Eclipse üzerinde Java tabanlı bir web uygulaması geliştirecek ve PostgreSQL veritabanına bağlanacak bir API oluşturmak için birkaç adımı takip etmeniz gerekecek. Bu adımlarda JSP sayfaları, servletler, JDBC bağlantısı ve projeyi dışa aktararak bir platforma (Render gibi) yüklemek de yer alacak. Şimdi size adım adım nasıl ilerleyeceğinizi anlatacağım.

**1. Eclipse IDE Kurulumu ve Proje Başlatma**

**1.1 Eclipse İnstallasyonu**

Eclipse IDE'yi indirip kurduğunuzdan emin olun. Eğer kurulu değilse, Eclipse resmi sitesinden uygun sürümü indirebilirsiniz.

**1.2 Yeni Web Projesi Oluşturma**

Eclipse'te yeni bir **Dynamic Web Project** (Dinamik Web Projesi) oluşturmanız gerekiyor.

1. **Eclipse'i Başlatın** ve "File" → "New" → "Dynamic Web Project" seçeneğini tıklayın.
2. Projeye bir isim verin, örneğin PostgreSQLWebApp.
3. **Runtime** olarak Apache Tomcat seçin (Tomcat'in kurulu olduğundan emin olun. Tomcat'i [buradan indirebilirsiniz](http://tomcat.apache.org/download-90.cgi)).
4. **Dynamic Web Module Version**: 3.1 seçebilirsiniz (JSP ve Servlet kullanabilmek için).
5. "Finish" butonuna tıklayın.

Bu işlem, web projelerinizi yönetebileceğiniz bir klasör yapısı oluşturur.

**2. PostgreSQL JDBC Bağlantısı**

Projenize PostgreSQL ile bağlantı kurabilmek için JDBC driver'ını eklemeniz gerekiyor. Bu bağlantıyı sağlamak için aşağıdaki adımları takip edin:

**2.1 PostgreSQL JDBC Driver'ı Ekleyin**

Eclipse projelerinde JDBC bağlantısını gerçekleştirebilmek için PostgreSQL JDBC sürücüsünü projenize eklemelisiniz. Eğer Maven kullanıyorsanız, bunu pom.xml dosyasına ekleyebilirsiniz.

**Maven Kullanıyorsanız:**

Projenizde Maven kullanarak PostgreSQL JDBC driver'ını eklemek için **pom.xml** dosyasına şu satırı ekleyin:

xml

Copy code

<dependency>

<groupId>org.postgresql</groupId>

<artifactId>postgresql</artifactId>

<version>42.5.0</version>

</dependency>

**Maven Kullanmıyorsanız:**

Eclipse'te, PostgreSQL JDBC jar dosyasını manuel olarak projenize dahil edebilirsiniz:

1. PostgreSQL JDBC sürücüsünü [PostgreSQL JDBC sürücü sayfasından](https://jdbc.postgresql.org/download.html) indirin.
2. **WebContent/WEB-INF/lib** klasörüne bu jar dosyasını ekleyin.

**2.2 JDBC Bağlantısını Oluşturun**

DatabaseConnection.java adlı bir sınıf oluşturun ve PostgreSQL veritabanı ile bağlantıyı sağlayacak kodu buraya yazın:

java

Copy code

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DatabaseConnection {

private static final String URL = "jdbc:postgresql://dpg-ct4cdttds78s73bhu7s0-a.frankfurt-postgres.render.com:5432/odev1";

private static final String USER = "sa";

private static final String PASSWORD = "qNgqTkPT8lnyPhO92vofBAiaQcHZpAsk";

public static Connection connect() throws SQLException {

try {

// PostgreSQL veritabanına bağlan

Connection connection = DriverManager.getConnection(URL, USER, PASSWORD);

return connection;

} catch (SQLException e) {

e.printStackTrace();

throw new SQLException("Veritabanına bağlanırken bir hata oluştu: " + e.getMessage());

}

}

}

**3. JSP Sayfası ve Servlet İle Veri Çekme**

**3.1 JSP Sayfası Oluşturma**

WebContent klasörü altında Users.jsp adında bir dosya oluşturun. Bu dosya, veritabanından çekilen kullanıcıları gösterecek:

jsp

Copy code

<%@ page import="java.util.\*, java.sql.\*" %>

<%@ page import="com.example.UserDAO" %>

<%@ page import="com.example.User" %>

<%

UserDAO userDAO = new UserDAO();

