

1. Préliminaires

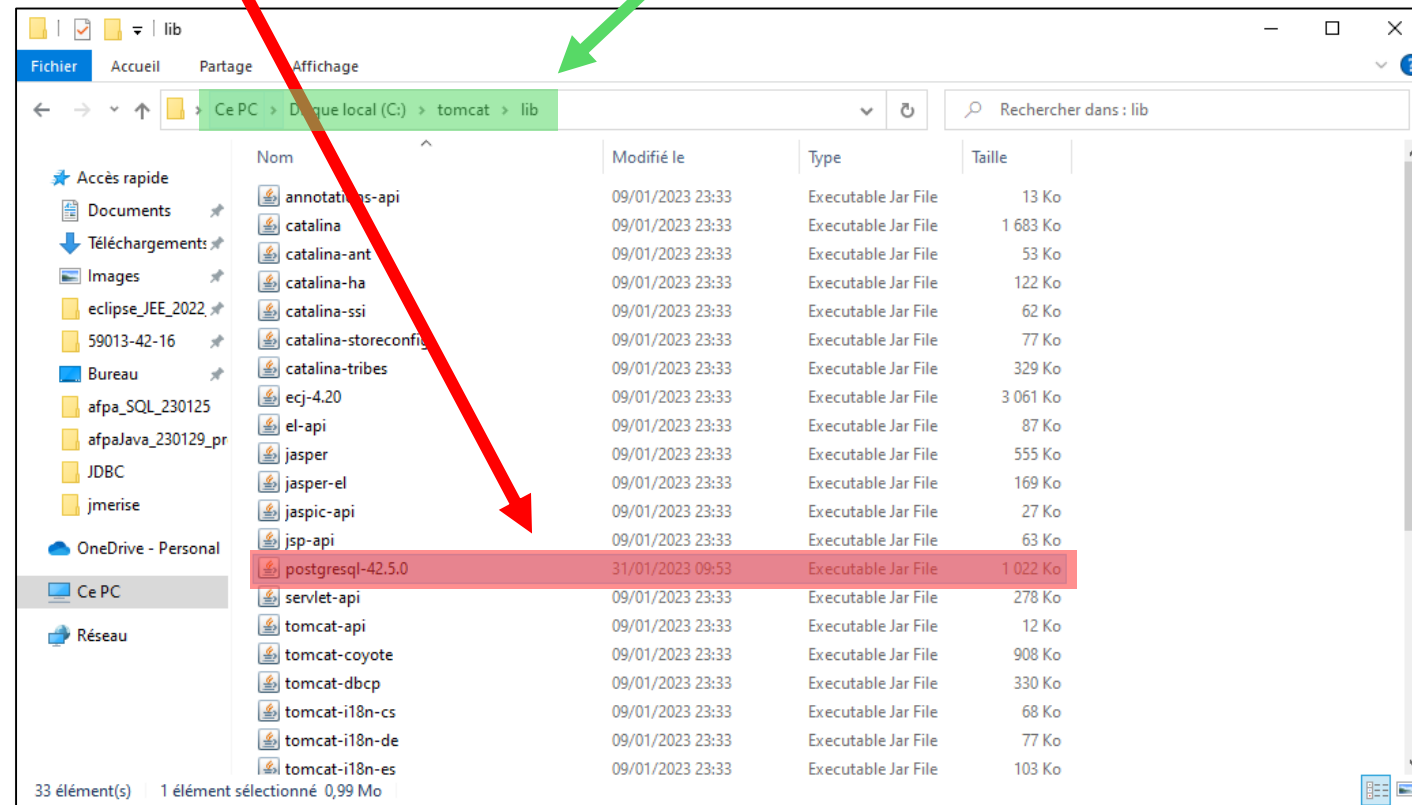
2. Création BDD

3. Servlet

4. ConnectionPGSQL

5. JSP

D'abord, placer le driver postgresql ('postgresql-xx.x.x.jar') dans le dossier 'lib' du répertoire de votre serveur Tomcat



Ajouter le .jar externe du driver postgresql dans le Build Path du projet.

The screenshot illustrates the steps to add the PostgreSQL driver JAR to the project's build path in Eclipse IDE. The process involves navigating through the Project Explorer, Properties dialog, Java Build Path settings, and the JAR Selection dialog to locate and add the 'postgresql-42.5.0.jar' file from the Tomcat lib directory.

Properties Dialog - Java Build Path

Libraries

- Source
- Project Libraries
- External Libraries
- Web App Libraries

JAR Selection

Organiser: Nouveau dossier

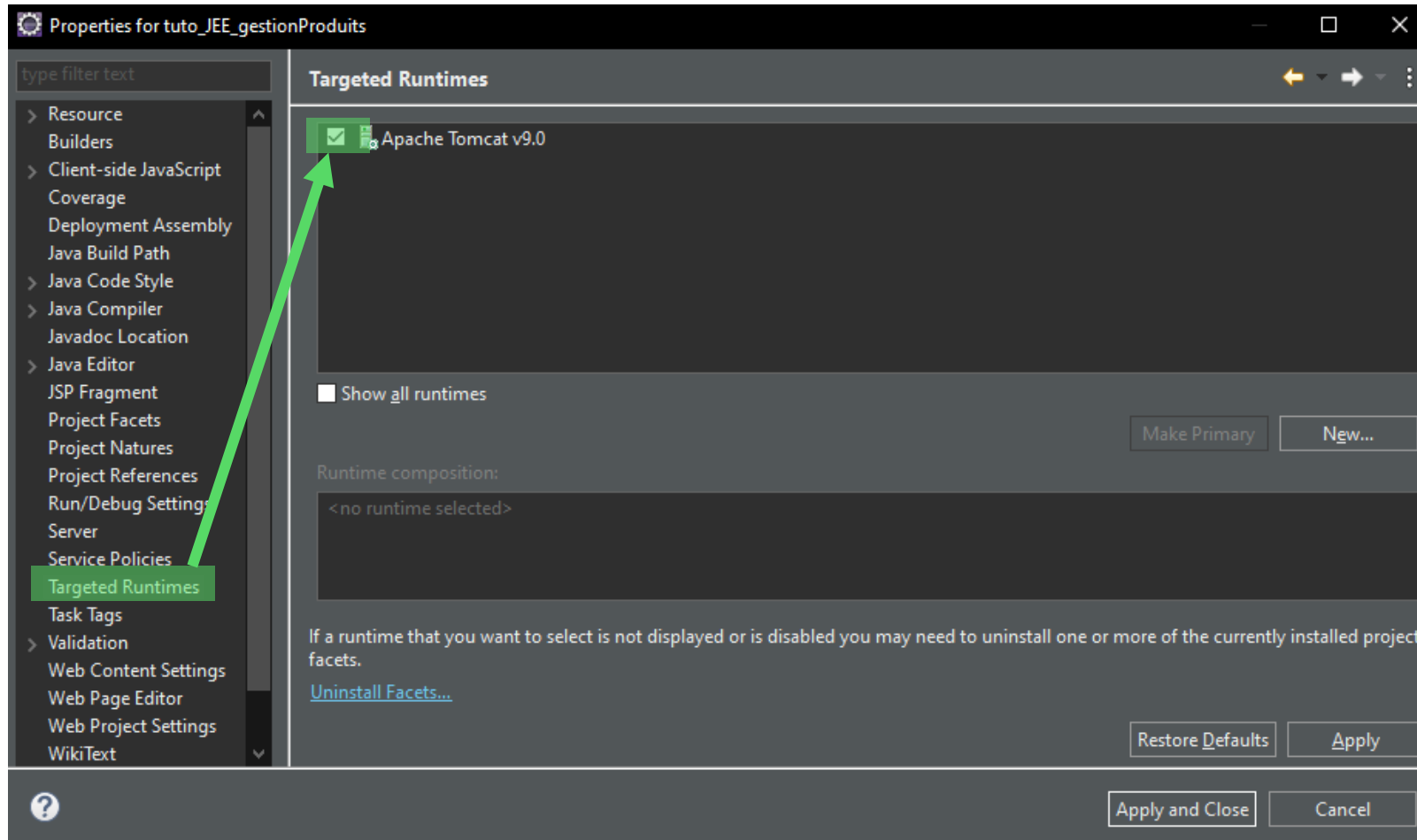
| Nom | Modifié le | Type | Taille |
|-------------------|------------------|---------------------|----------|
| catalina-tribes | 09/01/2023 23:33 | Executable Jar File | 329 Ko |
| ecj-4.20 | 09/01/2023 23:33 | Executable Jar File | 3 061 Ko |
| el-api | 09/01/2023 23:33 | Executable Jar File | 87 Ko |
| jasper | 09/01/2023 23:33 | Executable Jar File | 555 Ko |
| jasper-el | 09/01/2023 23:33 | Executable Jar File | 169 Ko |
| jaspic-api | 09/01/2023 23:33 | Executable Jar File | 27 Ko |
| jsp-api | 09/01/2023 23:33 | Executable Jar File | 63 Ko |
| postgresql-42.5.0 | 31/01/2023 09:53 | Executable Jar File | 1 022 Ko |
| servlet-api | 09/01/2023 23:33 | Executable Jar File | 278 Ko |
| tomcat-api | 09/01/2023 23:33 | Executable Jar File | 12 Ko |
| tomcat-coyote | 09/01/2023 23:33 | Executable Jar File | 908 Ko |
| tomcat-dbcp | 09/01/2023 23:33 | Executable Jar File | 330 Ko |
| tomcat-i18n-cs | 09/01/2023 23:33 | Executable Jar File | 68 Ko |
| tomcat-i18n-de | 09/01/2023 23:33 | Executable Jar File | 77 Ko |
| tomcat-i18n-es | 09/01/2023 23:33 | Executable Jar File | 103 Ko |
| tomcat-i18n-fr | 09/01/2023 23:33 | Executable Jar File | 145 Ko |

Nom du fichier: postgresql-42.5.0

Type: *.jar;*.zip

Ouvrir

Activer le serveur Tomcat dans la section 'Targeted Runtimes'



Création de la database 'gestion_produits' et de sa table 'produits' dans postgresql

Création de la database 'gestion_produits' et de sa table 'produits' dans postgresql

