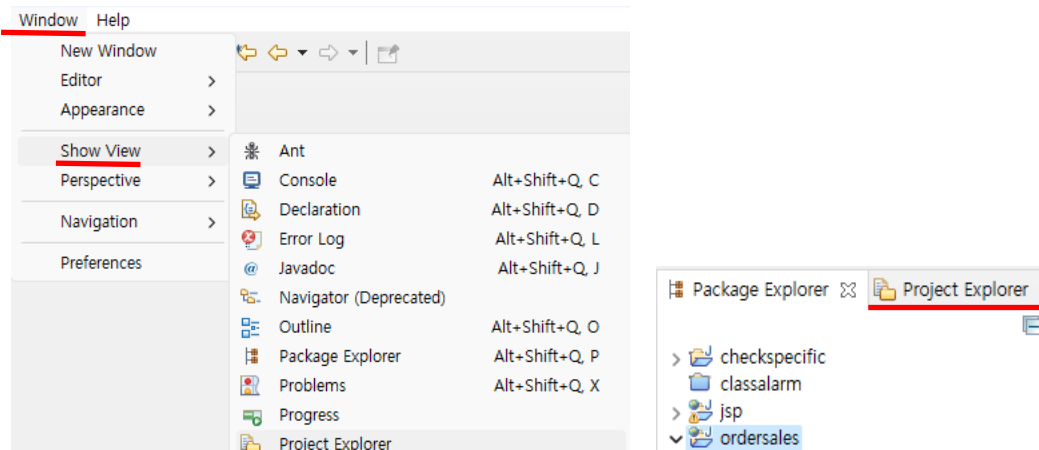
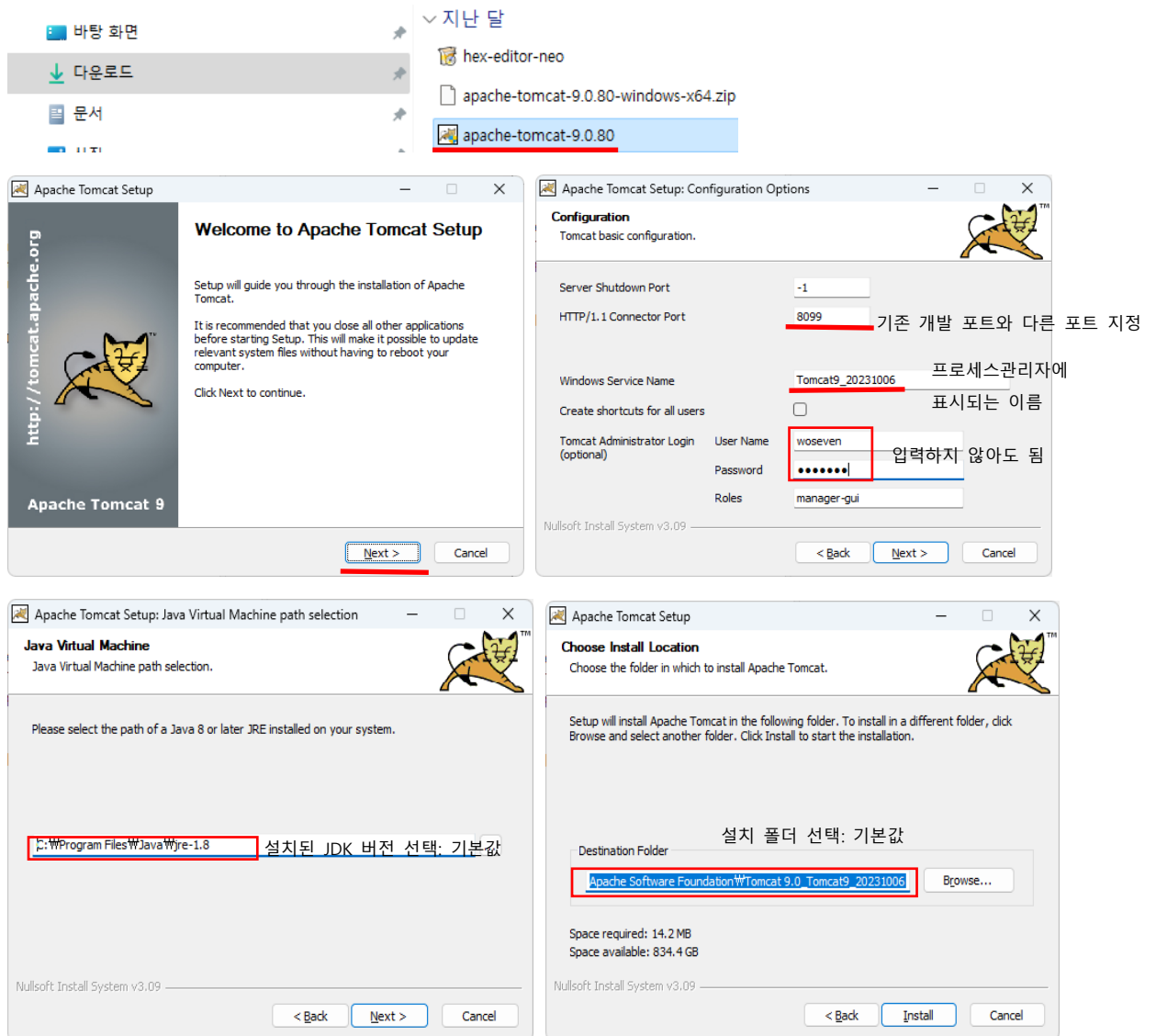


