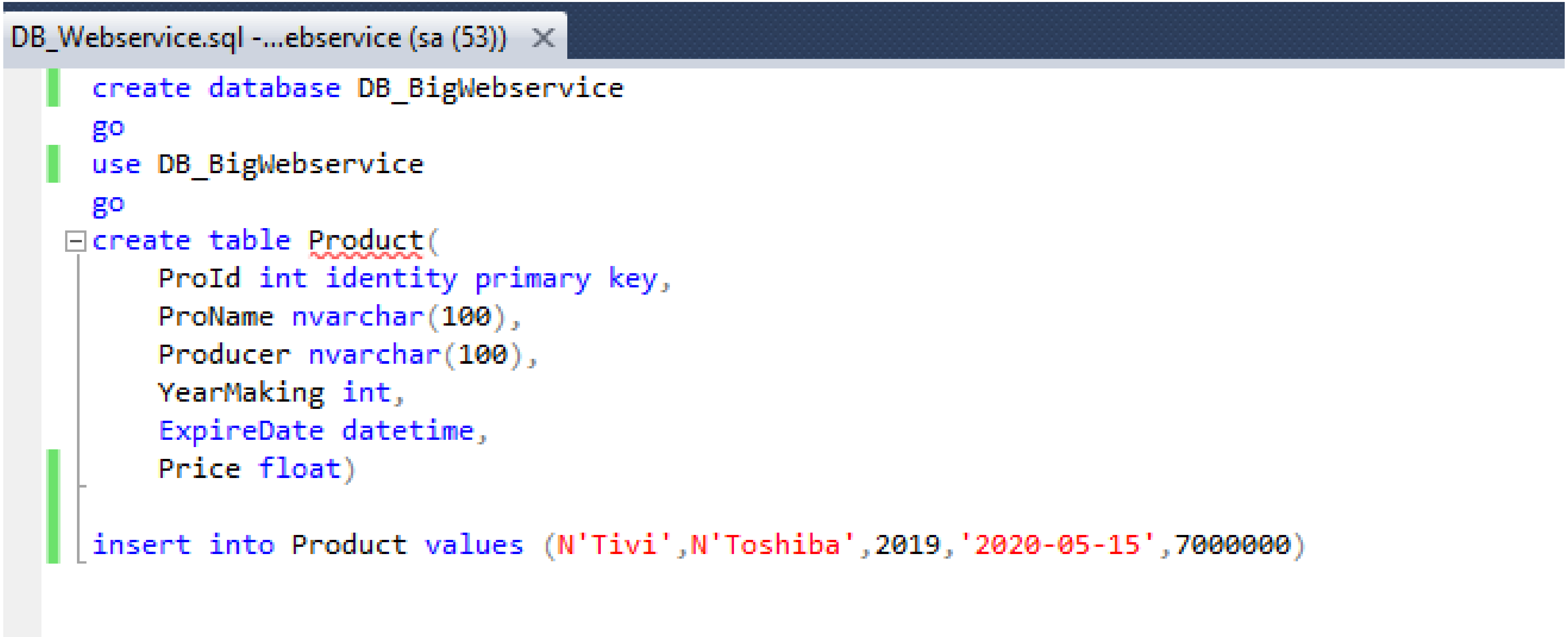
Bài 2
Phát Triển Web Services Với Big Web Service

MỤC TIÊU

- Tạo database cho ứng dụng
- Sử dụng hibernate để cài đặt các hàm tương tác database
- Tạo Big Web services để cung cấp các phương thức sử dụng các hàm cài đặt trong DAO

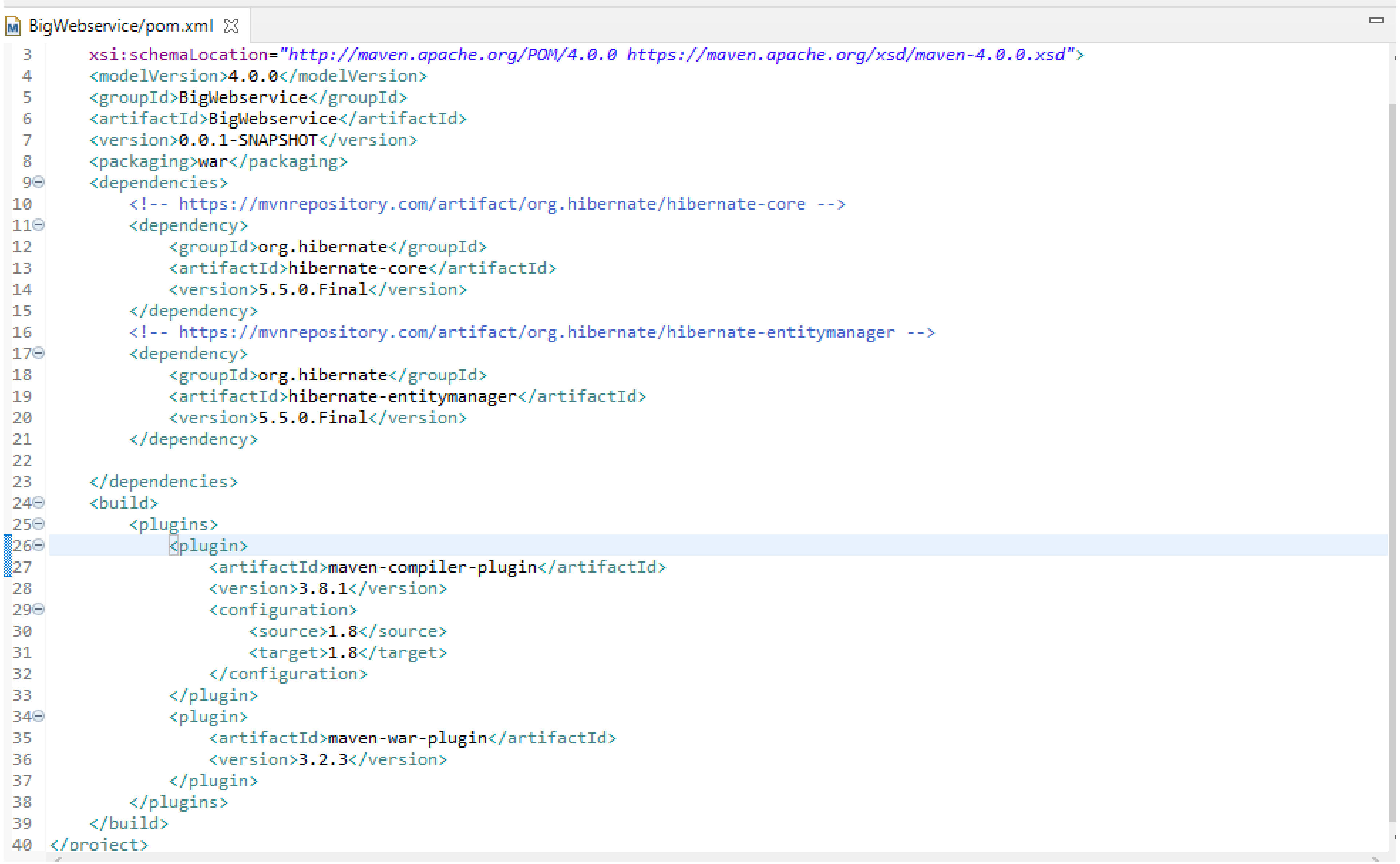
Tạo database cho ứng dụng



DB_Webservice.sql -...ebservice (sa (53))

```
create database DB_BigWebservice
go
use DB_BigWebservice
go
create table Product(
    ProId int identity primary key,
    ProName nvarchar(100),
    Producer nvarchar(100),
    YearMaking int,
    ExpireDate datetime,
    Price float)

insert into Product values (N'Tivi',N'Toshiba',2019,'2020-05-15',7000000)
```