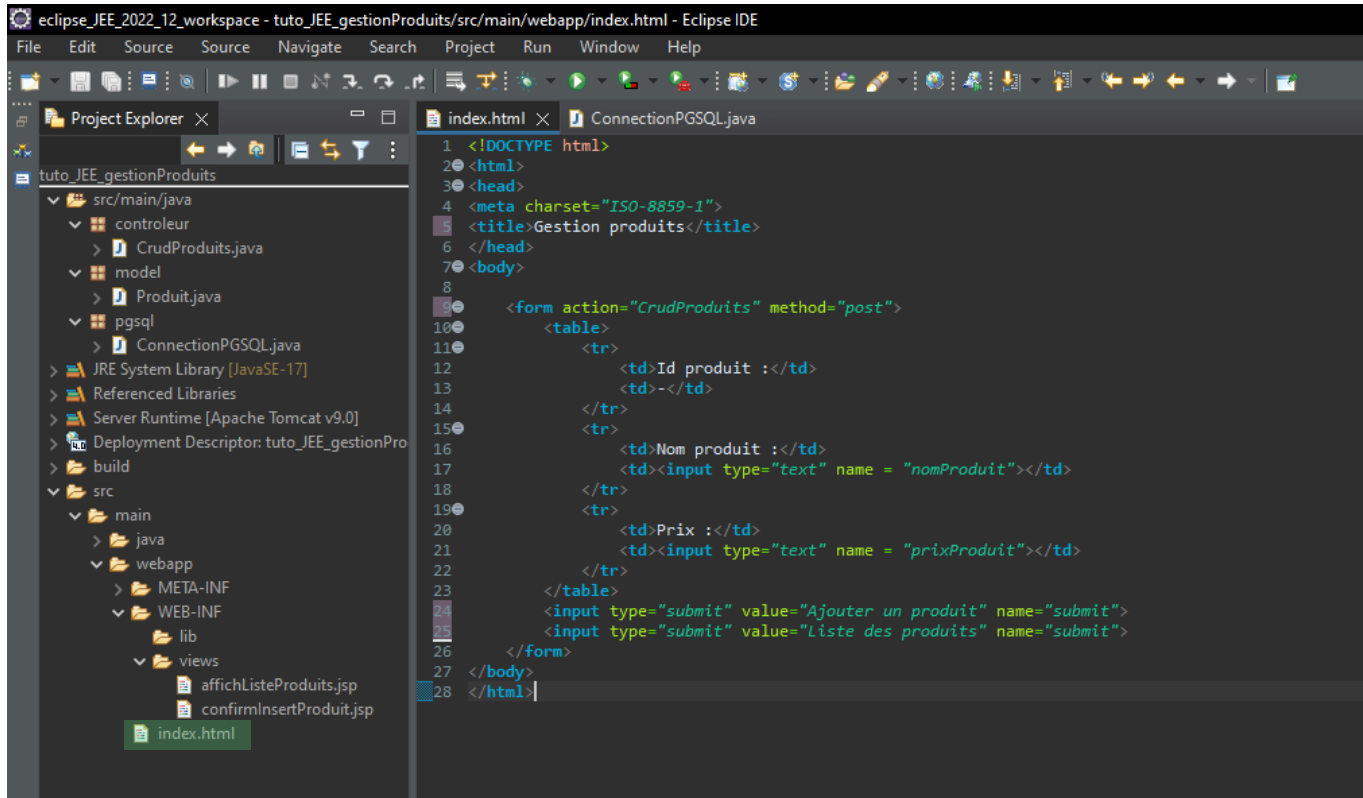
```
empserv=# create database gestion_produits
empserv=# ;
CREATE DATABASE
empserv=# \c gestion_produits
Vous êtes maintenant connecté à la base de données « gestion_produits » en tant qu'utilisateur « postgres ».
gestion_produits=# create table produits (id serial primary key, nom varchar(50), prix float);
CREATE TABLE
gestion_produits=# \d produits
```

| Colonne | Type | Collationnement | NULL-able | Par défaut |
|---------|-----------------------|-----------------|-----------|--------------------------------------|
| id | integer | | not null | nextval('produits_id_seq'::regclass) |
| nom | character varying(50) | | | |
| prix | double precision | | | |

```
Index :
    "produits_pkey" PRIMARY KEY, btree (id)

gestion_produits=#
```

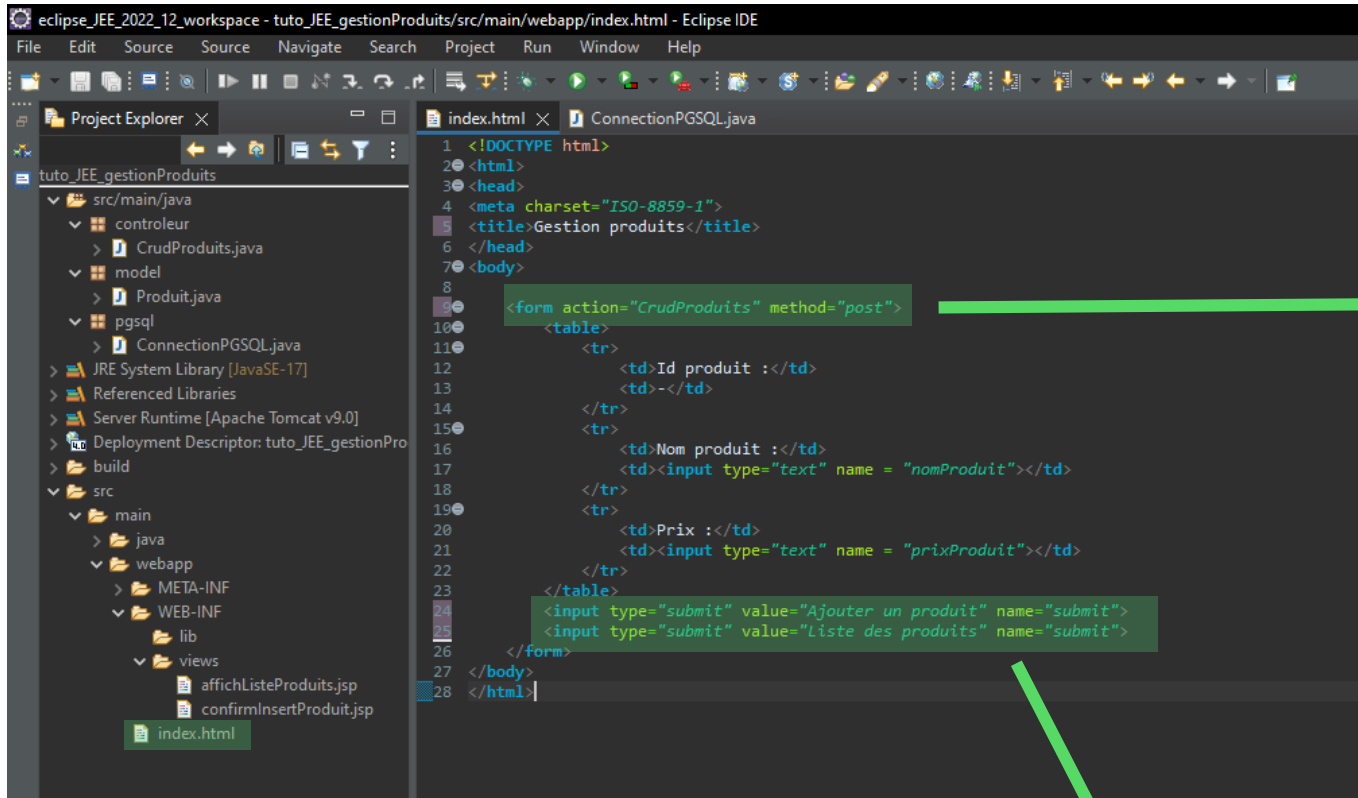

Index.html



The screenshot shows the Eclipse IDE interface. The Project Explorer on the left displays the project structure for 'tuto_JEE_gestionProduits'. The main Editor window shows the 'index.html' file, which contains an HTML form for adding and listing products. The form includes a table with columns for 'Id produit', 'Nom produit', and 'Prix', and two submit buttons: 'Ajouter un produit' and 'Liste des produits'.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="ISO-8859-1">
5 <title>Gestion produits</title>
6 </head>
7 <body>
8
9 <form action="CrudProduits" method="post">
10 <table>
11 <tr>
12 <td>Id produit :</td>
13 <td>-</td>
14 </tr>
15 <tr>
16 <td>Nom produit :</td>
17 <td><input type="text" name = "nomProduit"></td>
18 </tr>
19 <tr>
20 <td>Prix :</td>
21 <td><input type="text" name = "prixProduit"></td>
22 </tr>
23 </table>
24 <input type="submit" value="Ajouter un produit" name="submit">
25 <input type="submit" value="Liste des produits" name="submit">
26 </form>
27 </body>
28 </html>
```

Index.html



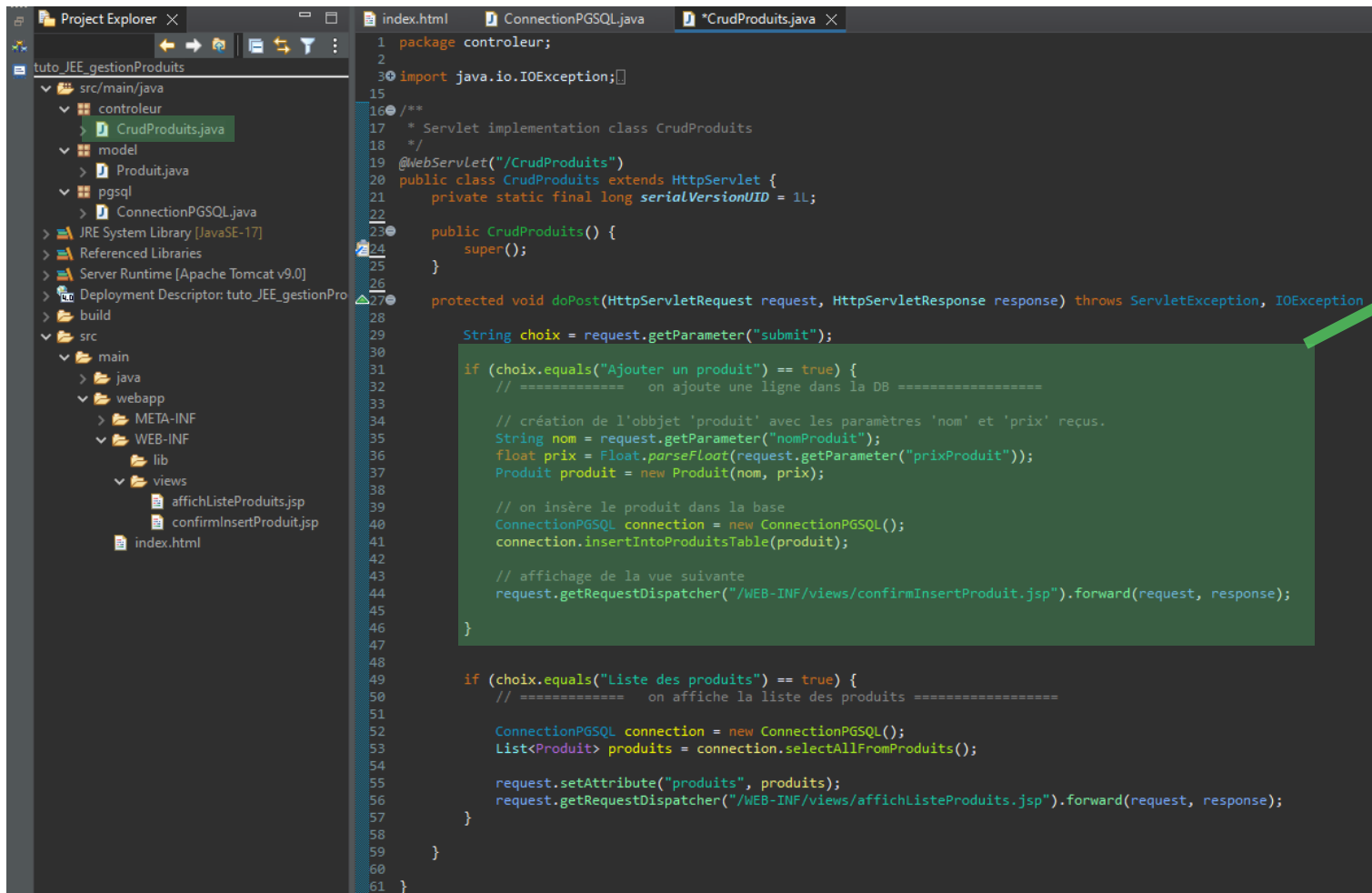
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="ISO-8859-1">
5 <title>Gestion produits</title>
6 </head>
7 <body>
8
9 <form action="CrudProduits" method="post">
10 <table>
11 <tr>
12 <td>Id produit :</td>
13 <td>-</td>
14 </tr>
15 <tr>
16 <td>Nom produit :</td>
17 <td><input type="text" name = "nomProduit"></td>
18 </tr>
19 <tr>
20 <td>Prix :</td>
21 <td><input type="text" name = "prixProduit"></td>
22 </tr>
23 </table>
24 <input type="submit" value="Ajouter un produit" name="submit">
25 <input type="submit" value="Liste des produits" name="submit">
26 </form>
27 </body>
28 </html>
```

