thuộc tính cần lấy giá trị, kết quả lưu vào biến value.

# Thêm document vào collection

```
BsonDocument document = new BsonDocument()  
    .Add(field1, valueOfField1)  
    .Add(field2, valueOfField2)  
    ...  
    .Add(fieldN, valueOfFieldN);  
collection.Insert(document);
```

## Trong đó:

- Field1, field2,...là tên của các thuộc tính.
- valueOfField1, valueOfField2,...là các giá trị thuộc tính tương ứng.

# Xóa document trong collection

```
var query = new QueryDocument(fieldName, valueOfField);  
collection.Remove(query);
```

# Sửa document trong collection

```
BsonDocument document = new BsonDocument()  
    .Add("_id", valueOf_id)  
    .Add(field1, valueOfField1)  
    .Add(field2, valueOfField2)  
    ...  
    .Add(fieldN, valueOfFieldN);  
collection.Save(document);
```

## Trong đó:

- `valueOf_id` là giá trị của thuộc tính `_id` của document cần sửa thông tin. Có thể viết hàm để lấy giá trị này.

# Bài tập tạo ứng dụng C#

Tạo ứng dụng Window form C# với các yêu cầu sau:

- Tạo một cơ sở dữ liệu MongoDB gồm 1 collection có tên NhanVien. Trong collection này nhập các document có nội dung như trên form.
- Cài đặt các chức năng trên form:
  - Chương trình khi khởi động sẽ load dữ liệu từ database lên form.
  - Nút thêm mới: Xóa trắng các textbox, đặt dấu nháy ở Mã nhân viên.
  - Nút Thêm, xóa, sửa: sẽ thêm, xóa, sửa nhân viên.
  - Khi click chọn dòng trên datagridview sẽ hiển thị thông tin lên textbox

# Bài tập tạo ứng dụng C#

frmMain

Mã nhân viên: NV002

Họ tên: Nguyễn Thị Lệ

Tuổi: 22

Mã phòng: P001

	MANV	HOTEN	TUOI	MAPH
	NV001	Trần Văn Minh	30	P001
▶	NV002	Nguyễn Thị Lệ	22	P001
	NV003	Đỗ Thanh Tuấn	43	P002
	NV003	Trương Minh Huy	50	P002
	NV004	Lương Mai Phương	32	P003
	NV005	Trần Thị Thùy Linh	23	P004

# Ứng dụng Java kết nối MongoDB

## ❖ Ngôn ngữ Java

1. Tên gói chương trình: **Mongodb Java Driver**

- Link download: [https://jar-download.com/?search\\_box=mongo-java-driver](https://jar-download.com/?search_box=mongo-java-driver)