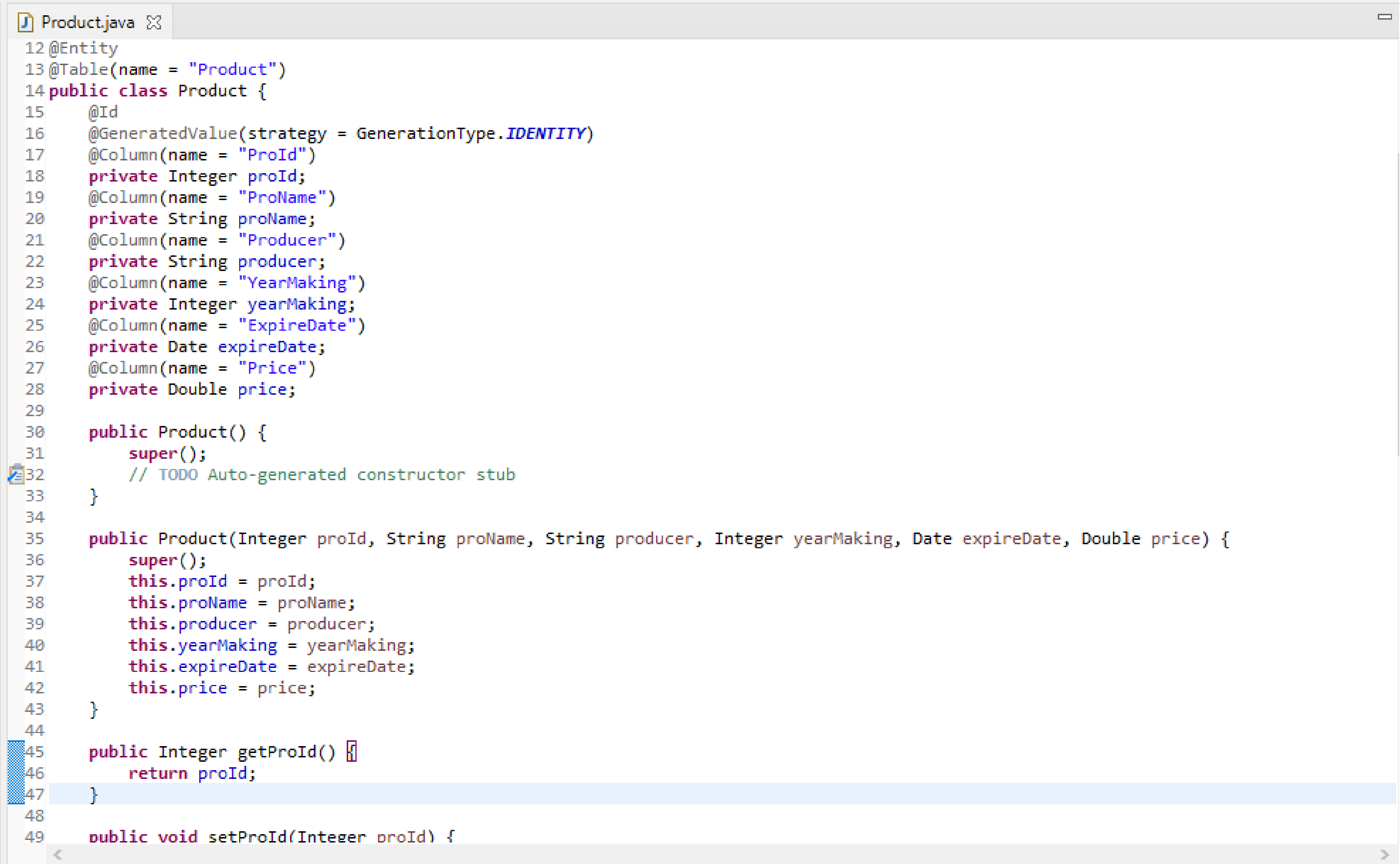
Tạo project và thêm thư viện vào file Pom.xml



BigWebservice/pom.xml

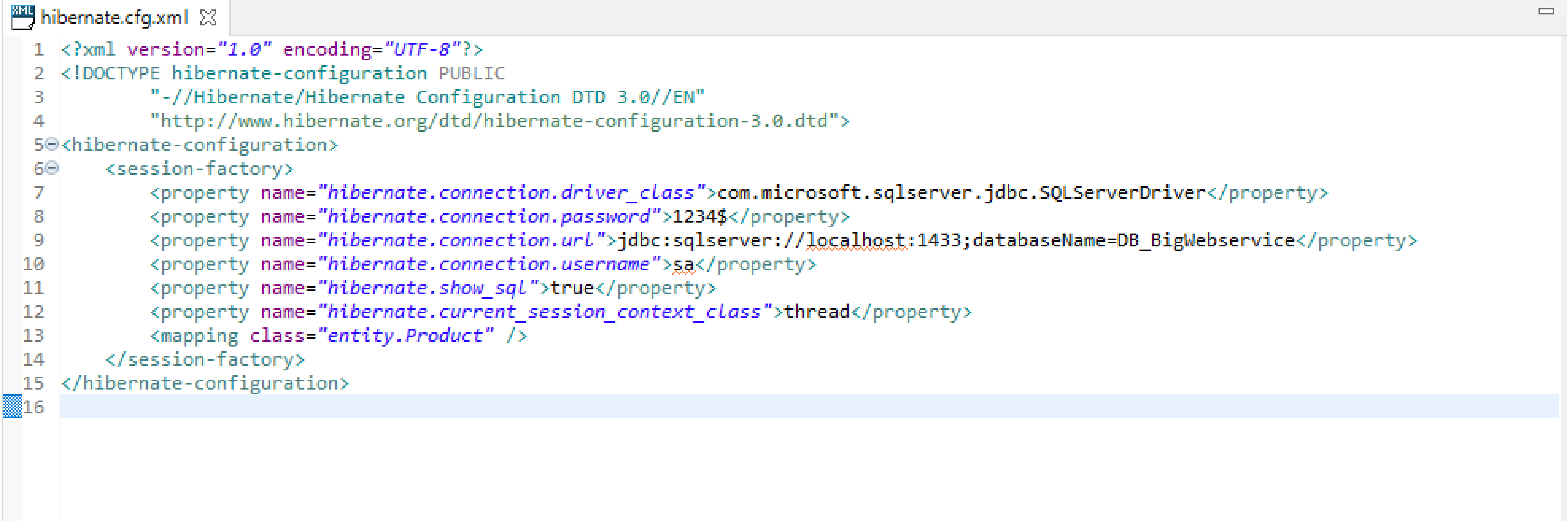
```xml
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>BigWebservice</groupId>
    <artifactId>BigWebservice</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>war</packaging>
    <dependencies>
        <!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
        <dependency>
            <groupId>org.hibernate</groupId>
            <artifactId>hibernate-core</artifactId>
            <version>5.5.0.Final</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-entitymanager -->
        <dependency>
            <groupId>org.hibernate</groupId>
            <artifactId>hibernate-entitymanager</artifactId>
            <version>5.5.0.Final</version>
        </dependency>

    </dependencies>
    <build>
        <plugins>
            <plugin>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.8.1</version>
                <configuration>
                    <source>1.8</source>
                    <target>1.8</target>
                </configuration>
            </plugin>
            <plugin>
                <artifactId>maven-war-plugin</artifactId>
                <version>3.2.3</version>
            </plugin>
        </plugins>
    </build>
</project>
```