Le formulaire envoie une requête POST vers le Servlet 'CrudProduits'

En fonction de la valeur de 'submit' le servlet effectuera soit :

- un ajout de produit dans la BDD
- un affichage de la liste des produits

Le Servlet CrudProduits.java

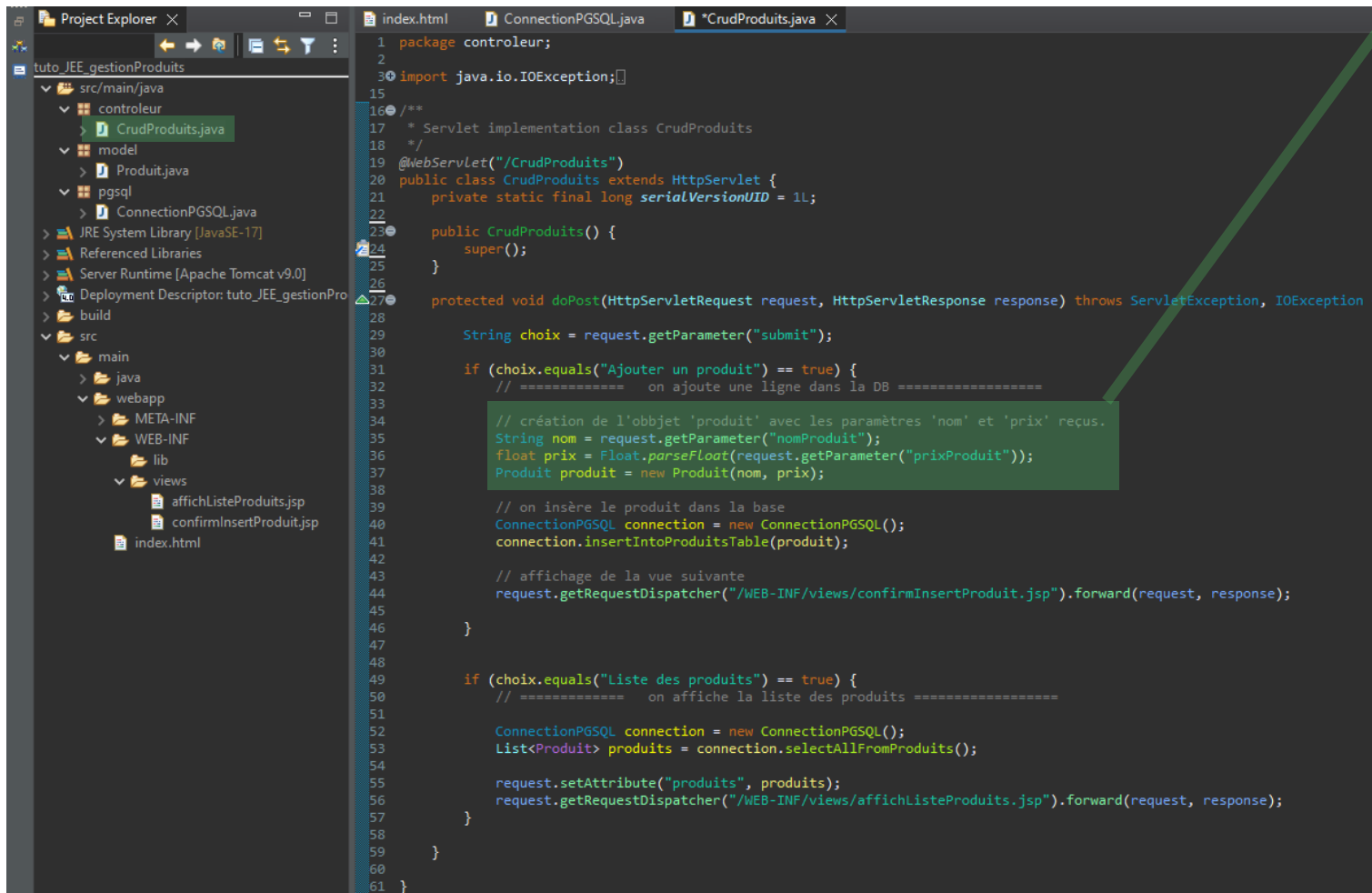


```
1 package controleur;
2
3 import java.io.IOException;
4
5
6
7
8
9
10
11
12
13
14
15 /**
16  * Servlet implementation class CrudProduits
17  */
18 @WebServlet("/CrudProduits")
19 public class CrudProduits extends HttpServlet {
20     private static final long serialVersionUID = 1L;
21
22     public CrudProduits() {
23         super();
24     }
25
26     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
27
28         String choix = request.getParameter("submit");
29
30         if (choix.equals("Ajouter un produit") == true) {
31             // ===== on ajoute une ligne dans la DB =====
32
33             // création de l'objet 'produit' avec les paramètres 'nom' et 'prix' reçus.
34             String nom = request.getParameter("nomProduit");
35             float prix = Float.parseFloat(request.getParameter("prixProduit"));
36             Produit produit = new Produit(nom, prix);
37
38             // on insère le produit dans la base
39             ConnectionPGSQL connection = new ConnectionPGSQL();
40             connection.insertIntoProduitsTable(produit);
41
42             // affichage de la vue suivante
43             request.getRequestDispatcher("/WEB-INF/views/confirmInsertProduit.jsp").forward(request, response);
44         }
45
46         if (choix.equals("Liste des produits") == true) {
47             // ===== on affiche la liste des produits =====
48
49             ConnectionPGSQL connection = new ConnectionPGSQL();
50             List<Produit> produits = connection.selectAllFromProduits();
51
52             request.setAttribute("produits", produits);
53             request.getRequestDispatcher("/WEB-INF/views/affichListeProduits.jsp").forward(request, response);
54         }
55     }
56 }
57
58
59
60
61 }
```

On se place dans le cas où on veut ajouter un produit dans la BDD

Le Servlet CrudProduits.java

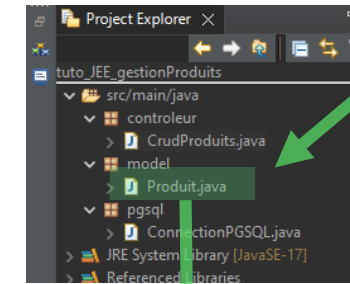
Création de l'objet produit selon le modèle Produit



```

1 package controleur;
2
3 import java.io.IOException;
4
5
6
7
8
9
10
11
12
13
14
15
16 /**
17  * Servlet implementation class CrudProduits
18  */
19 @WebServlet("/CrudProduits")
20 public class CrudProduits extends HttpServlet {
21     private static final long serialVersionUID = 1L;
22
23     public CrudProduits() {
24         super();
25     }
26
27     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
28
29         String choix = request.getParameter("submit");
30
31         if (choix.equals("Ajouter un produit") == true) {
32             // ===== on ajoute une ligne dans la DB =====
33
34             // création de l'objet 'produit' avec les paramètres 'nom' et 'prix' reçus.
35             String nom = request.getParameter("nomProduit");
36             float prix = Float.parseFloat(request.getParameter("prixProduit"));
37             Produit produit = new Produit(nom, prix);
38
39             // on insère le produit dans la base
40             ConnectionPGSQL connection = new ConnectionPGSQL();
41             connection.insertIntoProduitsTable(produit);
42
43             // affichage de la vue suivante
44             request.getRequestDispatcher("/WEB-INF/views/confirmInsertProduit.jsp").forward(request, response);
45
46         }
47
48         if (choix.equals("Liste des produits") == true) {
49             // ===== on affiche la liste des produits =====
50
51             ConnectionPGSQL connection = new ConnectionPGSQL();
52             List<Produit> produits = connection.selectAllFromProduits();
53
54             request.setAttribute("produits", produits);
55             request.getRequestDispatcher("/WEB-INF/views/affichListeProduits.jsp").forward(request, response);
56
57         }
58     }
59 }
60
61