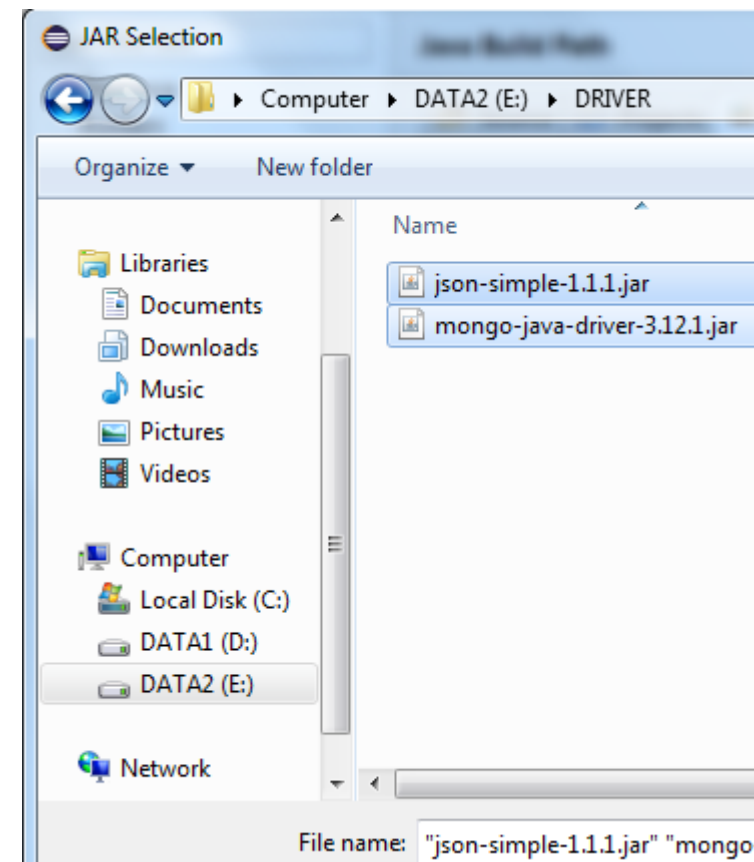
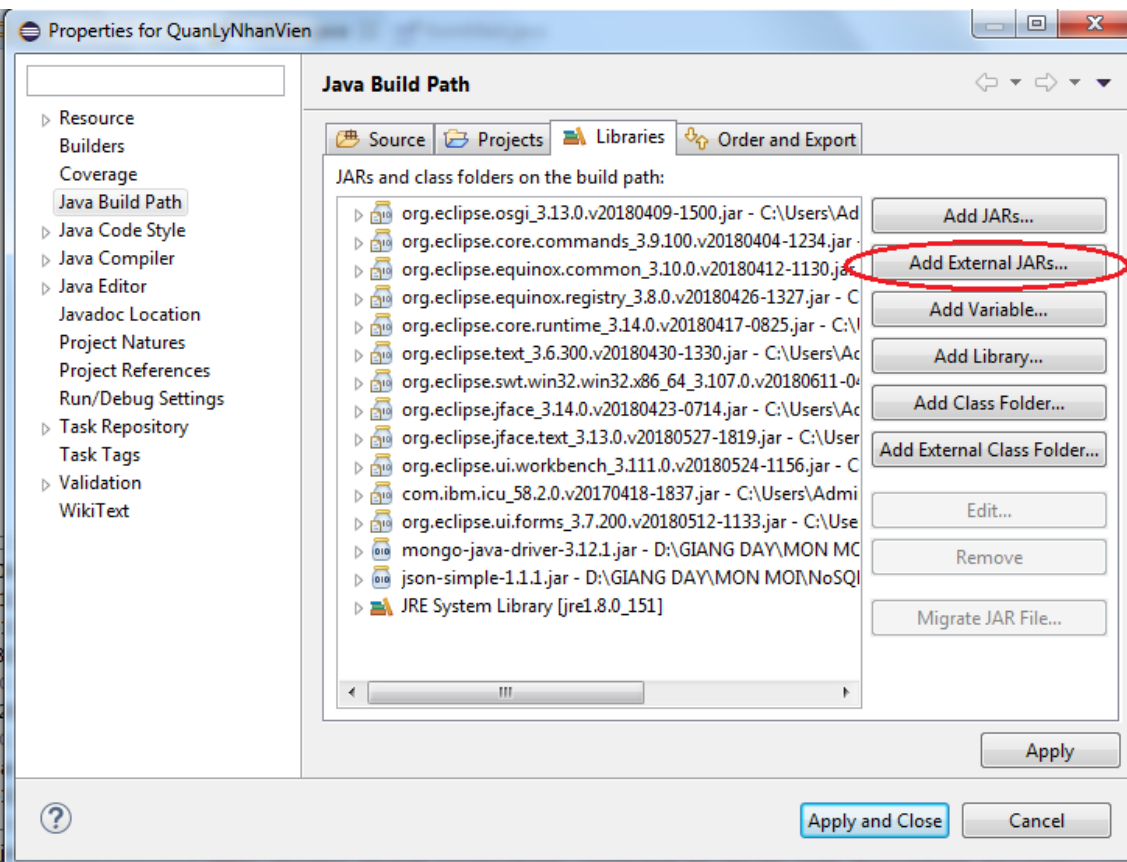
2. Thao tác trên cấu trúc Json: **Json-simple**

- Link download: <https://jar-download.com/artifacts/com.googlecode.json-simple/json-simple/1.1.1/source-code>



# Import MongoDB Java Driver

Click phải vào project > chọn Build Path > chọn Configure Build Path > click nút lệnh Add External Jars > chọn các file cần import.