Tạo entity class ánh xạ với bảng trong database



```java
@Entity
@Table(name = "Product")
public class Product {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "ProId")
    private Integer proId;
    @Column(name = "ProName")
    private String proName;
    @Column(name = "Producer")
    private String producer;
    @Column(name = "YearMaking")
    private Integer yearMaking;
    @Column(name = "ExpireDate")
    private Date expireDate;
    @Column(name = "Price")
    private Double price;

    public Product() {
        super();
        // TODO Auto-generated constructor stub
    }

    public Product(Integer proId, String proName, String producer, Integer yearMaking, Date expireDate, Double price) {
        super();
        this.proId = proId;
        this.proName = proName;
        this.producer = producer;
        this.yearMaking = yearMaking;
        this.expireDate = expireDate;
        this.price = price;
    }

    public Integer getProId() {
        return proId;
    }

    public void setProId(Integer proId) {
```

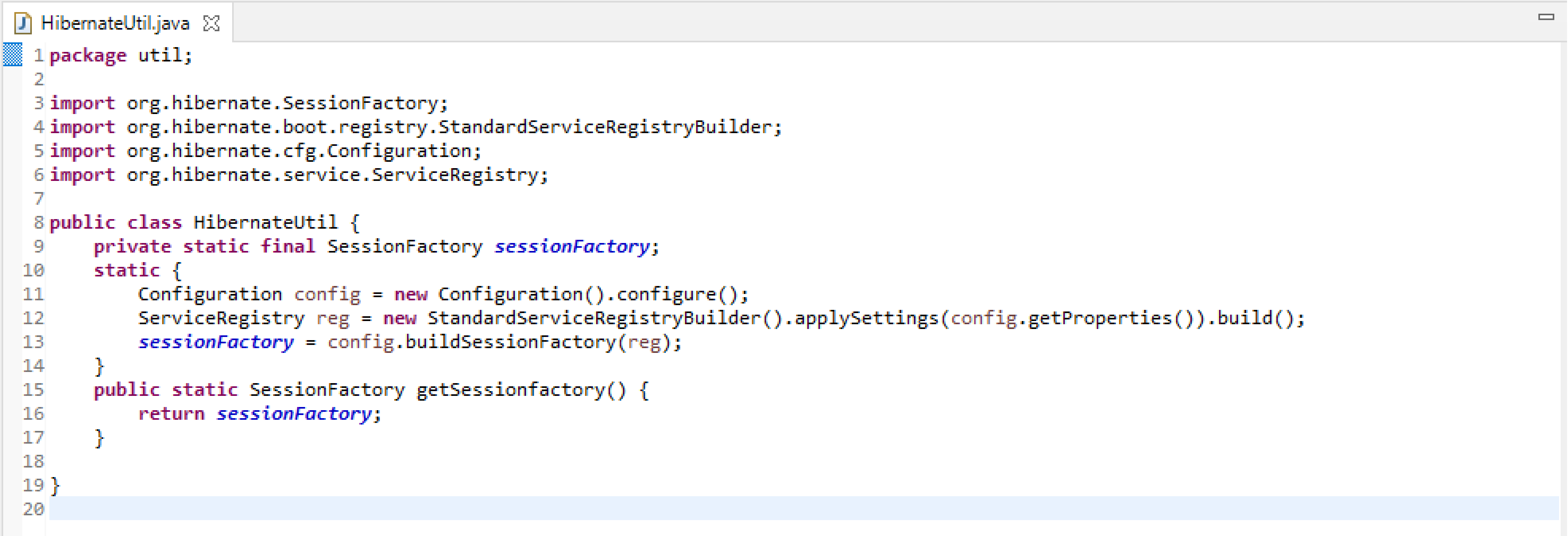
Tạo file cấu hình hibernate.cfg.xml



hibernate.cfg.xml

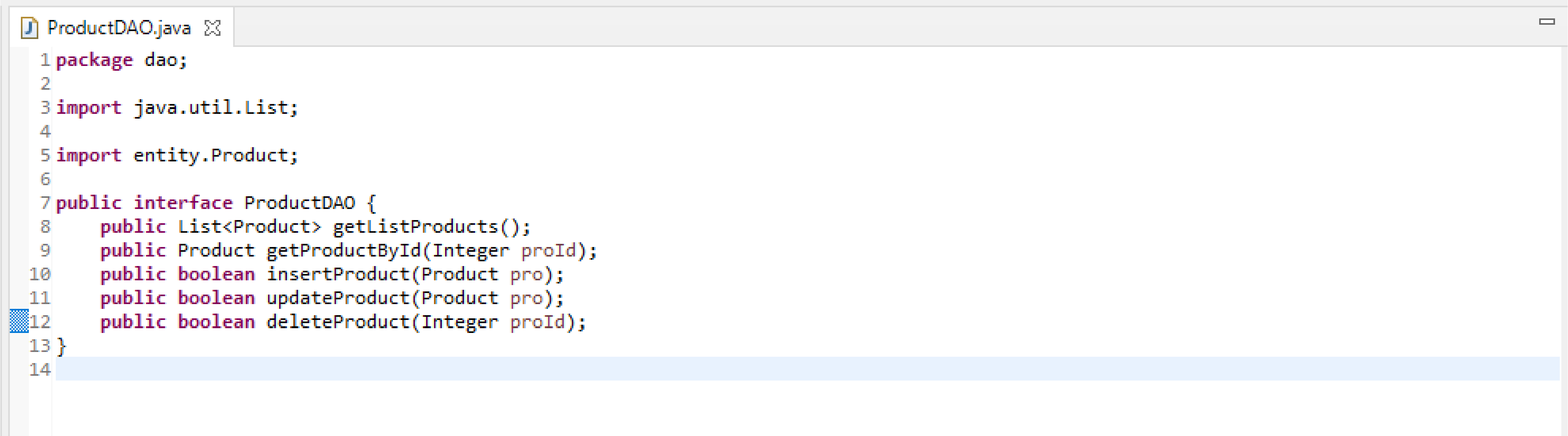
```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
        "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
        "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <property name="hibernate.connection.driver_class">com.microsoft.sqlserver.jdbc.SQLServerDriver</property>
        <property name="hibernate.connection.password">1234$</property>
        <property name="hibernate.connection.url">jdbc:sqlserver://localhost:1433;databaseName=DB_BigWebservice</property>
        <property name="hibernate.connection.username">sa</property>
        <property name="hibernate.show_sql">true</property>
        <property name="hibernate.current_session_context_class">thread</property>
        <mapping class="entity.Product" />
    </session-factory>
</hibernate-configuration>