1. 프로젝트 탐색기 추가

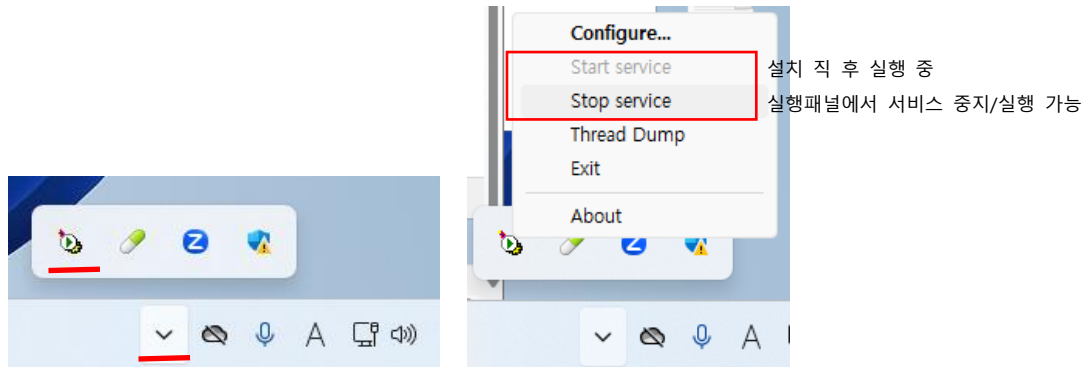


2. 톰캣 인스톨 파일 설치(압축 파일X)