# Kết nối MongoDB

Sau khi cài đặt *mongoDB Java driver*, ngoài các package mặc định, import thêm các package sau để kết nối mongodb:

```
17 import org.eclipse.swt.widgets.Item;
18
19 import org.json.simple.*;
20 import org.json.simple.parser.JSONParser;
21 import org.json.simple.parser.ParseException;
22
23 import com.mongodb.MongoClient;
24 import com.mongodb.client.MongoCollection;
25 import com.mongodb.client.MongoDatabase;
26 import static com.mongodb.client.model.Filters.*;
27
```

# Kết nối MongoDB

## ❖ Kết nối cơ sở dữ liệu MongoDB:

```
MongoClient mongoClient = new MongoClient("localhost", 27017);
```

## ❖ Lấy đối tượng database:

```
MongoDatabase database = mongoClient.getDatabase(dbName);
```

## ❖ Lấy đối tượng collection:

```
MongoCollection<Document> collection =  
    database.getCollection(collectName);
```

# Duyệt các Document

```
for (Document doc : collection.find()){  
    JSONParser parser = new JSONParser();  
    JSONObject json = (JSONObject) parser.parse(doc.toJson());  
    String value = json.get(fieldName).toString();  
}
```

## Trong đó:

- fieldName: là tên của thuộc tính cần lấy giá trị, kết quả lưu vào biến value.

# Thêm document vào collection

```
Document doc=new Document(field1, valueOfField1)
    .append(field2, valueOfField2)
    .append(field3, valueOfField3)
    ...
    .append(fieldN, valueOfFieldN)
collection.insertOne(doc);
```

## Trong đó:

- Field1, field2,...là tên của các thuộc tính.
- valueOfField1, valueOfField2,...là các giá trị thuộc tính tương ứng.

# Xóa document trong collection

```
collection.deleteOne(eq(FieldName,Value));
```

## Trong đó:

- FieldName: là tên của các thuộc tính cần thiết lập điều kiện xóa.
- Value: là giá trị của thuộc tính làm điều kiện xóa.
- eq: (equal) so sánh bằng. Ngoài ra còn có các ký hiệu khác như: ne (not equal), gt (greater than), lt (less than), gte (greater than or equal), lte (less than or equal)