```

Cài đặt file HibernateUtil.java



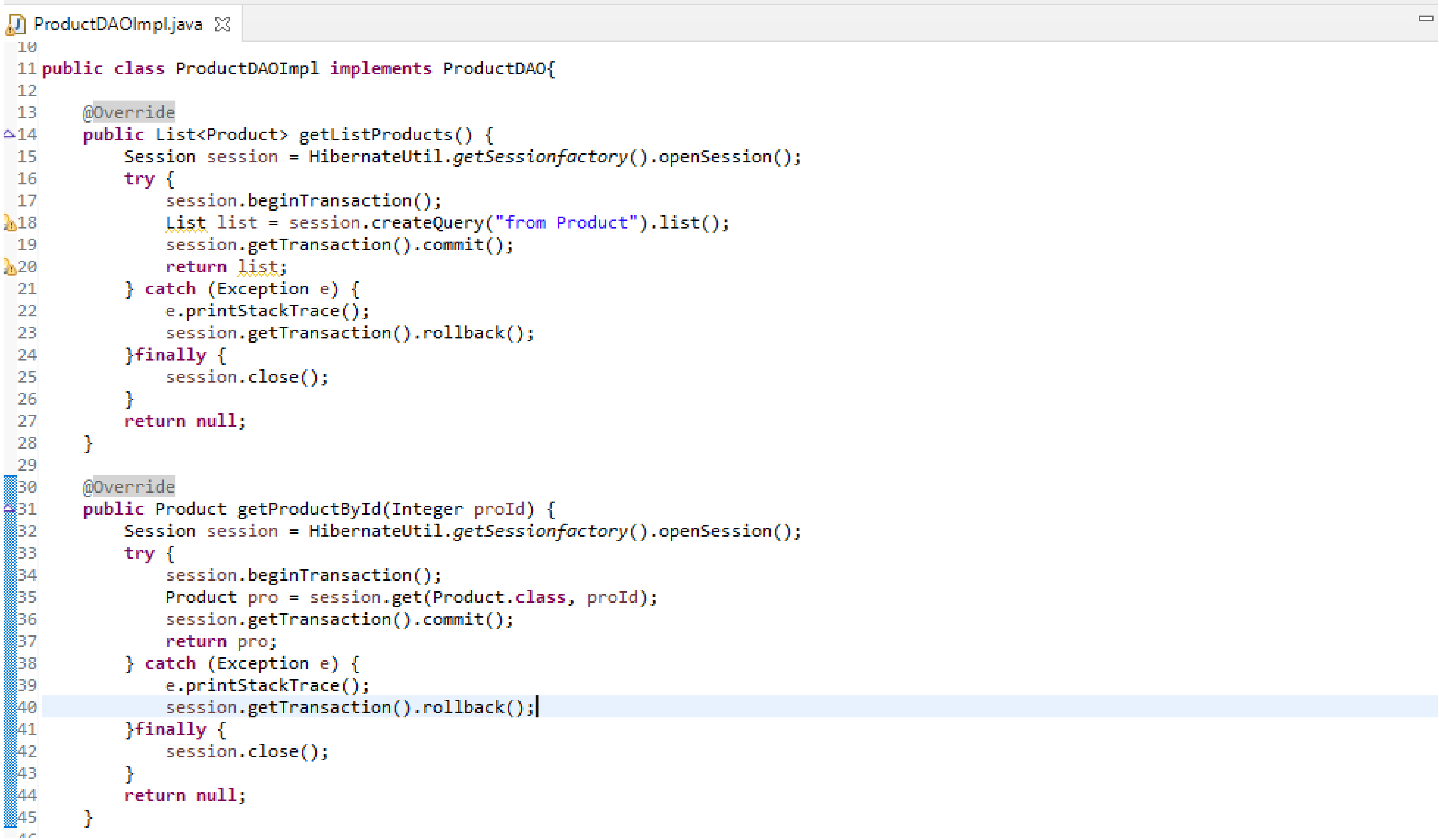
```java
package util;

import org.hibernate.SessionFactory;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;
import org.hibernate.service.ServiceRegistry;

public class HibernateUtil {
    private static final SessionFactory sessionFactory;
    static {
        Configuration config = new Configuration().configure();
        ServiceRegistry reg = new StandardServiceRegistryBuilder().applySettings(config.getProperties()).build();
        sessionFactory = config.buildSessionFactory(reg);
    }
    public static SessionFactory getSessionfactory() {
        return sessionFactory;
    }

}

```

Khai báo các hàm trong giao diện DAO



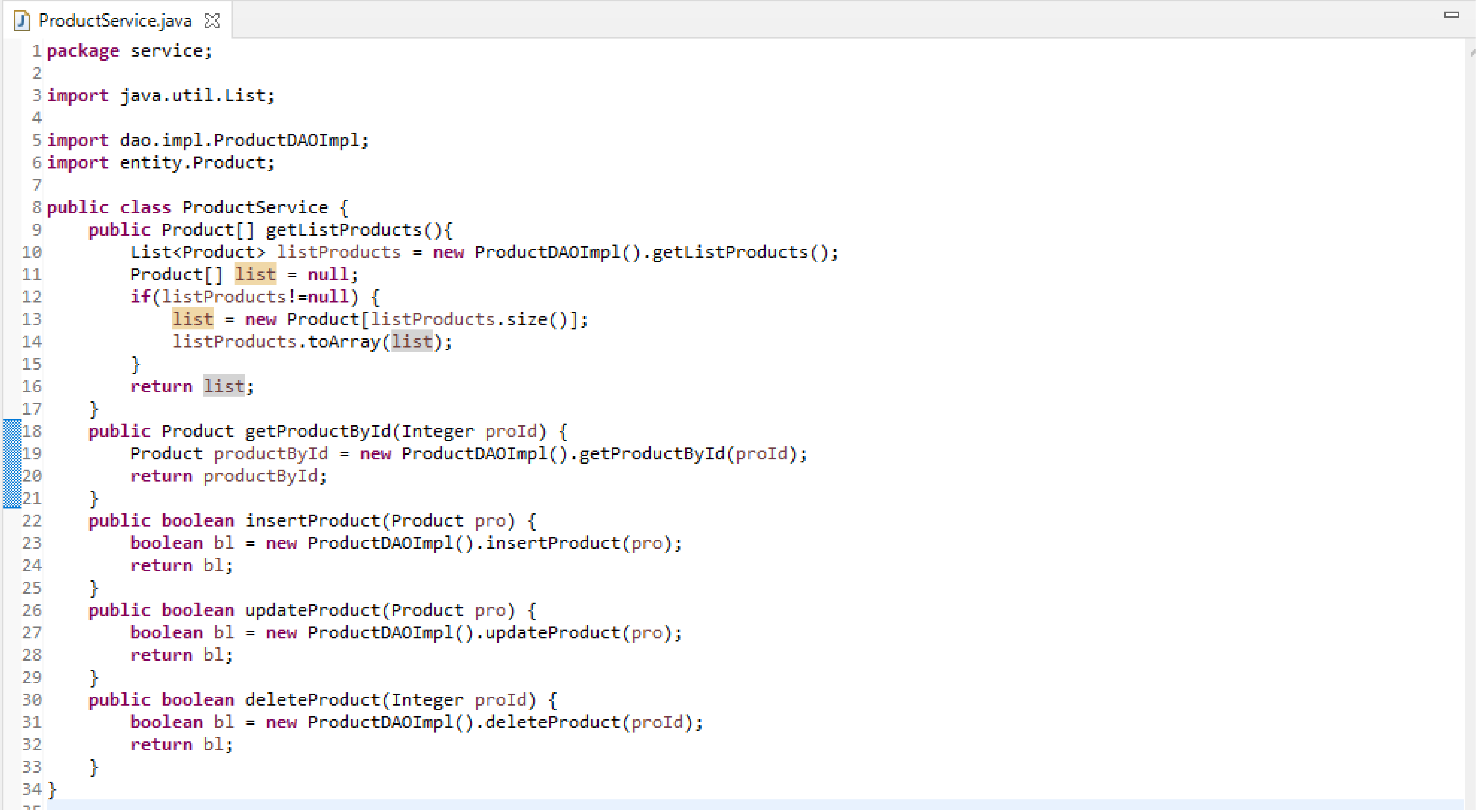
```java
package dao;

import java.util.List;

import entity.Product;

public interface ProductDAO {
    public List<Product> getListProducts();
    public Product getProductById(Integer proId);
    public boolean insertProduct(Product pro);
    public boolean updateProduct(Product pro);
    public boolean deleteProduct(Integer proId);
}
```