```

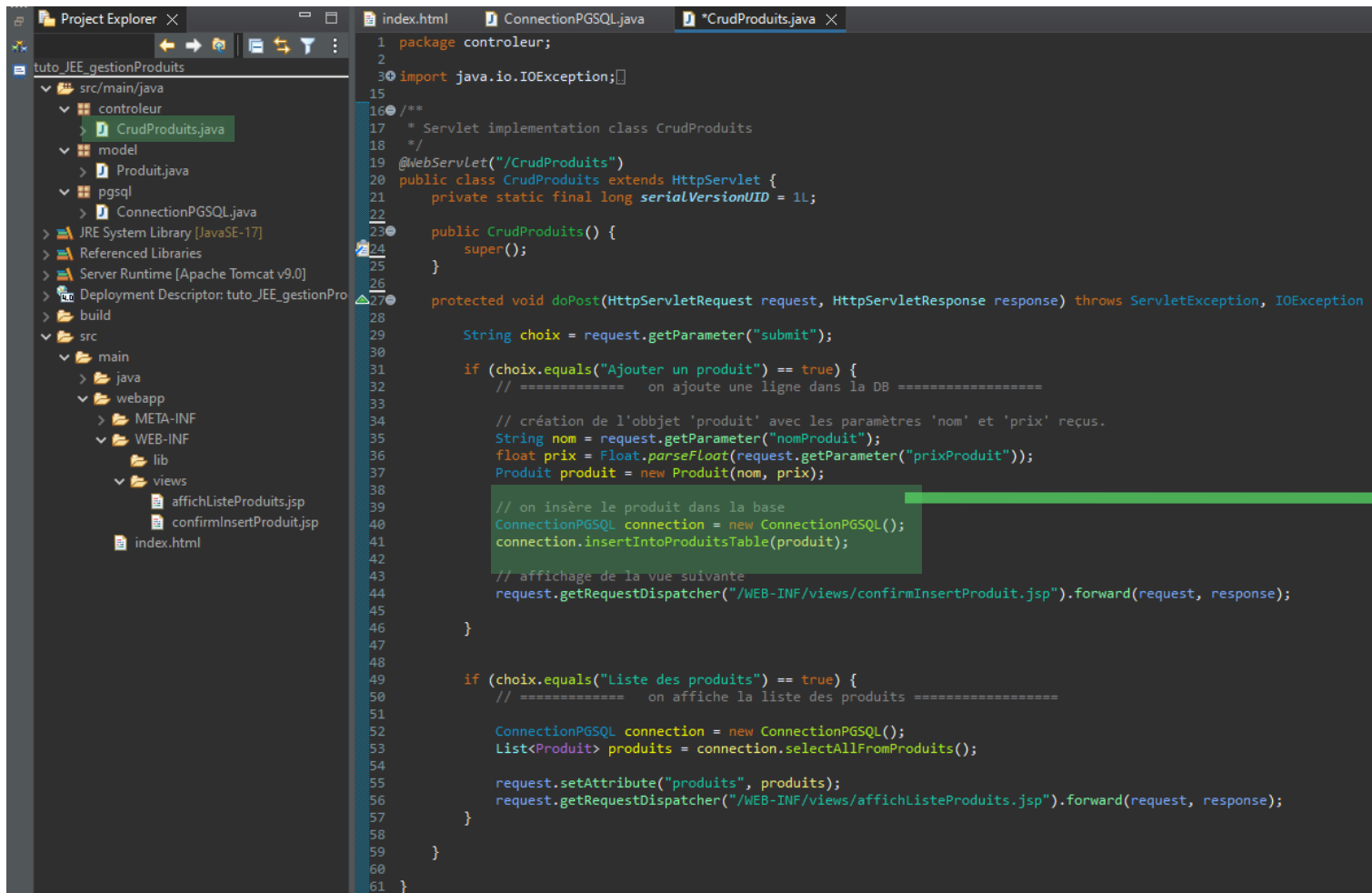


```

1 package model;
2
3 public class Produit {
4
5     int id;
6     String nom;
7     float prix;
8
9     public Produit(String nom, float prix) {
10         this.id = id;
11         this.nom = nom;
12         this.prix = prix;
13     }
14
15     public Produit(int id, String nom, float prix) {
16         this.id = id;
17         this.nom = nom;
18         this.prix = prix;
19     }
20
21     public int getId() {
22         return id;
23     }
24
25     public String getNom() {
26         return nom;
27     }
28
29     public float getPrix() {
30         return prix;
31     }
32
33     public void setNom(String nom) {
34         this.nom = nom;
35     }
36
37     public void setPrix(float prix) {
38         this.prix = prix;
39     }
40
41     public String toString() {
42         return "Produit [id=" + id + ", nom=" + nom + ",
43     }
44
45 }
46

```

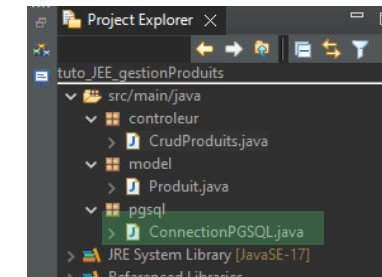
Le Servlet CrudProduits.java



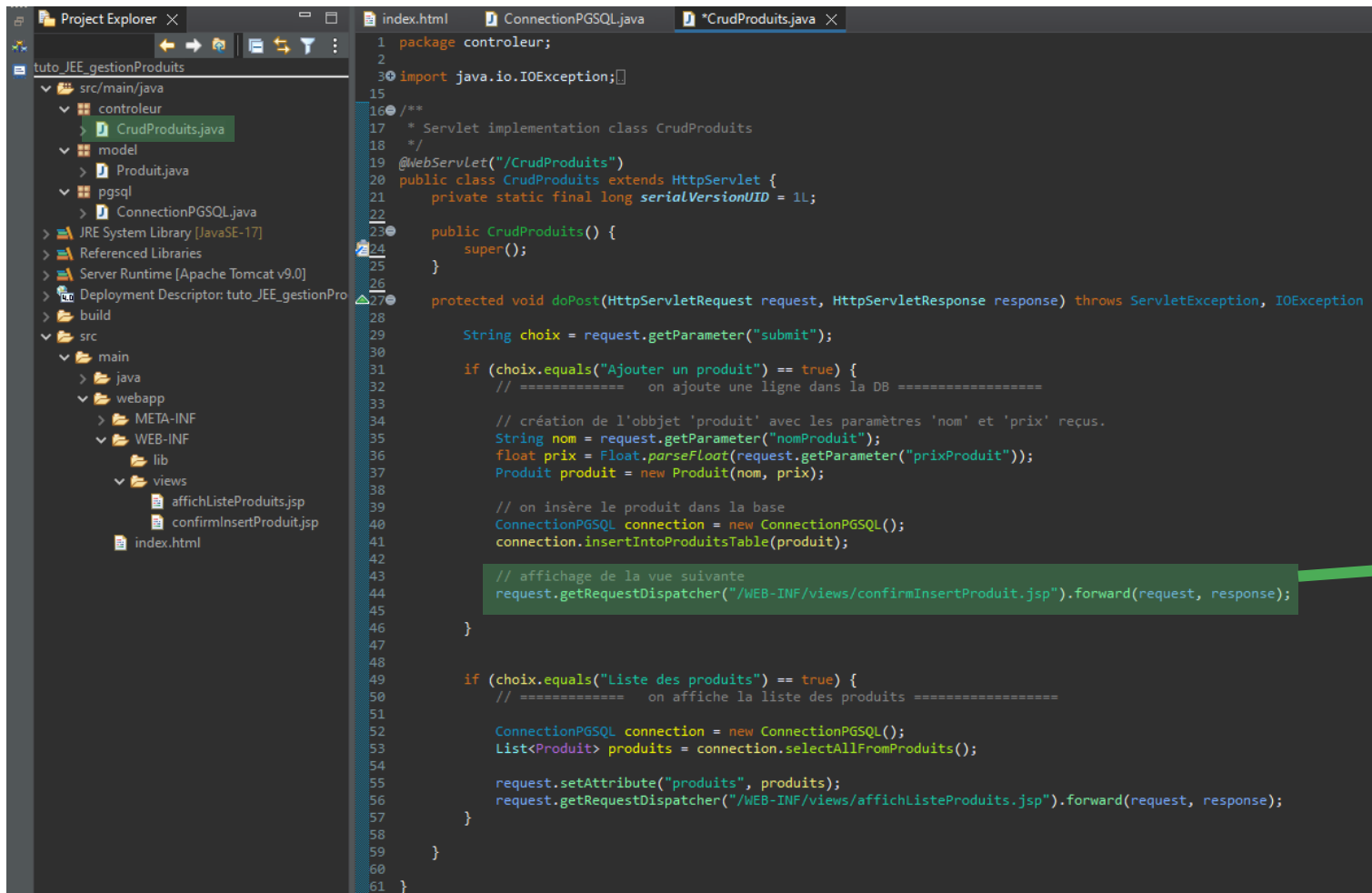
```
1 package controleur;
2
3 import java.io.IOException;
4
5
6
7
8
9
10
11
12
13
14
15 /**
16  * Servlet implementation class CrudProduits
17  */
18 @WebServlet("/CrudProduits")
19 public class CrudProduits extends HttpServlet {
20     private static final long serialVersionUID = 1L;
21
22     public CrudProduits() {
23         super();
24     }
25
26     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
27
28         String choix = request.getParameter("submit");
29
30         if (choix.equals("Ajouter un produit") == true) {
31             // ===== on ajoute une ligne dans la DB =====
32
33             // création de l'objet 'produit' avec les paramètres 'nom' et 'prix' reçus.
34             String nom = request.getParameter("nomProduit");
35             float prix = Float.parseFloat(request.getParameter("prixProduit"));
36             Produit produit = new Produit(nom, prix);
37
38             // on insère le produit dans la base
39             ConnectionPGSQL connection = new ConnectionPGSQL();
40             connection.insertIntoProduitsTable(produit);
41
42             // affichage de la vue suivante
43             request.getRequestDispatcher("/WEB-INF/views/confirmInsertProduit.jsp").forward(request, response);
44         }
45
46         if (choix.equals("Liste des produits") == true) {
47             // ===== on affiche la liste des produits =====
48
49             ConnectionPGSQL connection = new ConnectionPGSQL();
50             List<Produit> produits = connection.selectAllFromProduits();
51
52             request.setAttribute("produits", produits);
53             request.getRequestDispatcher("/WEB-INF/views/affichListeProduits.jsp").forward(request, response);
54         }
55     }
56 }
57
58
59
60
61 }
```

Création d'une connexion
Insertion de l'objet dans la base

Ceci grâce à la classe 'ConnectionPGSQL'