List<User> users = userDAO.getAllUsers();

%>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Users List</title>

</head>

<body>

<h2>Users List</h2>

<table border="1">

<thead>

<tr>

<th>ID</th>

<th>Email</th>

<th>Name</th>

<th>Gender</th>

<th>City</th>

</tr>

</thead>

<tbody>

<% for (User user : users) { %>

<tr>

<td><%= user.getId() %></td>

<td><%= user.getEmail() %></td>

<td><%= user.getName() %></td>

<td><%= user.getGender() %></td>

<td><%= user.getCity() %></td>

</tr>

<% } %>

</tbody>

</table>

</body>

</html>

**3.2 Servlet Oluşturma**

Veritabanından veri çekmek ve JSP sayfasına yönlendirmek için bir servlet oluşturun:

java

Copy code

@WebServlet("/getUsers")

public class GetUsersServlet extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

UserDAO userDAO = new UserDAO();

try {

List<User> users = userDAO.getAllUsers();

request.setAttribute("users", users);

request.getRequestDispatcher("/Users.jsp").forward(request, response);

} catch (SQLException e) {

e.printStackTrace();

response.getWriter().write("Veritabanına bağlanırken bir hata oluştu: " + e.getMessage());

}

}

}

**4. Proje Yapısı**

Projede oluşturduğunuz dosyaların genel yapısı şu şekilde olacak:

bash

Copy code

PostgreSQLWebApp/

├── build/

├── lib/ # Buraya PostgreSQL JDBC jar dosyasını eklediniz.

├── WebContent/

│ ├── index.jsp

│ ├── Users.jsp # Verileri görüntüleyen JSP

│ └── WEB-INF/

│ └── web.xml # Web uygulaması yapılandırma dosyası

├── src/

│ └── com/example/

│ ├── DatabaseConnection.java # Veritabanı bağlantı sınıfı

│ ├── UserDAO.java # Kullanıcı verilerini çeken DAO sınıfı

│ ├── User.java # Kullanıcı sınıfı (Java Bean)

│ ├── GetUsersServlet.java # Servlet sınıfı

**5. Render'a Yükleme**

Eclipse üzerinde geliştirdiğiniz projeyi Render'a yüklemek için şu adımları takip edebilirsiniz:

**5.1 Render'a Hesap Oluşturun**

1. [Render](https://render.com/) sitesine gidin ve bir hesap oluşturun.

**5.2 Yeni Uygulama Oluşturun**

1. Render dashboard'unda "Create a New Web Service" butonuna tıklayın.
2. **GitHub** ile entegrasyon kullanarak projeyi yükleyebilirsiniz. GitHub üzerinde bir repo oluşturun ve proje dosyalarınızı buraya push edin.
3. **GitHub Repo** bağlantısını Render'da belirterek yükleme işlemi başlatabilirsiniz.
4. Render, projeyi otomatik olarak derleyecek ve çalıştıracaktır.

**5.3 JDBC Bağlantısı İçin Dikkat Edilmesi Gerekenler**

Render üzerinde veritabanı bağlantısını yapabilmek için, PostgreSQL veritabanınızın dışarıdan erişilebilir olması gerekir. Eğer Render üzerinde de veritabanı kullanacaksanız, Render’ın sunduğu PostgreSQL hizmetini kullanabilirsiniz.

Veritabanı bağlantısı için environment variables (çevresel değişkenler) kullanarak URL, USER, ve PASSWORD bilgilerini daha güvenli şekilde yapılandırabilirsiniz.

**5.4 Uygulamanızın Çalışıp Çalışmadığını Kontrol Edin**

Projenizi Render üzerinde çalıştırdıktan sonra, uygulamanızın URL'sine giderek doğru şekilde çalışıp çalışmadığını kontrol edebilirsiniz.

**6. Sonuç**

Bu adımları takip ederek, Eclipse üzerinde bir web uygulaması geliştirebilir, PostgreSQL veritabanı ile bağlantı kurabilir ve JSP ile verileri görüntüleyebilirsiniz. Son olarak, Render’a yükleme işlemi ile uygulamanızı çevrimiçi hale getirebilirsiniz.