# Sửa document trong collection

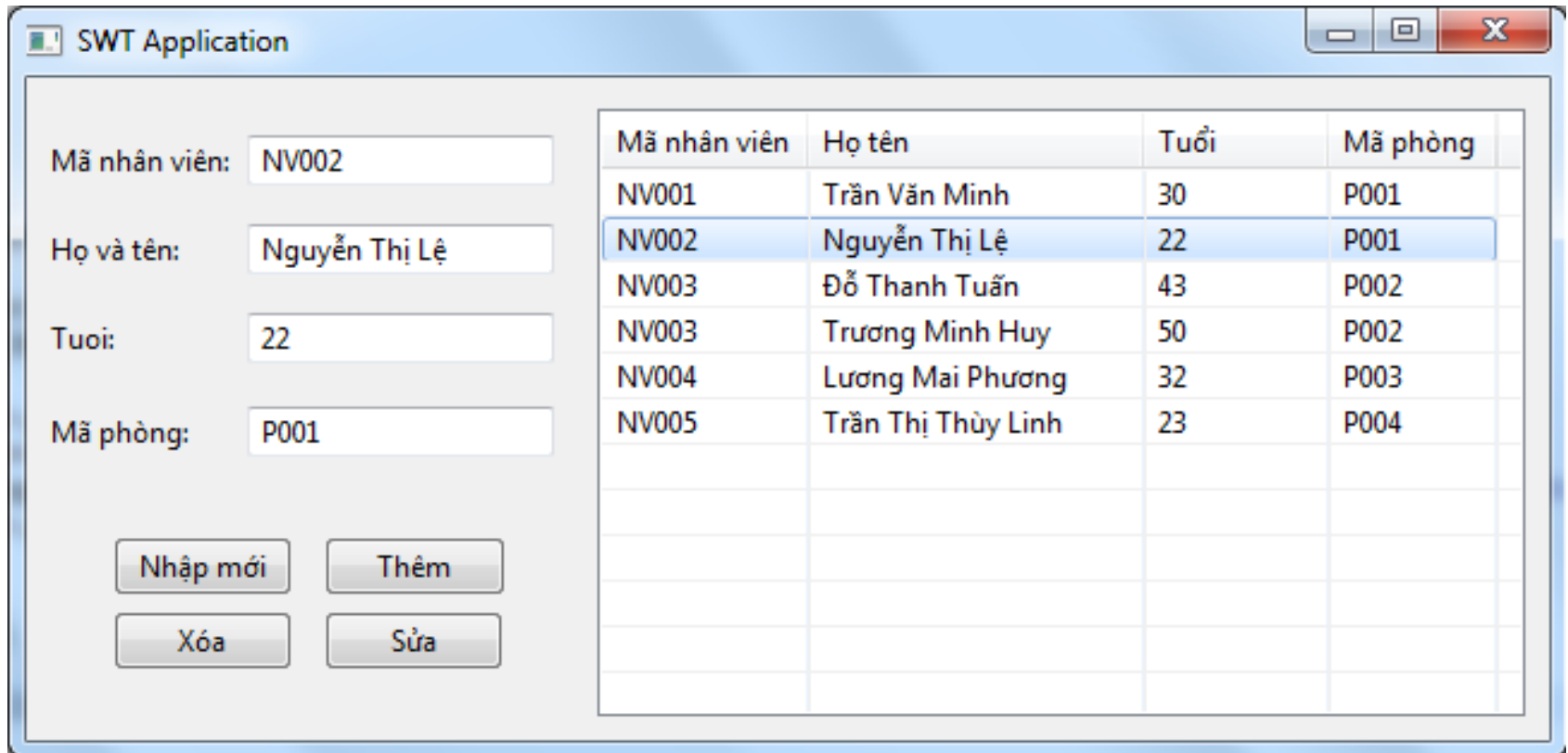
```
collection.updateOne(eq(FieldName,oldValue), new  
Document("$set", new Document(FieldName,newValue)));
```

## Trong đó:

- **FieldName** là tên thuộc tính của document cần sửa thông tin. **oldValue** là giá trị cũ của thuộc tính. **newValue** là giá trị mới cần sửa.

# Bài tập tạo ứng dụng Java

Tạo ứng dụng Java, sử dụng gói giao diện SWT trên Eclipse hoặc tương đương với các yêu cầu như trong bài tập ứng dụng c#.



The screenshot shows a Java SWT Application window titled "SWT Application". On the left, there is a form with four input fields and four buttons. The input fields are labeled "Mã nhân viên:", "Họ và tên:", "Tuoi:", and "Mã phòng:", with values "NV002", "Nguyễn Thị Lệ", "22", and "P001" respectively. The buttons are labeled "Nhập mới", "Thêm", "Xóa", and "Sửa". On the right, there is a table with four columns: "Mã nhân viên", "Họ tên", "Tuổi", and "Mã phòng". The table contains five rows of data, with the second row (NV002, Nguyễn Thị Lệ, 22, P001) highlighted in blue.