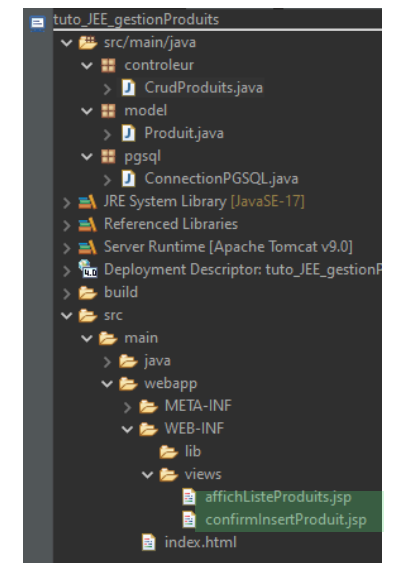


Le Servlet CrudProduits.java

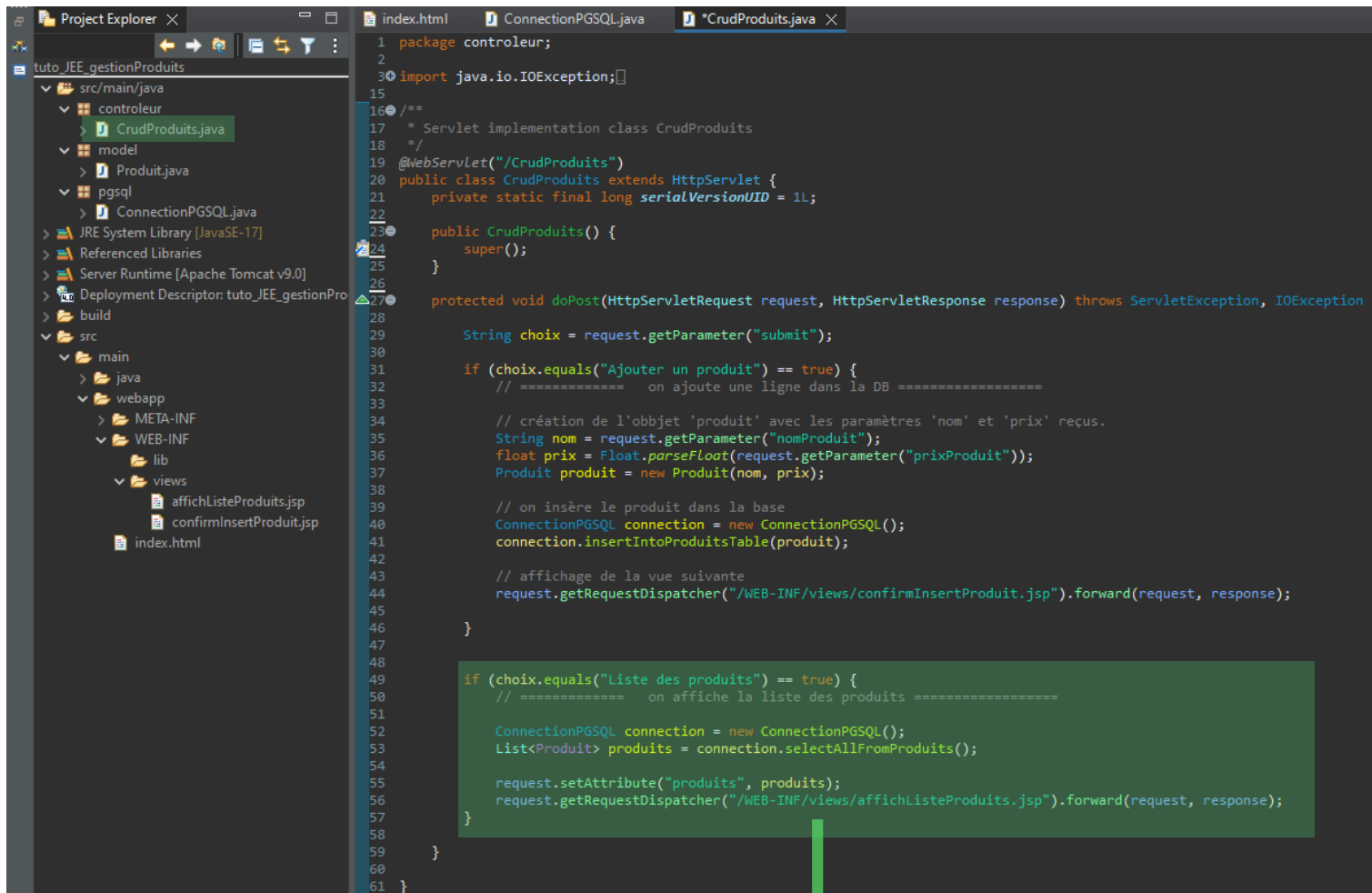


```
1 package controleur;
2
3 import java.io.IOException;
4
5
6
7
8
9
10
11
12
13
14
15 /**
16  * Servlet implementation class CrudProduits
17  */
18 @WebServlet("/CrudProduits")
19 public class CrudProduits extends HttpServlet {
20     private static final long serialVersionUID = 1L;
21
22     public CrudProduits() {
23         super();
24     }
25
26     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
27
28         String choix = request.getParameter("submit");
29
30         if (choix.equals("Ajouter un produit") == true) {
31             // ===== on ajoute une ligne dans la DB =====
32
33             // création de l'objet 'produit' avec les paramètres 'nom' et 'prix' reçus.
34             String nom = request.getParameter("nomProduit");
35             float prix = Float.parseFloat(request.getParameter("prixProduit"));
36             Produit produit = new Produit(nom, prix);
37
38             // on insère le produit dans la base
39             ConnectionPGSQL connection = new ConnectionPGSQL();
40             connection.insertIntoProduitsTable(produit);
41
42             // affichage de la vue suivante
43             request.getRequestDispatcher("/WEB-INF/views/confirmInsertProduit.jsp").forward(request, response);
44
45         }
46
47         if (choix.equals("Liste des produits") == true) {
48             // ===== on affiche la liste des produits =====
49
50             ConnectionPGSQL connection = new ConnectionPGSQL();
51             List<Produit> produits = connection.selectAllFromProduits();
52
53             request.setAttribute("produits", produits);
54             request.getRequestDispatcher("/WEB-INF/views/affichListeProduits.jsp").forward(request, response);
55
56         }
57     }
58 }
59
60
61 }
```

Affichage de la vue de confirmation d'insertion



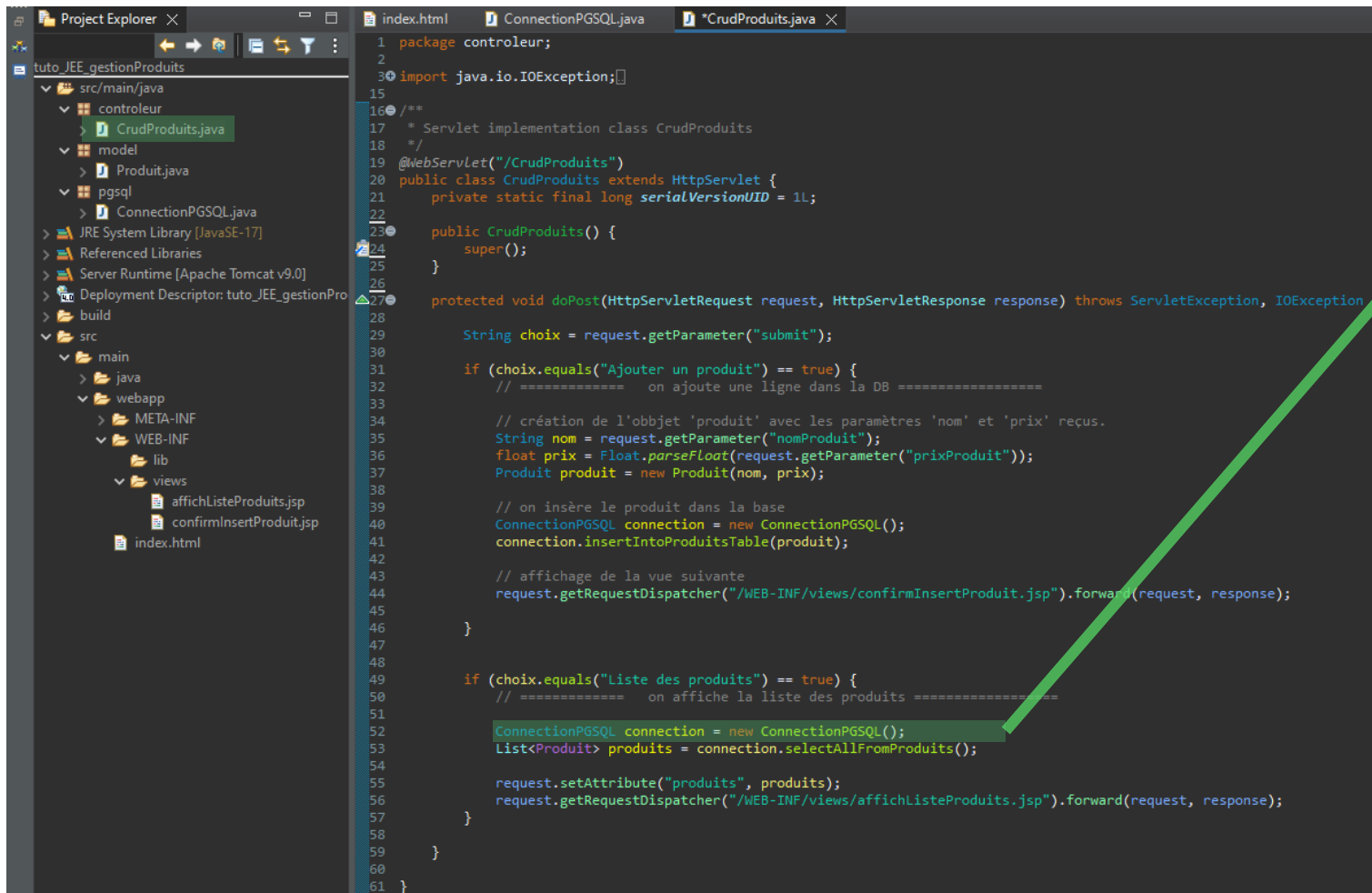
Le Servlet CrudProduits.java



```
1 package controleur;
2
3 import java.io.IOException;
4
5
6
7
8
9
10
11
12
13
14
15
16 /**
17  * Servlet implementation class CrudProduits
18  */
19 @WebServlet("/CrudProduits")
20 public class CrudProduits extends HttpServlet {
21     private static final long serialVersionUID = 1L;
22
23     public CrudProduits() {
24         super();
25     }
26
27     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
28
29         String choix = request.getParameter("submit");
30
31         if (choix.equals("Ajouter un produit") == true) {
32             // ===== on ajoute une ligne dans la DB =====
33
34             // création de l'objet 'produit' avec les paramètres 'nom' et 'prix' reçus.
35             String nom = request.getParameter("nomProduit");
36             float prix = Float.parseFloat(request.getParameter("prixProduit"));
37             Produit produit = new Produit(nom, prix);
38
39             // on insère le produit dans la base
40             ConnectionPGSQL connection = new ConnectionPGSQL();
41             connection.insertIntoProduitsTable(produit);
42
43             // affichage de la vue suivante
44             request.getRequestDispatcher("/WEB-INF/views/confirmInsertProduit.jsp").forward(request, response);
45
46         }
47
48
49         if (choix.equals("Liste des produits") == true) {
50             // ===== on affiche la liste des produits =====
51
52             ConnectionPGSQL connection = new ConnectionPGSQL();
53             List<Produit> produits = connection.selectAllFromProduits();
54
55             request.setAttribute("produits", produits);
56             request.getRequestDispatcher("/WEB-INF/views/affichListeProduits.jsp").forward(request, response);
57         }
58     }
59 }
60
61 }
```