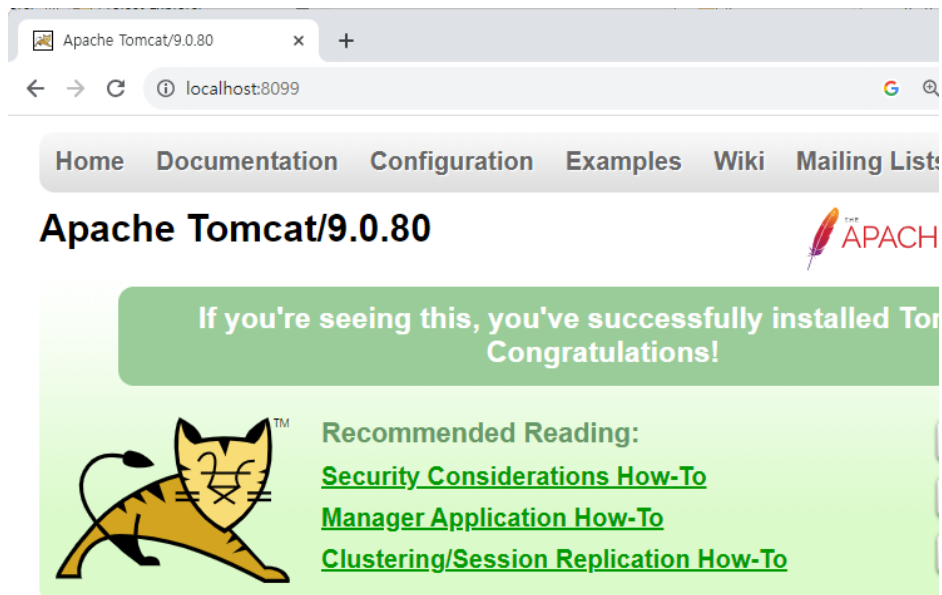


3. 설치된 톰캣 실행 확인: 설치 완료 되면 자동 실행

가. 실행 설정 메뉴: 오른쪽 하단

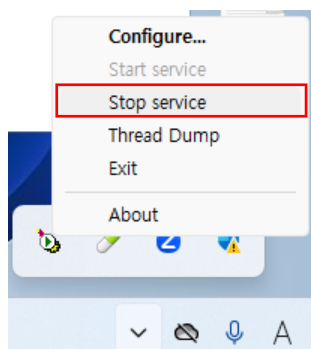


나. 톰캣 설치 및 정상 실행 여부 확인: <http://localhost:8099/>

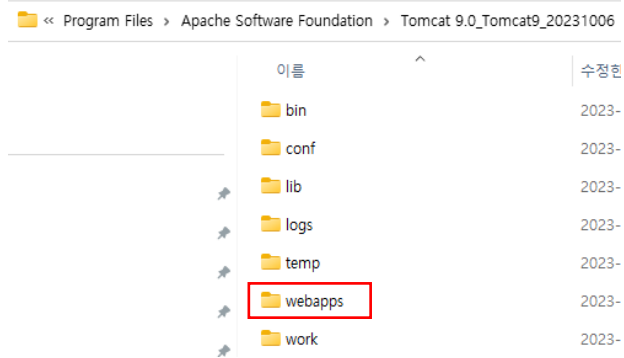


4. Project Explorer 선택 후 프로젝트 배포

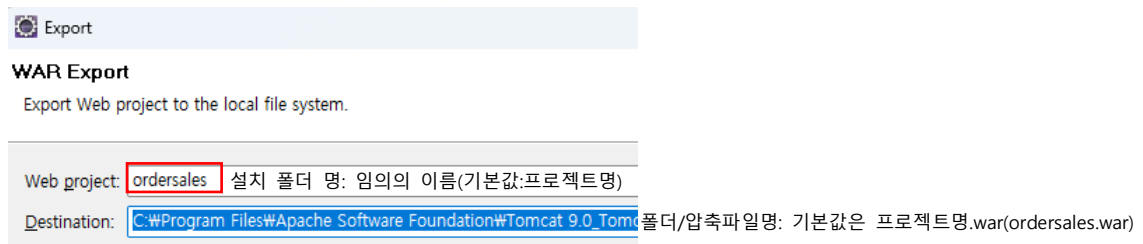
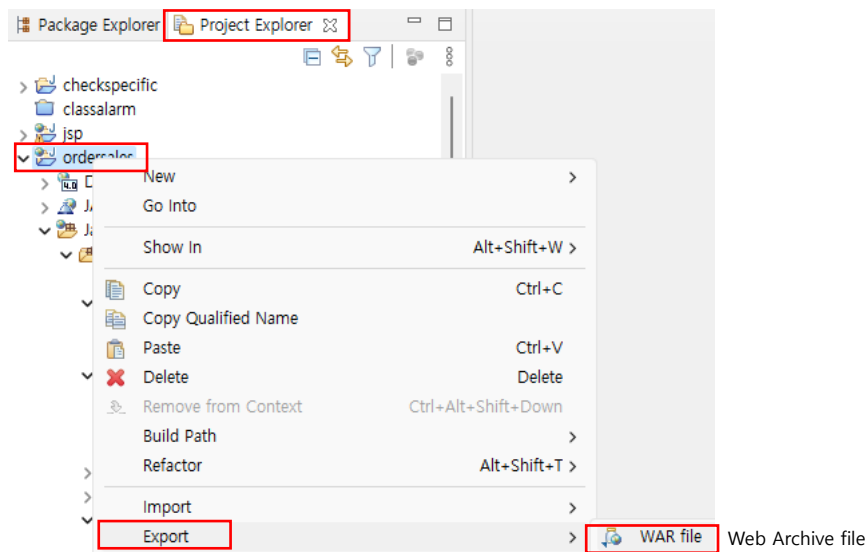
가. 톰캣 종료