Cài đặt các hàm trong lớp thực thi DAO



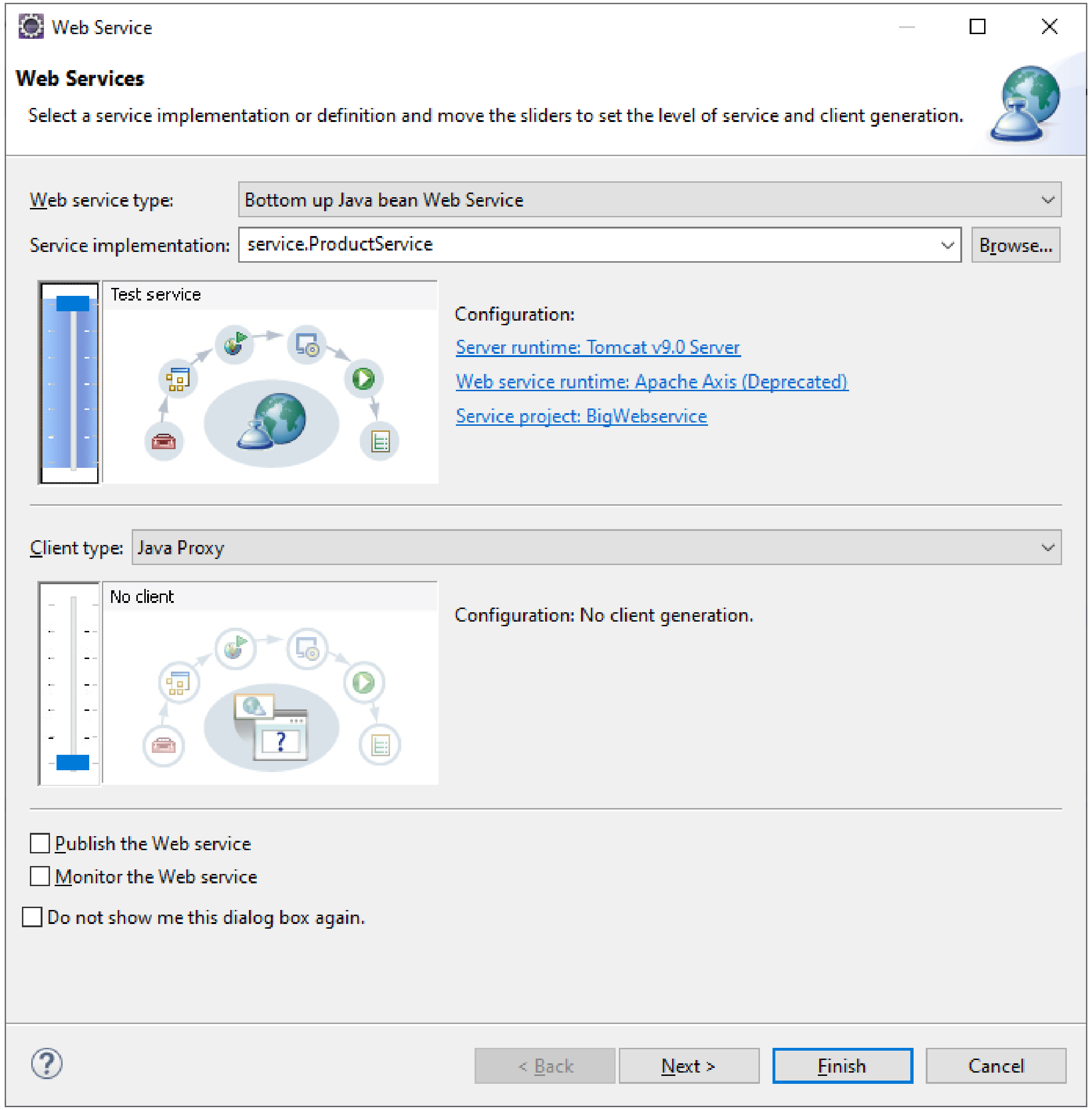
```java
public class ProductDAOImpl implements ProductDAO{

    @Override
    public List<Product> getListProducts() {
        Session session = HibernateUtil.getSessionfactory().openSession();
        try {
            session.beginTransaction();
            List list = session.createQuery("from Product").list();
            session.getTransaction().commit();
            return list;
        } catch (Exception e) {
            e.printStackTrace();
            session.getTransaction().rollback();
        }finally {
            session.close();
        }
        return null;
    }

    @Override
    public Product getProductById(Integer proId) {
        Session session = HibernateUtil.getSessionfactory().openSession();
        try {
            session.beginTransaction();
            Product pro = session.get(Product.class, proId);
            session.getTransaction().commit();
            return pro;
        } catch (Exception e) {
            e.printStackTrace();
            session.getTransaction().rollback();
        }finally {
            session.close();
        }
        return null;
    }
```

Tạo lớp ProductService để cài đặt thành web services



```java
package service;

import java.util.List;

import dao.impl.ProductDAOImpl;
import entity.Product;

public class ProductService {
    public Product[] getListProducts(){
        List<Product> listProducts = new ProductDAOImpl().getListProducts();
        Product[] list = null;
        if(listProducts!=null) {
            list = new Product[listProducts.size()];
            listProducts.toArray(list);
        }
        return list;
    }
    public Product getProductById(Integer proId) {
        Product productById = new ProductDAOImpl().getProductById(proId);
        return productById;
    }
    public boolean insertProduct(Product pro) {
        boolean bl = new ProductDAOImpl().insertProduct(pro);
        return bl;
    }
    public boolean updateProduct(Product pro) {
        boolean bl = new ProductDAOImpl().updateProduct(pro);
        return bl;
    }
    public boolean deleteProduct(Integer proId) {
        boolean bl = new ProductDAOImpl().deleteProduct(proId);
        return bl;
    }
}
```

Tạo Big Web services cho lớp ProductService



Web Services

Select a service implementation or definition and move the sliders to set the level of service and client generation.