On se place maintenant dans le cas où on veut afficher la liste des produits

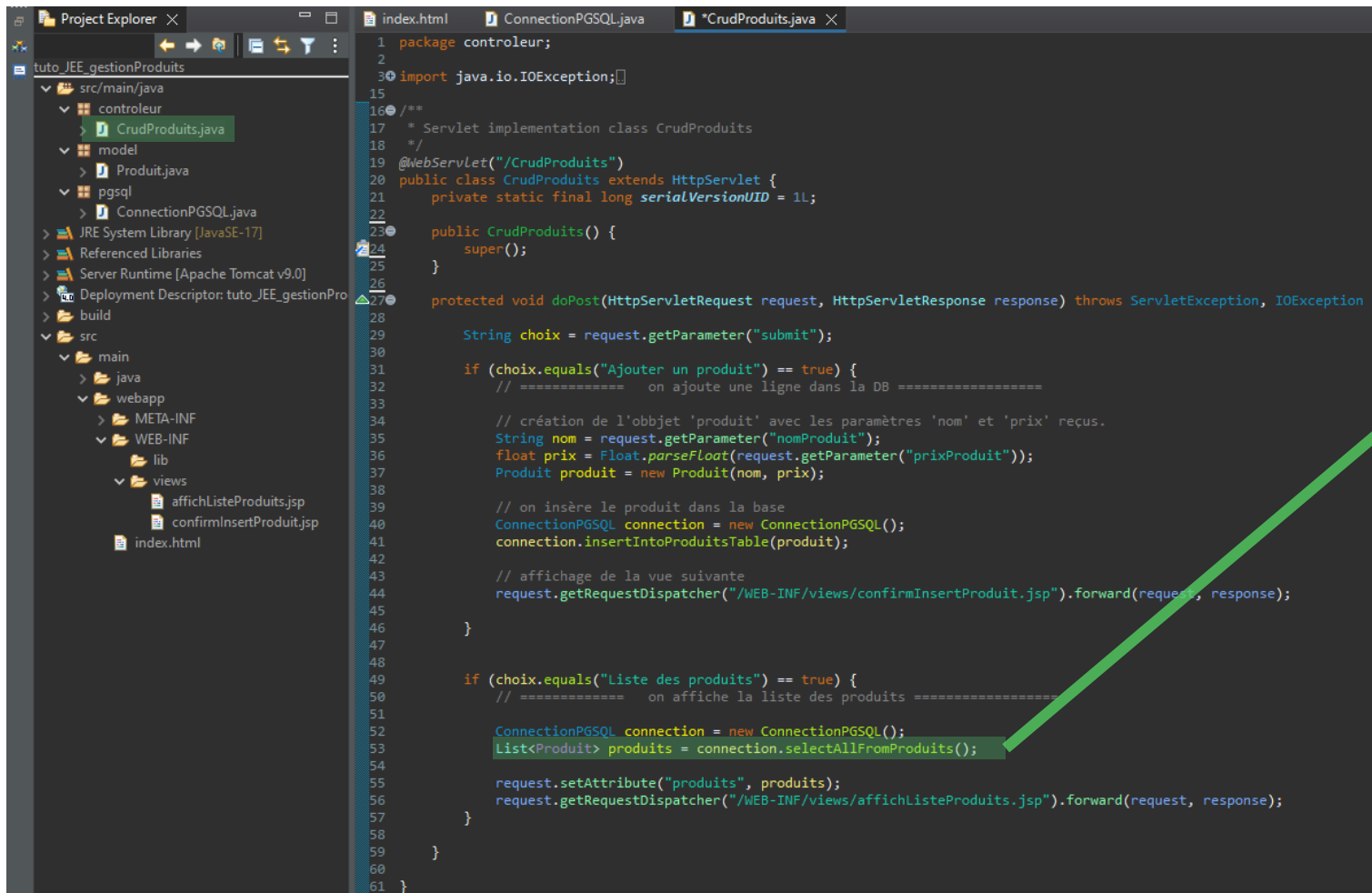
Le Servlet CrudProduits.java



```
1 package controleur;
2
3 import java.io.IOException;
4
5
6
7
8
9
10
11
12
13
14
15
16 /**
17  * Servlet implementation class CrudProduits
18  */
19 @WebServlet("/CrudProduits")
20 public class CrudProduits extends HttpServlet {
21     private static final long serialVersionUID = 1L;
22
23     public CrudProduits() {
24         super();
25     }
26
27     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
28
29         String choix = request.getParameter("submit");
30
31         if (choix.equals("Ajouter un produit") == true) {
32             // ===== on ajoute une ligne dans la DB =====
33
34             // création de l'objet 'produit' avec les paramètres 'nom' et 'prix' reçus.
35             String nom = request.getParameter("nomProduit");
36             float prix = Float.parseFloat(request.getParameter("prixProduit"));
37             Produit produit = new Produit(nom, prix);
38
39             // on insère le produit dans la base
40             ConnectionPGSQL connection = new ConnectionPGSQL();
41             connection.insertIntoProduitsTable(produit);
42
43             // affichage de la vue suivante
44             request.getRequestDispatcher("/WEB-INF/views/confirmInsertProduit.jsp").forward(request, response);
45         }
46     }
47
48     if (choix.equals("Liste des produits") == true) {
49         // ===== on affiche la liste des produits =====
50
51         ConnectionPGSQL connection = new ConnectionPGSQL();
52         List<Produit> produits = connection.selectAllFromProduits();
53
54         request.setAttribute("produits", produits);
55         request.getRequestDispatcher("/WEB-INF/views/affichListeProduits.jsp").forward(request, response);
56     }
57 }
58
59
60
61 }
```

Création d'une connexion

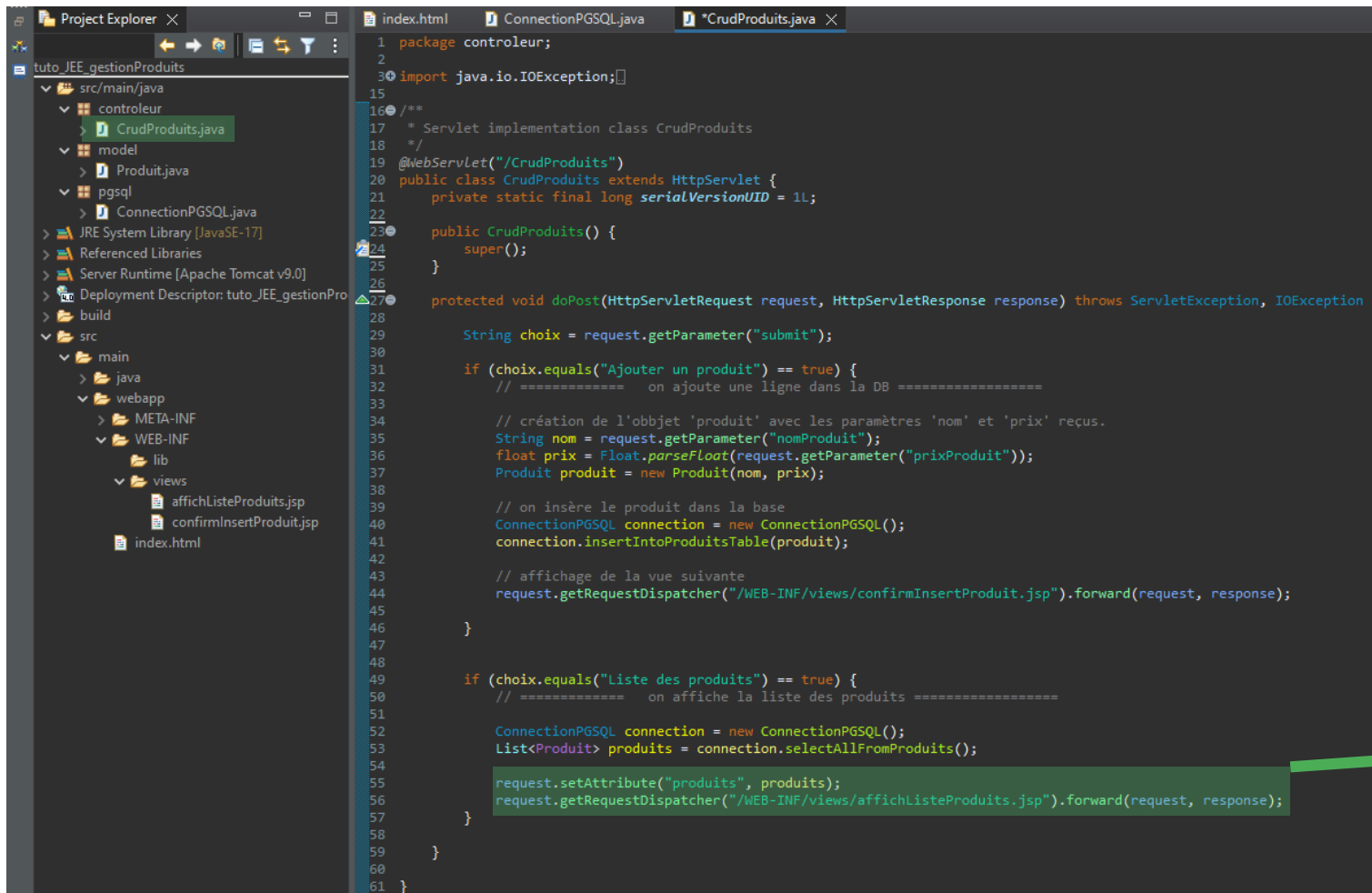
Le Servlet CrudProduits.java



```
1 package controleur;
2
3 import java.io.IOException;
4
5
6
7
8
9
10
11
12
13
14
15
16 /**
17  * Servlet implementation class CrudProduits
18  */
19 @WebServlet("/CrudProduits")
20 public class CrudProduits extends HttpServlet {
21     private static final long serialVersionUID = 1L;
22
23     public CrudProduits() {
24         super();
25     }
26
27     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
28
29         String choix = request.getParameter("submit");
30
31         if (choix.equals("Ajouter un produit") == true) {
32             // ===== on ajoute une ligne dans la DB =====
33
34             // création de l'objet 'produit' avec les paramètres 'nom' et 'prix' reçus.
35             String nom = request.getParameter("nomProduit");
36             float prix = Float.parseFloat(request.getParameter("prixProduit"));
37             Produit produit = new Produit(nom, prix);
38
39             // on insère le produit dans la base
40             ConnectionPGSQL connection = new ConnectionPGSQL();
41             connection.insertIntoProduitsTable(produit);
42
43             // affichage de la vue suivante
44             request.getRequestDispatcher("/WEB-INF/views/confirmInsertProduit.jsp").forward(request, response);
45         }
46
47
48         if (choix.equals("Liste des produits") == true) {
49             // ===== on affiche la liste des produits =====
50
51             ConnectionPGSQL connection = new ConnectionPGSQL();
52             List<Produit> produits = connection.selectAllFromProduits();
53
54             request.setAttribute("produits", produits);
55             request.getRequestDispatcher("/WEB-INF/views/affichListeProduits.jsp").forward(request, response);
56         }
57     }
58 }
59
60
61 }
```