나. 배포 폴더 확인: 톰캣 설치 폴더/webapps

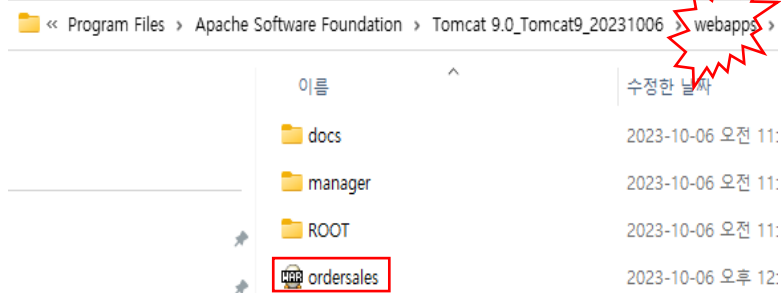


다. 배포(Deployment)



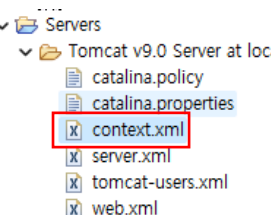
라. 배포 파일 확인

주의사항: 배포위치는 반드시 webapps 폴더 아래



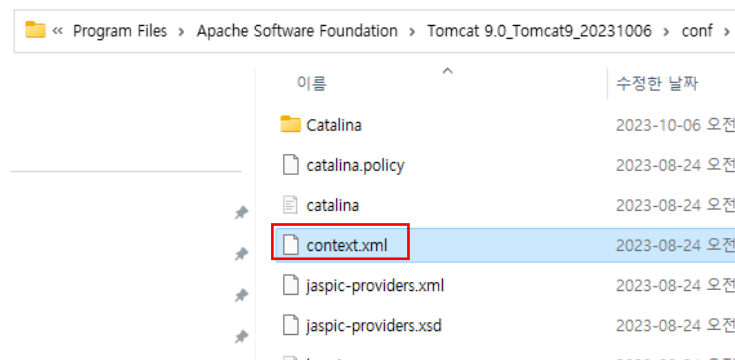
5. DBCP를 사용할 경우 추가사항

가. DBCP(Database Connection Pool) 설정 파일 및 설정 내용 복사



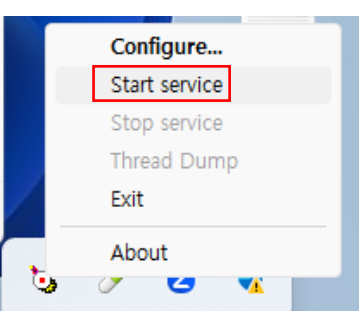
```
<Resource
    auth="Container"
    driverClassName="oracle.jdbc.OracleDriver"
    url="jdbc:oracle:thin:@localhost:1521:xe"
    username="kht123"
    password="kht123"
    name="jdbc/ordersales"
    type="javax.sql.DataSource"
    maxActive="50"
    maxWait="1000" />
```

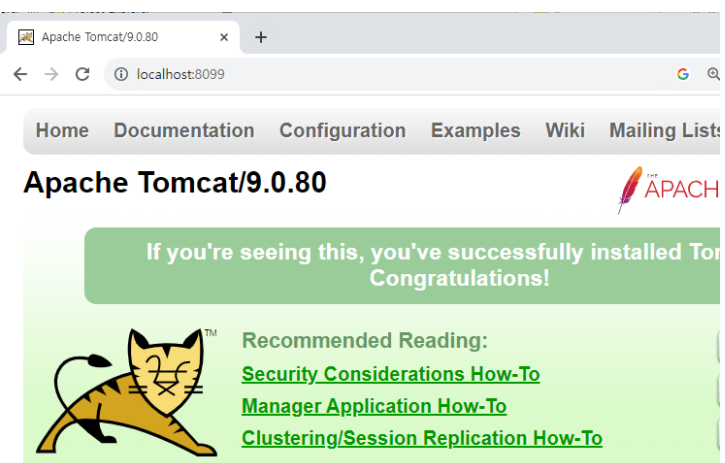
나. 외부 톰캣 설정파일로 이동해 붙여 넣기



```
<Resource
    auth="Container"
    driverClassName="oracle.jdbc.OracleDriver"
    url="jdbc:oracle:thin:@localhost:1521:xe"
    username="kht123"
    password="kht123"
    name="jdbc/ordersales"
    type="javax.sql.DataSource"
    maxActive="50"
    maxWait="1000" />
```