Mã nhân viên	Họ tên	Tuổi	Mã phòng
NV001	Trần Văn Minh	30	P001
NV002	Nguyễn Thị Lệ	22	P001
NV003	Đỗ Thanh Tuấn	43	P002
NV003	Trương Minh Huy	50	P002
NV004	Lương Mai Phương	32	P003
NV005	Trần Thị Thùy Linh	23	P004

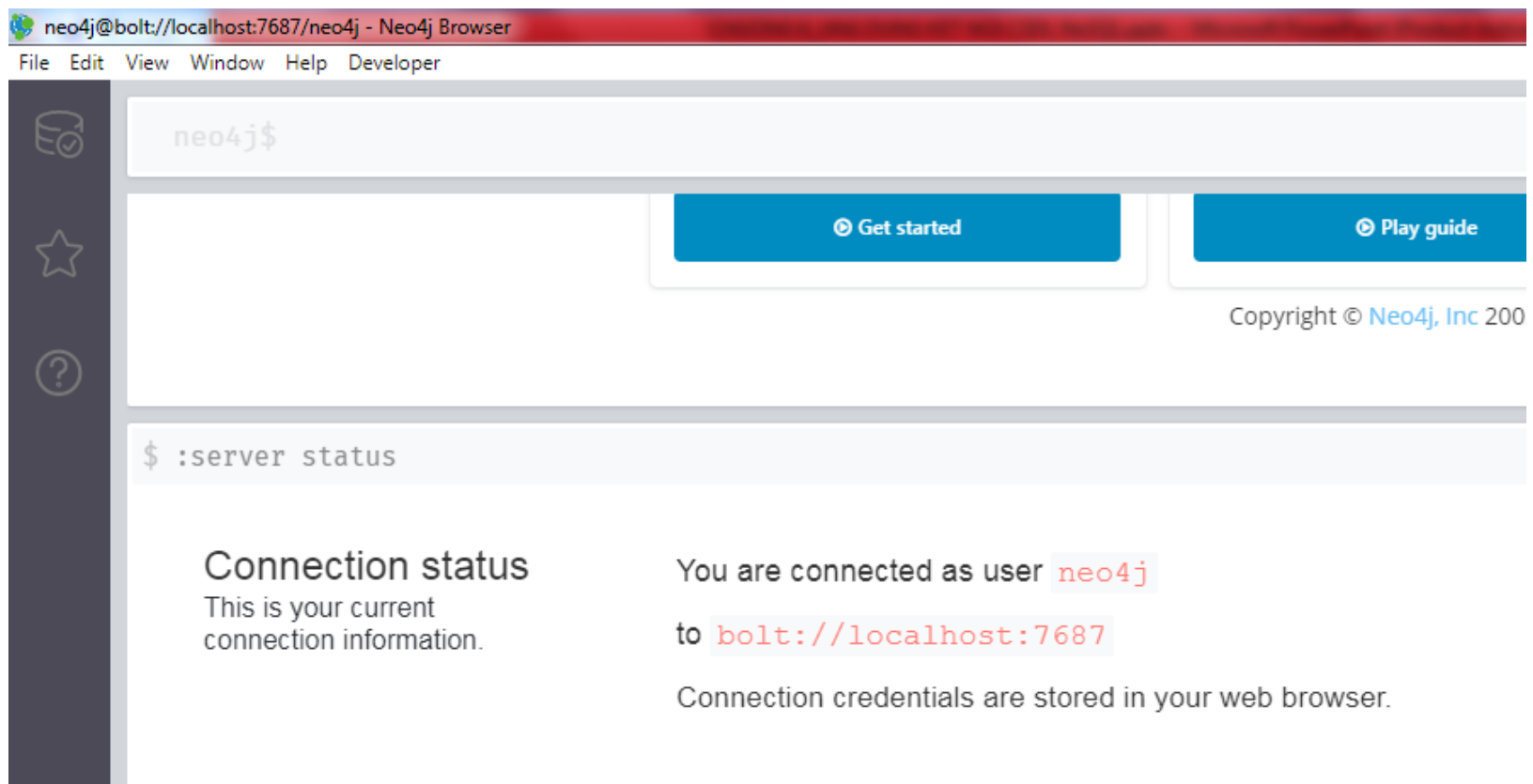


# Ứng dụng Java kết nối Neo4J

1. Tên gói chương trình: **neo4j-java-driver**
  - Link download: <https://jar-download.com/artifact-search/neo4j-java-driver>
  - Giải nén và import vào project