Web service type: Bottom up Java bean Web Service

Service implementation: service.ProductService

Test service

Configuration:

Server runtime: Tomcat v9.0 Server

Web service runtime: Apache Axis (Deprecated)

Service project: BigWebservice

Client type: Java Proxy

No client

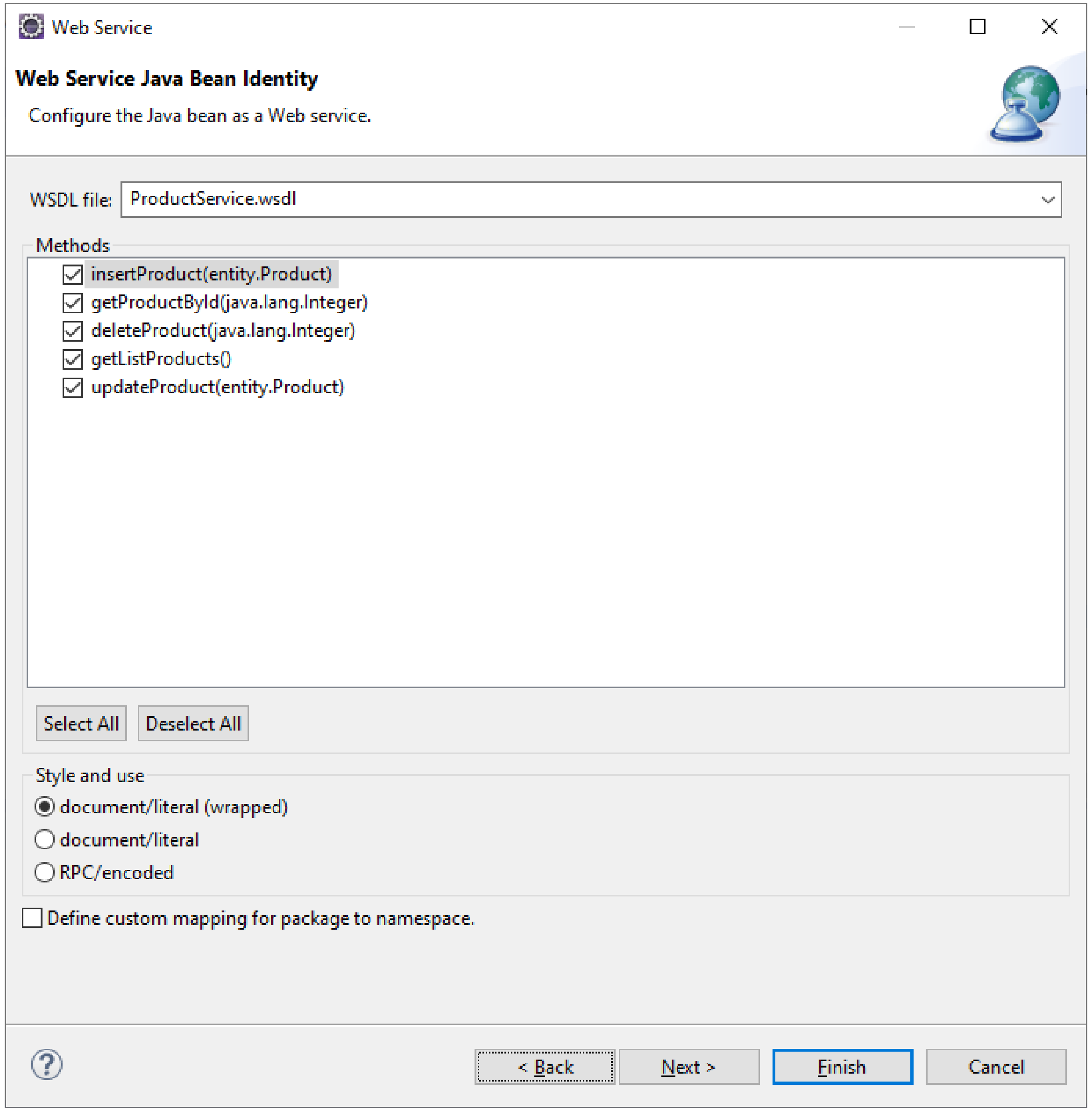
Configuration: No client generation.

☐ Publish the Web service

☐ Monitor the Web service

☐ Do not show me this dialog box again.

Chọn các phương thức được cung cấp



Web Service Java Bean Identity

Configure the Java bean as a Web service.

WSDL file: ProductService.wsdl

Methods

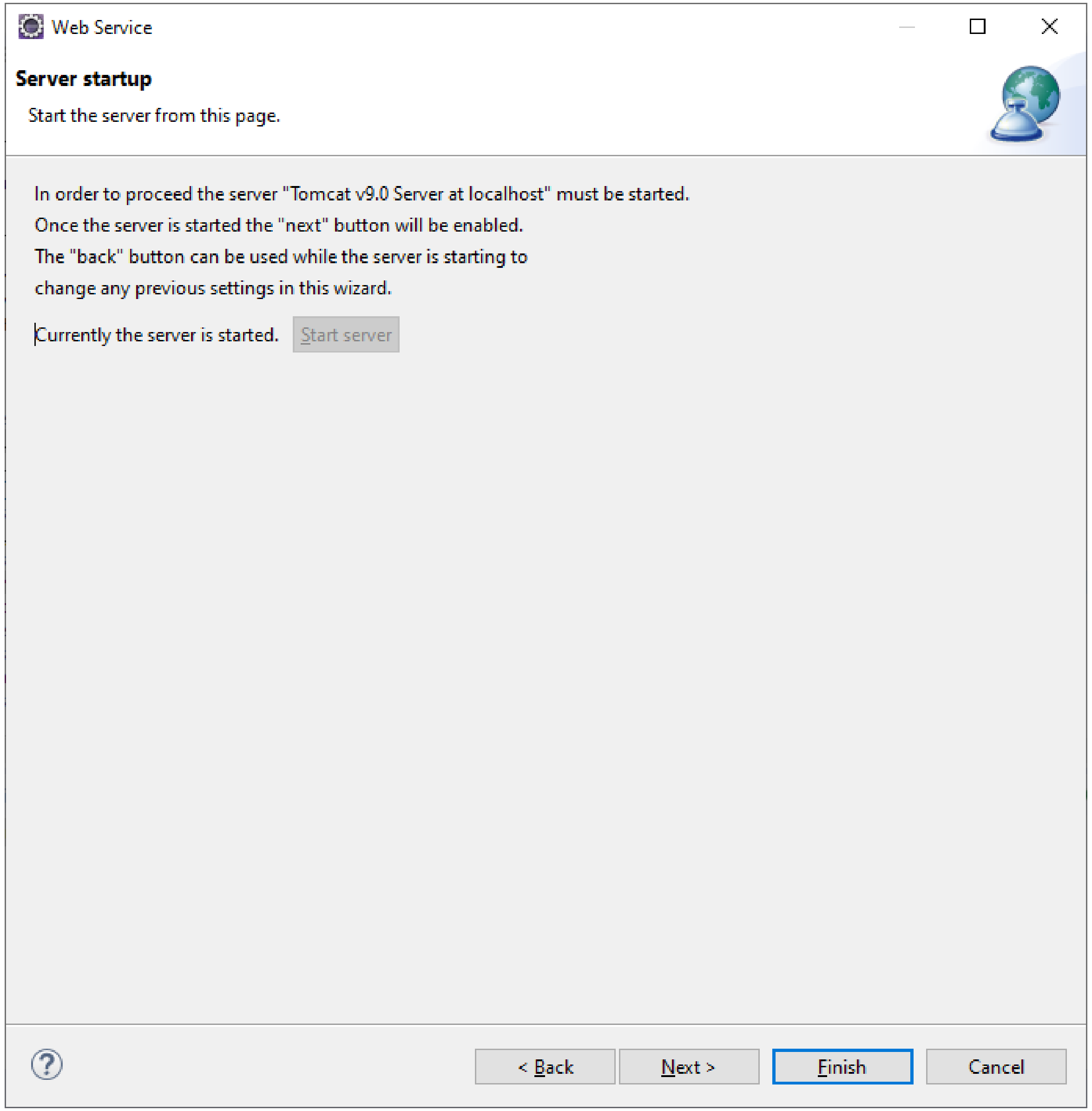
- ☑ insertProduct(entity.Product)
- ☑ getProductById(java.lang.Integer)
- ☑ deleteProduct(java.lang.Integer)
- ☑ getListProducts()
- ☑ updateProduct(entity.Product)

Style and use

- ● document/literal (wrapped)
- ○ document/literal
- ○ RPC/encoded

☐ Define custom mapping for package to namespace.