Récupération des produits de la table dans une liste nommée 'produits'

Le Servlet CrudProduits.java



```
1 package controleur;
2
3 import java.io.IOException;
4
5
6
7
8
9
10
11
12
13
14
15
16 /**
17  * Servlet implementation class CrudProduits
18  */
19 @WebServlet("/CrudProduits")
20 public class CrudProduits extends HttpServlet {
21     private static final long serialVersionUID = 1L;
22
23     public CrudProduits() {
24         super();
25     }
26
27     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
28
29         String choix = request.getParameter("submit");
30
31         if (choix.equals("Ajouter un produit") == true) {
32             // ===== on ajoute une ligne dans la DB =====
33
34             // création de l'objet 'produit' avec les paramètres 'nom' et 'prix' reçus.
35             String nom = request.getParameter("nomProduit");
36             float prix = Float.parseFloat(request.getParameter("prixProduit"));
37             Produit produit = new Produit(nom, prix);
38
39             // on insère le produit dans la base
40             ConnectionPGSQL connection = new ConnectionPGSQL();
41             connection.insertIntoProduitsTable(produit);
42
43             // affichage de la vue suivante
44             request.getRequestDispatcher("/WEB-INF/views/confirmInsertProduit.jsp").forward(request, response);
45         }
46     }
47
48     if (choix.equals("Liste des produits") == true) {
49         // ===== on affiche la liste des produits =====
50
51         ConnectionPGSQL connection = new ConnectionPGSQL();
52         List<Produit> produits = connection.selectAllFromProduits();
53
54         request.setAttribute("produits", produits);
55         request.getRequestDispatcher("/WEB-INF/views/affichListeProduits.jsp").forward(request, response);
56     }
57 }
58
59
60
61 }
```

Envoi de la liste vers la JSP d'affichage

La classe ConnectionPGSQL

La classe ConnectionPGSQL

Elle se compose de 3 méthodes utilisées dans les Servlets.

La classe ConnectionPGSQL

Elle se compose de 3 méthodes utilisées dans les Servlets.

```
1 package pgsql;
2
3 import java.sql.*;
4
5
6 public class ConnectionPGSQL {
7
8     private static final String DB_URL = "jdbc:postgresql://localhost:5432/gestion_produits";
9     private static final String USER = "postgres";
10    private static final String PASS = "mdp";
11
12    public Connection getConnection() {
13
14
15    public void insertIntoProduitsTable(Produit produit) {
16
17
18    public List<Produit> selectAllFromProduits() {
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37 }
```

La classe ConnectionPGSQL

Elle se compose de 3 méthodes utilisées dans les Servlets.

```
1 package pgsql;
2
3 import java.sql.*;
4
5 public class ConnectionPGSQL {
6
7     private static final String DB_URL = "jdbc:postgresql://localhost:5432/gestion_produits";
8     private static final String USER = "postgres";
9     private static final String PASS = "mdp";
10
11     public Connection getConnection() {
12
13     }
14
15     public void insertIntoProduitsTable(Produit produit) {
16
17     }
18
19     public List<Produit> selectAllFromProduits() {
20
21     }
22 }
```



Vos infos de connexion

La classe ConnectionPGSQL

Elle se compose de 3 méthodes utilisées dans les Servlets.

```
1 package pgsql;
2
3 import java.sql.*;
4
5 public class ConnectionPGSQL {
6
7     private static final String DB_URL = "jdbc:postgresql://localhost:5432/gestion_produits";
8     private static final String USER = "postgres";
9     private static final String PASS = "mdp";
10
11     public Connection getConnection() {
12
13     }
14
15     public void insertIntoProduitsTable(Produit produit) {
16
17     }
18
19     public List<Produit> selectAllFromProduits() {
20
21     }
22
23 }
```



Crée et renvoie une connexion

La classe ConnectionPGSQL

Elle se compose de 3 méthodes utilisées dans les Servlets.

```
1 package pgsql;
2
3 import java.sql.*;
4
5 public class ConnectionPGSQL {
6
7     private static final String DB_URL = "jdbc:postgresql://localhost:5432/gestion_produits";
8     private static final String USER = "postgres";
9     private static final String PASS = "mdp";
10
11     public Connection getConnection() {
12
13     }
14
15     public void insertIntoProduitsTable(Produit produit) {
16
17     }
18
19     public List<Produit> selectAllFromProduits() {
20
21     }
22
23 }
```

Insertion d'un produit dans la BDD

La classe ConnectionPGSQL

Elle se compose de 3 méthodes utilisées dans les Servlets.

```
1 package pgsql;
2
3 import java.sql.*;
4
5 public class ConnectionPGSQL {
6
7     private static final String DB_URL = "jdbc:postgresql://localhost:5432/gestion_produits";
8     private static final String USER = "postgres";
9     private static final String PASS = "mdp";
10
11     public Connection getConnection() {}
12
13     public void insertIntoProduitsTable(Produit produit) {}
14
15     public List<Produit> selectAllFromProduits() {}
16
17 }
```

Effectue un SELECT * from produits;
Et renvoie une liste de produits

La classe ConnectionPGSQL

La classe ConnectionPGSQL

La méthode getConnection()

La classe ConnectionPGSQL

La méthode getConnection()

```
1 package pgsq1;
2
3+ import java.sql.*;
4
5
6
7
8
9 public class ConnectionPGSQL {
10
11     private static final String DB_URL = "jdbc:postgresql://localhost:5432/gestion_produits";
12     private static final String USER = "postgres";
13     private static final String PASS = "mdp";
14
15- public Connection getConnection() {
16     Connection conn = null;
17     try {
18         conn = DriverManager.getConnection(DB_URL, USER, PASS);
19     } catch (SQLException e) {
20         e.printStackTrace();
21     }
22     return conn;
23 }
24
25+ public void insertIntoProduitsTable(Produit produit) {}
26
27
28
29
30
31+ public List<Produit> selectAllFromProduits() {}
32
33
34
35
36
37 }
```

La classe ConnectionPGSQL

La méthode getConnection()

```
1 package pgsq1;
2
3 import java.sql.*;
4
5
6
7
8
9 public class ConnectionPGSQL {
10
11     private static final String DB_URL = "jdbc:postgresql://localhost:5432/gestion_produits";
12     private static final String USER = "postgres";
13     private static final String PASS = "mdp";
14
15     public Connection getConnection() {
16         Connection conn = null;
17         try {
18             conn = DriverManager.getConnection(DB_URL, USER, PASS);
19         } catch (SQLException e) {
20             e.printStackTrace();
21         }
22         return conn;
23     }
24
25     public void insertIntoProduitsTable(Produit produit) {}
26
27
28
29
30
31     public List<Produit> selectAllFromProduits() {}
32
33
34
35
36
37 }
```

On essaye de se connecter à la BDD selon les infos explicitées juste au dessus.

La classe ConnectionPGSQL

La classe ConnectionPGSQL

La méthode InsertIntoProduits(Produit produit)

La classe ConnectionPGSQL

La méthode InsertIntoProduits(Produit produit)

```
package pgsql;

import java.sql.*;

public class ConnectionPGSQL {

    private static final String DB_URL = "jdbc:postgresql://localhost:5432/gestion_produits";
    private static final String USER = "postgres";
    private static final String PASS = "mdp";

    public Connection getConnection() {}

    public void insertIntoProduitsTable(Produit produit) {
        Connection conn = getConnection();
        Statement stmt = null;
        try {
            stmt = conn.createStatement();
            String nom = produit.getNom();
            String prix = String.valueOf(produit.getPrix());
            String sql = "INSERT INTO produits (nom, prix) " + "VALUES ('" + nom + "', " + prix + ")";
            stmt.executeUpdate(sql);
            System.out.println("Insertion into produits table successful.");
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            try {
                if (stmt != null) {
                    stmt.close();
                }
                if (conn != null) {
                    conn.close();
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }

    public List<Produit> selectAllFromProduits() {}

}
```


La classe ConnectionPGSQL

La méthode InsertIntoProduits(Produit produit)

```
package pgsq1;

import java.sql.*;

public class ConnectionPGSQL {

    private static final String DB_URL = "jdbc:postgresql://localhost:5432/gestion_produits";
    private static final String USER = "postgres";
    private static final String PASS = "mdp";

    public Connection getConnection() {}

    public void insertIntoProduitsTable(Produit produit) {
        Connection conn = getConnection();
        Statement stmt = null;
        try {
            stmt = conn.createStatement();
            String nom = produit.getNom();
            String prix = String.valueOf(produit.getPrix());
            String sql = "INSERT INTO produits (nom, prix) " + "VALUES ('" + nom + "', " + prix + ")";
            stmt.executeUpdate(sql);
            System.out.println("Insertion into produits table successful.");
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            try {
                if (stmt != null) {
                    stmt.close();
                }
                if (conn != null) {
                    conn.close();
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }

    public List<Produit> selectAllFromProduits() {}

}
```

Création d'une connexion 'conn'

La classe ConnectionPGSQL

La méthode InsertIntoProduits(Produit produit)

```
package pgsql;

import java.sql.*;

public class ConnectionPGSQL {

    private static final String DB_URL = "jdbc:postgresql://localhost:5432/gestion_produits";
    private static final String USER = "postgres";
    private static final String PASS = "mdp";

    public Connection getConnection() {}

    public void insertIntoProduitsTable(Produit produit) {
        Connection conn = getConnection();
        Statement stmt = null;
        try {
            stmt = conn.createStatement();
            String nom = produit.getNom();
            String prix = String.valueOf(produit.getPrix());
            String sql = "INSERT INTO produits (nom, prix) " + "VALUES ('" + nom + "', " + prix + ")";
            stmt.executeUpdate(sql);
            System.out.println("Insertion into produits table successful.");
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            try {
                if (stmt != null) {
                    stmt.close();
                }
                if (conn != null) {
                    conn.close();
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }

    public List<Produit> selectAllFromProduits() {}

}
```