# Kết nối Neo4J

- Khởi động Neo4J Desktop > start database
- Khởi động Neo4J Browser để lấy thông tin user và uri hoặc gõ lệnh :server status;



# Kết nối Neo4J (tt)

## ❖ Import driver

```
import org.neo4j.driver.*;
```

## ❖ Kết nối Neo4J:

```
driver=GraphDatabase.driver(uri,AuthTokens.basic(user,password))
```

## ❖ Ví dụ:

```
driver = GraphDatabase.driver("bolt://localhost:7687",  
AuthTokens.basic("neo4j", "123"));
```

# Cấu trúc lệnh thao tác

```
try (Session session = driver.session())  
{  
    //lệnh xử lý  
}
```

# Xử lý kết quả truy vấn

## ❖ Xử lý kết quả truy vấn:

```
try (Session session = driver.session())
{
    Result result=session.run("Lệnh truy vấn...As name");
    while(result.hasNext())
    {
        Record record=result.next();
        System.out.Print(record.get("name").asString());
    }
}
```

# Xử lý kết quả truy vấn (tt)

## ❖ Ví dụ:

```
private void printPeople(String nodeName)
{
    try (Session session = driver.session())
    {
        Result result=session.run("MATCH (p:Person {name:\""+
+ nodeName +"\"})-[:IS_FRIENDS_WITH]->(q) return q.name
AS name");
        while (result.hasNext())
        {
            Record record = result.next();
            System.out.println(record.get("name").asString());
        }
    }
}
```

# Sử dụng tham số

```
Map<String,Object> params=new HashMap<>();  
params.put(attribute,value);
```

```
private void printPeople(String name_value)  
{  
    try (Session session = driver.session())  
    {  
        Map<String,Object> params=new HashMap<>();  
        params.put("name_parameter",name_value);  
        Result result=session.run("MATCH (p:Person{name:$name_parameter}),  
        (q:Person) WHERE (p)-[:IS_FRIENDS_WITH]->(q) return q.name AS  
name",params);  
        while (result.hasNext())  
        {  
            Record record = result.next();  
            System.out.println(record.get("name").asString());  
        }  
    }  
}
```

# Sử dụng tham số (tt)

```
private void addPerson(String name, int age)
{
    try (Session session = driver.session())
    {
        Map<String, Object> params = new HashMap<>();
        params.put("name", name);
        params.put("age", age);
        session.writeTransaction(tx -> tx.run("CREATE(a:Person
{name: $name, age: $age})", params));
    }
}
```



# Bài tập tạo ứng với Neo4J

Tạo ứng dụng Java, sử dụng gói giao diện SWT trên Eclipse hoặc tương đương.

The screenshot shows a Java SWT Application window titled "SWT Application". The window contains a form with three sections for interacting with a Neo4J database:

- Top Section:** A label "Nhấn nút" (Click button) next to a dropdown menu showing "Person". To the right is a "Tạo nút" (Create node) button. Below this is a label "Thuộc tính" (Property) next to an empty text input field.
- Middle Section:** A label "Chọn nút" (Select node) next to two dropdown menus, the first showing "Lan" and the second showing "Bình". To the right is a "Tạo liên kết" (Create relationship) button. Below this is a label "Liên kết" (Relationship) next to a dropdown menu showing "IS\_FRIENDS\_WITH". Below that is a label "Thuộc tính" (Property) next to an empty text input field.
- Bottom Section:** A label "Chọn nút, liên kết" (Select node, relationship) next to two dropdown menus, the first showing "Lan" and the second showing "IS\_FRIENDS\_WITH". To the right is a "Tìm kiếm" (Search) button. Below this is an empty text input field.



Hết chương 4