Start server



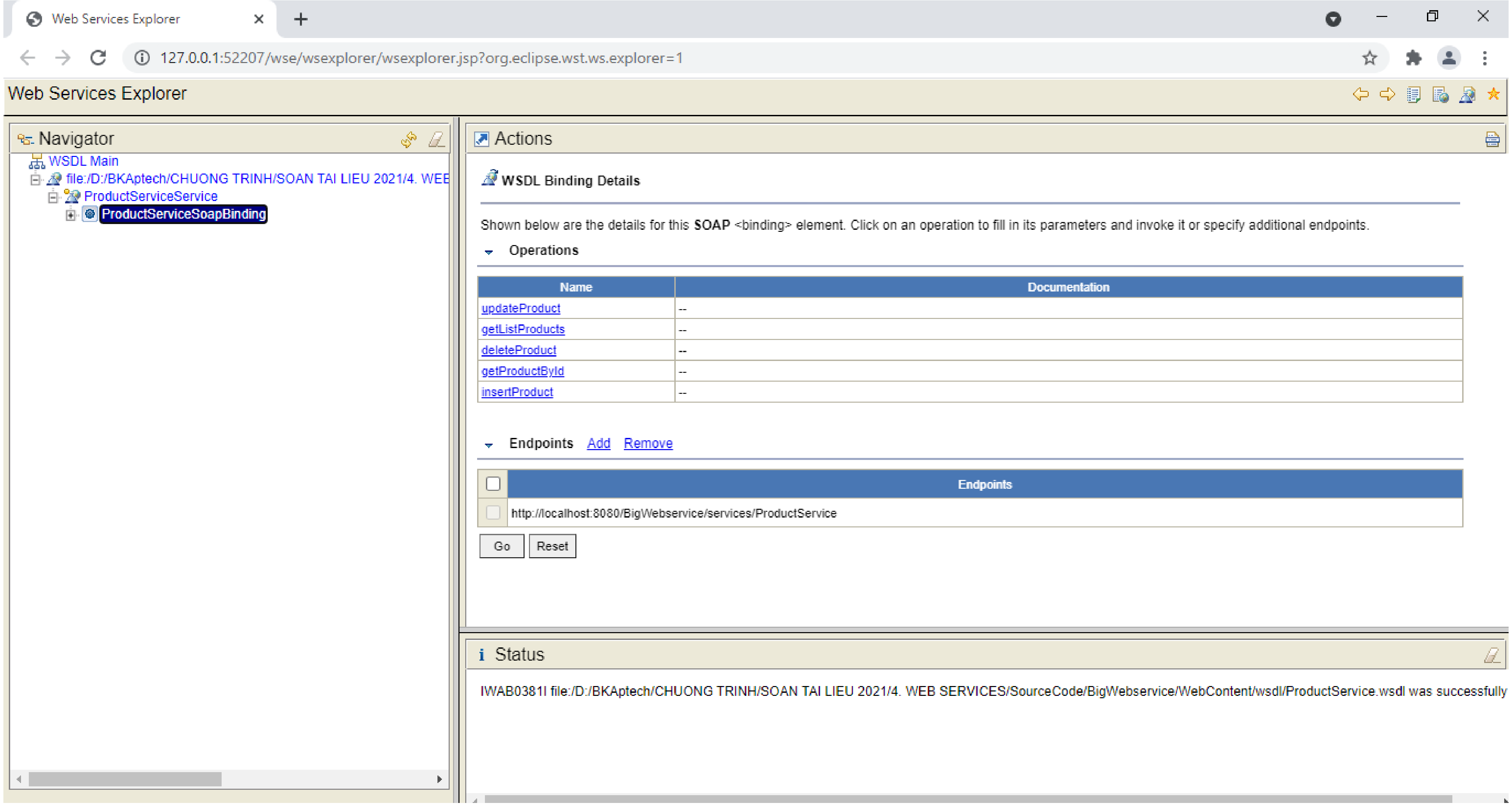
Server startup

Start the server from this page.

In order to proceed the server "Tomcat v9.0 Server at localhost" must be started.
Once the server is started the "next" button will be enabled.
The "back" button can be used while the server is starting to
change any previous settings in this wizard.

Currently the server is started.

Kết quả khi Big Web services được deploy thành công



Web Services Explorer
127.0.0.1:52207/wse/wsexplorer/wsexplorer.jsp?org.eclipse.wst.ws.explorer=1
Web Services Explorer
Navigator
WSDL Main
file:/D:/BKAptech/CHUONG TRINH/SOAN TAI LIEU 2021/4. WEE
ProductServiceService
ProductServiceSoapBinding
Actions
Status

WSDL Binding Details

Shown below are the details for this **SOAP** <binding> element. Click on an operation to fill in its parameters and invoke it or specify additional endpoints.

Operations

Name	Documentation
updateProduct	--
getListProducts	--
deleteProduct	--
getProductById	--
insertProduct	--