On récupère les infos du produit pour former une requête SQL sous format String

La classe ConnectionPGSQL

La méthode InsertIntoProduits(Produit produit)

```
package pgsq1;

import java.sql.*;

public class ConnectionPGSQL {

    private static final String DB_URL = "jdbc:postgresql://localhost:5432/gestion_produits";
    private static final String USER = "postgres";
    private static final String PASS = "mdp";

    public Connection getConnection() {}

    public void insertIntoProduitsTable(Produit produit) {
        Connection conn = getConnection();
        Statement stmt = null;
        try {
            stmt = conn.createStatement();
            String nom = produit.getNom();
            String prix = String.valueOf(produit.getPrix());
            String sql = "INSERT INTO produits (nom, prix) " + "VALUES ('" + nom + "', " + prix + ")";
            stmt.executeUpdate(sql);
            System.out.println("Insertion into produits table successful.");
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            try {
                if (stmt != null) {
                    stmt.close();
                }
                if (conn != null) {
                    conn.close();
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }

    public List<Produit> selectAllFromProduits() {}

}
```

On exécute la requête SQL

La classe ConnectionPGSQL

La méthode InsertIntoProduits(Produit produit)

```
package pgsql;

import java.sql.*;

public class ConnectionPGSQL {

    private static final String DB_URL = "jdbc:postgresql://localhost:5432/gestion_produits";
    private static final String USER = "postgres";
    private static final String PASS = "mdp";

    public Connection getConnection() {}

    public void insertIntoProduitsTable(Produit produit) {
        Connection conn = getConnection();
        Statement stmt = null;
        try {
            stmt = conn.createStatement();
            String nom = produit.getNom();
            String prix = String.valueOf(produit.getPrix());
            String sql = "INSERT INTO produits (nom, prix) " + "VALUES ('" + nom + "', " + prix + ")";
            stmt.executeUpdate(sql);
            System.out.println("Insertion into produits table successful.");
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            try {
                if (stmt != null) {
                    stmt.close();
                }
                if (conn != null) {
                    conn.close();
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }

    public List<Produit> selectAllFromProduits() {}

}
```

Gestion des erreurs et clôture de la connexion

La classe ConnectionPGSQL

La classe ConnectionPGSQL

La méthode selectAllFromProduits

La classe ConnectionPGSQL

La méthode selectAllFromProduits

```
package pgsql;

import java.sql.*;

public class ConnectionPGSQL {

    private static final String DB_URL = "jdbc:postgresql://localhost:5432/gestion_produits";
    private static final String USER = "postgres";
    private static final String PASS = "mdp";

    public Connection getConnection() {}

    public void insertIntoProduitsTable(Produit produit) {}

    public List<Produit> selectAllFromProduits() {
        Connection conn = getConnection();
        Statement stmt = null;
        ResultSet rs = null;
        List<Produit> produits = new ArrayList<>();
        try {
            stmt = conn.createStatement();
            String sql = "SELECT * FROM produits";
            rs = stmt.executeQuery(sql);
            while (rs.next()) {
                int id = rs.getInt("id");
                String nom = rs.getString("nom");
                float prix = rs.getFloat("prix");
                produits.add(new Produit(id, nom, prix));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            try {
                if (rs != null) {
                    rs.close();
                }
                if (stmt != null) {
                    stmt.close();
                }
                if (conn != null) {
                    conn.close();
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        return produits;
    }
}
```

Création d'une connexion
Initialisation des variables

La classe ConnectionPGSQL

La méthode selectAllFromProduits

```
package pgsql;

import java.sql.*;

public class ConnectionPGSQL {

    private static final String DB_URL = "jdbc:postgresql://localhost:5432/gestion_produits";
    private static final String USER = "postgres";
    private static final String PASS = "mdp";

    public Connection getConnection() {}

    public void insertIntoProduitsTable(Produit produit) {}

    public List<Produit> selectAllFromProduits() {
        Connection conn = getConnection();
        Statement stmt = null;
        ResultSet rs = null;
        List<Produit> produits = new ArrayList<>();
        try {
            stmt = conn.createStatement();
            String sql = "SELECT * FROM produits";
            rs = stmt.executeQuery(sql);
            while (rs.next()) {
                int id = rs.getInt("id");
                String nom = rs.getString("nom");
                float prix = rs.getFloat("prix");
                produits.add(new Produit(id, nom, prix));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            try {
                if (rs != null) {
                    rs.close();
                }
                if (stmt != null) {
                    stmt.close();
                }
                if (conn != null) {
                    conn.close();
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        return produits;
    }
}
```

On exécute la requête 'SELECT * FROM produits;'

Les résultats sont stockés dans la variable nommée 'rs'

On peut considérer cette variable comme un tableau contenant une ligne pour chaque produit.

La classe ConnectionPGSQL

La méthode selectAllFromProduits

```
package pgsql;

import java.sql.*;

public class ConnectionPGSQL {

    private static final String DB_URL = "jdbc:postgresql://localhost:5432/gestion_produits";
    private static final String USER = "postgres";
    private static final String PASS = "mdp";

    public Connection getConnection() {}

    public void insertIntoProduitsTable(Produit produit) {}

    public List<Produit> selectAllFromProduits() {
        Connection conn = getConnection();
        Statement stmt = null;
        ResultSet rs = null;
        List<Produit> produits = new ArrayList<>();
        try {
            stmt = conn.createStatement();
            String sql = "SELECT * FROM produits";
            rs = stmt.executeQuery(sql);
            while (rs.next()) {
                int id = rs.getInt("id");
                String nom = rs.getString("nom");
                float prix = rs.getFloat("prix");
                produits.add(new Produit(id, nom, prix));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            try {
                if (rs != null) {
                    rs.close();
                }
                if (stmt != null) {
                    stmt.close();
                }
                if (conn != null) {
                    conn.close();
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        return produits;
    }
}
```

Tant qu'il y a une ligne non traitée (mal formulé désolé)

Alors on crée un produit avec les attributs de la ligne.

Ce produit est alors inséré dans la liste 'produits'

La classe ConnectionPGSQL

La méthode selectAllFromProduits

```
package pgsql;

import java.sql.*;

public class ConnectionPGSQL {

    private static final String DB_URL = "jdbc:postgresql://localhost:5432/gestion_produits";
    private static final String USER = "postgres";
    private static final String PASS = "mdp";

    public Connection getConnection() {

    }

    public void insertIntoProduitsTable(Produit produit) {

    }

    public List<Produit> selectAllFromProduits() {
        Connection conn = getConnection();
        Statement stmt = null;
        ResultSet rs = null;
        List<Produit> produits = new ArrayList<>();
        try {
            stmt = conn.createStatement();
            String sql = "SELECT * FROM produits";
            rs = stmt.executeQuery(sql);
            while (rs.next()) {
                int id = rs.getInt("id");
                String nom = rs.getString("nom");
                float prix = rs.getFloat("prix");
                produits.add(new Produit(id, nom, prix));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            try {
                if (rs != null) {
                    rs.close();
                }
                if (stmt != null) {
                    stmt.close();
                }
                if (conn != null) {
                    conn.close();
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        return produits;
    }
}
```

La méthode renvoie la liste 'produits', cette liste sera traitée par le Servlet ayant appelé cette méthode.

Le servlet l'enverra sur la JSP qui l'affichera

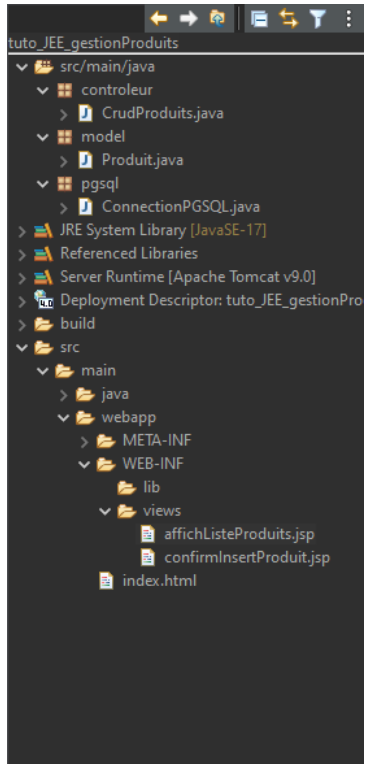
Les JSP

Les JSP

Elles récupèrent les infos envoyés par le Servlet pour les afficher

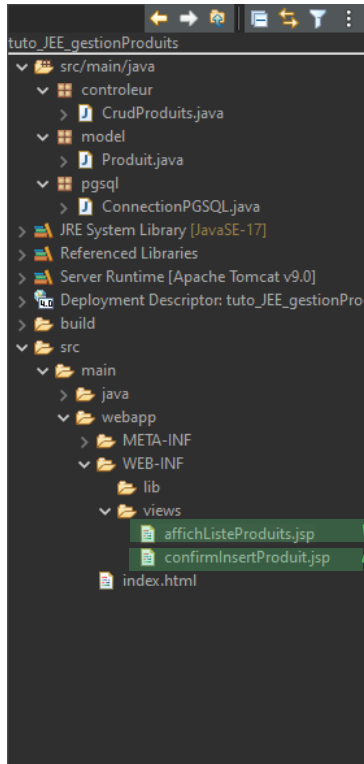
Les JSP

Elles récupèrent les infos envoyés par le Servlet pour les afficher



Les JSP

Elles récupèrent les infos envoyés par le Servlet pour les afficher



affichListeProduits.jsp

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="UTF-8"%>
2 <%@ page import="model.Produit"%>
3 <%@ page import="pgsql.ConnectionPGSQL"%>
4 <%@ page import="java.util.List"%>
5
6 <!DOCTYPE html>
7 <html>
8 <head>
9 <meta charset="ISO-8859-1">
10
11 <title>Liste des produits</title>
12 </head>
13 <body>
14 <h1>Liste des produits</h1>
15 <table border="1">
16 <tr>
17 <th>ID</th>
18 <th>Nom</th>
19 <th>Prix</th>
20 </tr>
21 <%
22     ConnectionPGSQL connectionPGSQL = new ConnectionPGSQL();
23     List<Produit> produits = connectionPGSQL.selectAllFromProduits();
24     for (Produit produit : produits) {
25
26 <tr>
27 <td><%= produit.getId() %></td>
28 <td><%= produit.getNom() %></td>
29 <td><%= produit.getPrix() %></td>
30 </tr>
31 <%
32     }
33 <%
34 </table>
35
36 <br><a href='index.html'>index.html</a>
37 </body>
38 </html>
39
```

confirmInsertProduit.jsp

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="UTF-8"%>
2 <%@ page import="model.Produit"%>
3 <%@ page import="pgsql.ConnectionPGSQL"%>
4 <%@ page import="java.util.List"%>
5
6 <!DOCTYPE html>
7 <html>
8 <head>
9 <meta charset="ISO-8859-1">
10
11 <title>Confirmation insert produit</title>
12 </head>
13 <body>
14 <h1>Produit enregistré dans la base</h1>
15 <br><a href='index.html'>index.html</a>
16 </body>
17 </html>
18
```