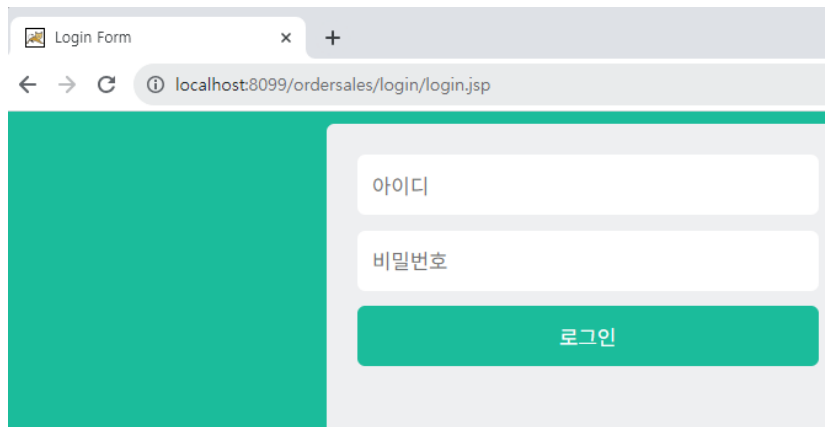
6. 톰캣 서비스 시작 및 톰캣 기본 페이지 다시 확인



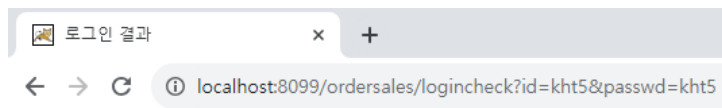


7. 배포 서비스 실행 테스트

가. View 테스트



나. Service 및 Dao 테스트



아이디	암호	주소
kht0	kht0	서울시
kht22	kht22	광주시
kht26	kht26	뉴욕시
kht1	kht1	서울시
kht2	kht2	서구시
kht3	kht3	서디시
kht4	kht4	서동시
kht5	kht5	서운시
kht6	kht6	서산시

8. 확인 사항: JSP와 서블릿 코드 내 지정된 페이지 URL을 절대 경로로 설정했을 경우, 새로 설치한 톰캣 WAS서버 설치할 때 설정한 포트로 변경을 직접해주거나, 상대경로로 지정해주어야 함.

9. DBCP용 유틸 파일과 활용 예제

가. 설정 파일: Context.xml

```
<Resource
    Auth="Container"
    DriverClassName="oracle.jdbc.OracleDriver"
    url="jdbc:oracle:thin:@localhost:1521:xe"
    Username="kht123"
    password="kht123"
    name="jdbc/ordersales"
    type="javax.sql.DataSource"
    maxActive="50"
    maxWait="1000" />
```

나. 유틸 파일

Package com.team.ordersales.login.utils;

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
```

```
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.sql.DataSource;
```

```
public class DBManager {
    // 1. D/B 연결
    public static Connection getConnection() {
        Connection conn = null;
        try {
            // 컨텍스트 객체 생성
            Context context = new InitialContext();
            // 데이터베이스 소스 정보 추출
            DataSource dataSource = (DataSource)context.lookup(
                "java:comp/env/jdbc/ordersales");
            // 데이터 소스에 기반한 연결 객체 추출
            conn = dataSource.getConnection();

        } catch (NamingException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }

        return conn;
    }

    // 정적 select문 사용 후 종료
    public static void close(Connection conn,
        Statement stmt, ResultSet rs) {
        try {
            rs.close();
            stmt.close();
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    // 정적 insert/update/delete문 사용 후 종료
```

```

public static void close(Connection conn,
                        Statement stmt) {
    try {
        stmt.close();
        conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

// 동적 select문 사용 후 종료
public static void close(Connection conn,
                        PreparedStatement pstmt, ResultSet rs) {
    try {
        rs.close();
        pstmt.close();
        conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

// 동적 insert/update/delete문 사용 후 종료
public static void close(Connection conn,
                        PreparedStatement pstmt) {
    try {
        pstmt.close();
        conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}

```


다. 활용 예제

```
try {
    Class.forName("oracle.jdbc.OracleDriver");
} catch (ClassNotFoundException e) {
    e.printStackTrace();
}

conn = DBManager.getConnection();

try {
    pstmt = conn.prepareStatement(sql);
    pstmt.setString(1, sParaId);
    pstmt.setString(2, sParaPasswd);
    rs = pstmt.executeQuery();

    while(rs.next()) {
        ++iCnt;
        break;
    }

    DBManager.close(conn, pstmt, rs);
}
```