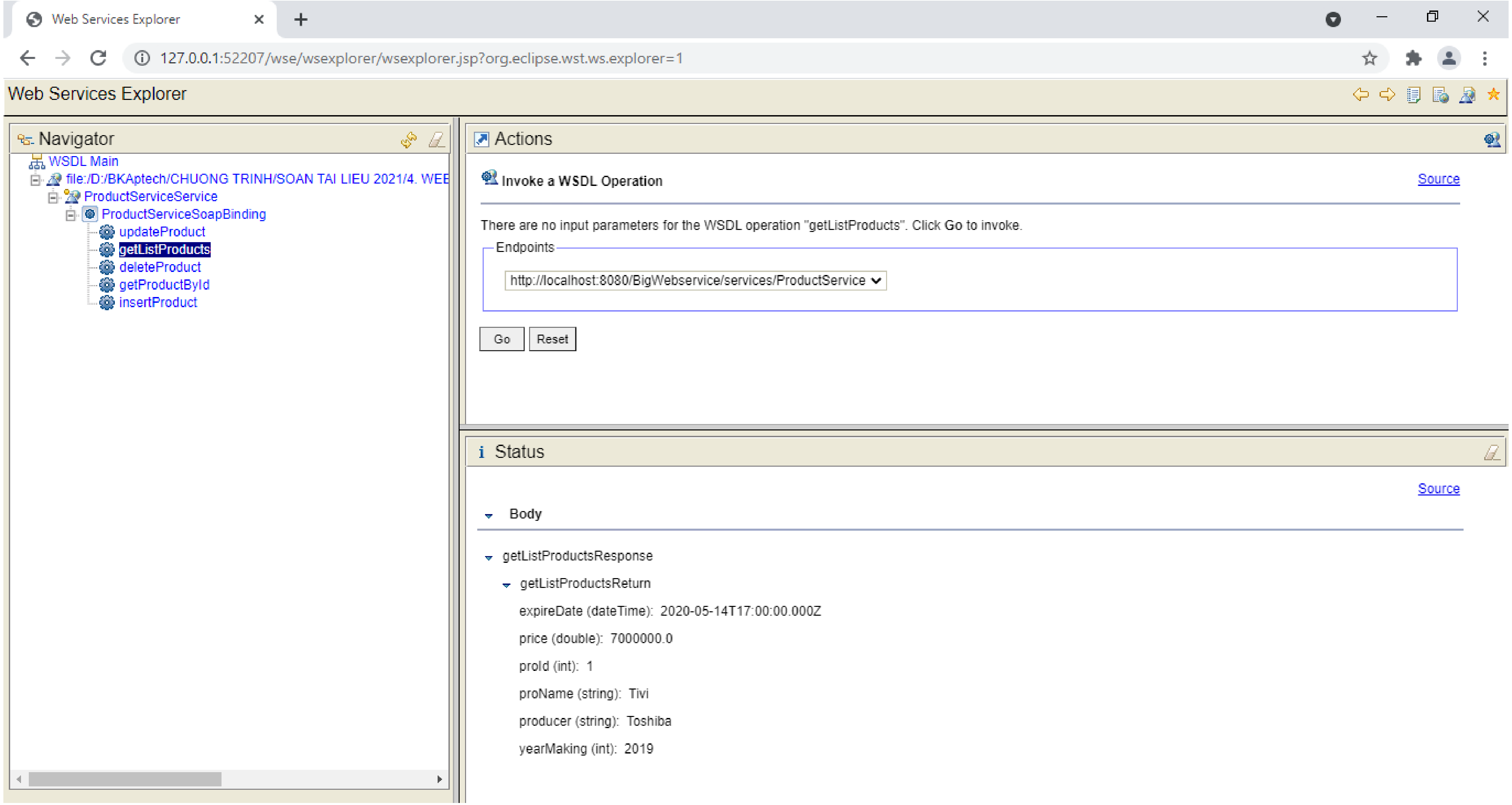
Endpoints Add Remove

□	Endpoints
□	http://localhost:8080/BigWebservice/services/ProductService

Go Reset

IWAB0381I file:/D:/BKAptech/CHUONG TRINH/SOAN TAI LIEU 2021/4. WEB SERVICES/SourceCode/BigWebservice/WebContent/wsdl/ProductService.wsdl was successfully

Test chức năng get data



Invoke a WSDL Operation

There are no input parameters for the WSDL operation "getListProducts". Click Go to invoke.

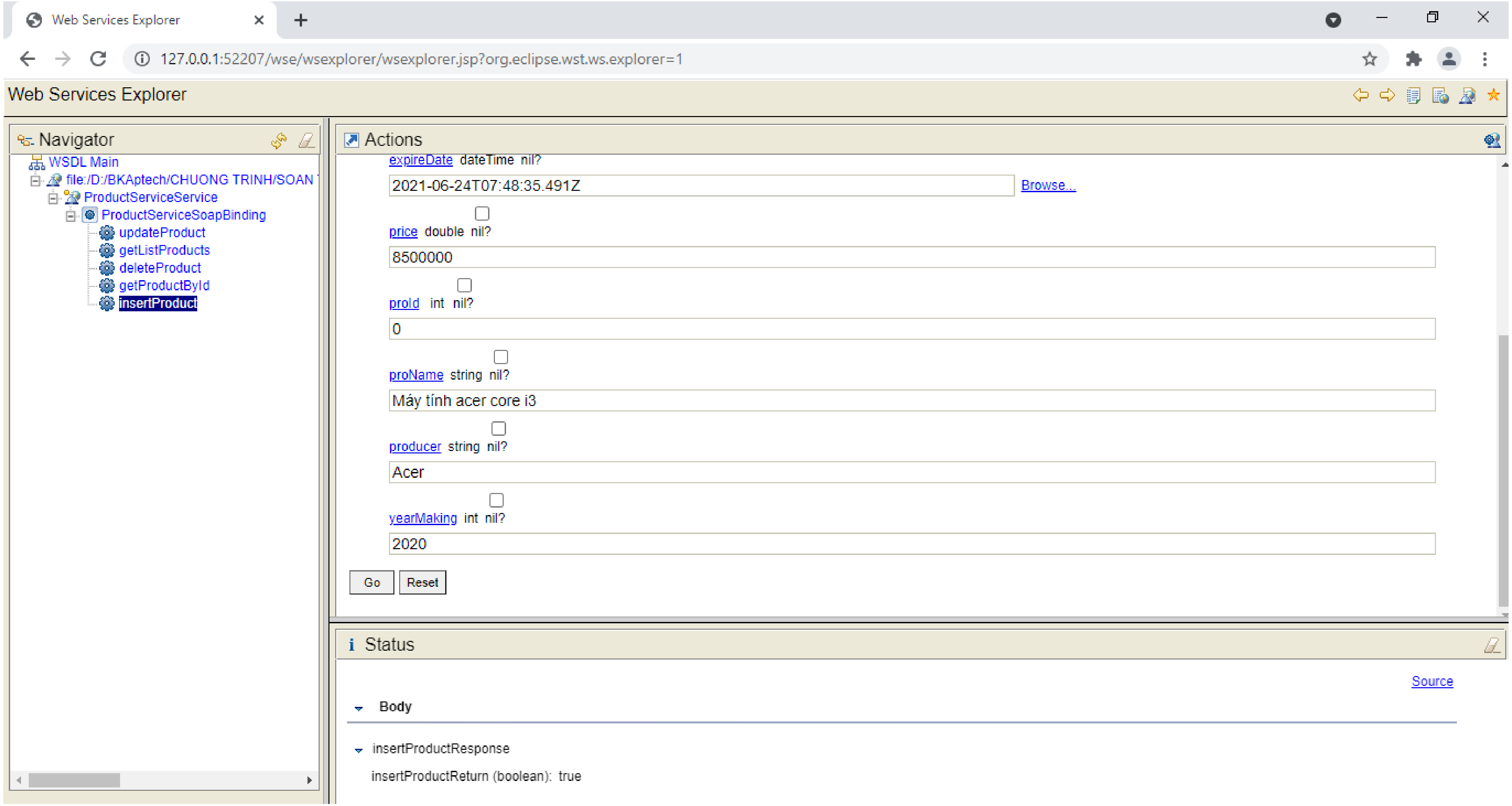
Endpoints

http://localhost:8080/BigWebservice/services/ProductService

Body

```
getListProductsResponse
  getListProductsReturn
    expireDate (dateTime):  2020-05-14T17:00:00.000Z
    price (double):  7000000.0
    proId (int):  1
    proName (string):  Tivi
    producer (string):  Toshiba
    yearMaking (int):  2019
```

Test chức năng insert data



expireDate dateTime nil?

2021-06-24T07:48:35.491Z

□ price double nil?

8500000

□ proId int nil?

0

□ proName string nil?

Máy tính acer core i3

□ producer string nil?

Acer

□ yearMaking int nil?

2020

Body

insertProductResponse

insertProductReturn (boolean): true

HỎI ĐÁP

TRẢI NGHIỆM THỰC HÀNH

HỆ THỐNG ĐÀO TẠO CNTT QUỐC TẾ BACHKHOA - APTECH

TRÂN TRỌNG CẢM ƠN!

238 Hoàng Quốc Việt, Bắc Từ Liêm, Hà Nội

☎ 0968.27.6996

✉ tuyensinh@bachkhoa-aptech.edu.vn

www.bachkhoa-aptech.edu.vn