

Servlet İçinden DataSource Erişimi



Servlet İçinden DataSource Erişimi (@Resource ile)

```
@WebServlet("/currentTime")
public class TimeServlet extends HttpServlet {

    @Resource(name = "jdbc/testDB")
    private DataSource dataSource;

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

        Connection c = dataSource.getConnection();
        CallableStatement cstmt = c.prepareCall("call current_timestamp()");
        cstmt.execute();
        ResultSet rs = cstmt.getResultSet();
        rs.next();

        response.getWriter().write(rs.getTimestamp(1).toString());

        ...
    }
}
```

DataSource nesnesi web server'da Belirtilen isim ile tanımlanmıştır @Resource anotasyonu ile bu isme lookup yapılır ve enjekte edilir

Servlet İçinden DataSource Erişimi (Explicit JNDI Lookup ile)

```
InitialContext context = new InitialContext();
DataSource dataSource =
    (DataSource) context.lookup("java:comp/env/jdbc/testDB");
Connection c = dataSource.getConnection();
CallableStatement cstmt = c.prepareCall("call current_timestamp()");
cstmt.execute();
ResultSet rs = cstmt.getResultSet();
rs.next();
response.getWriter().write(rs.getTimestamp(1).toString());
```

<web-app>

...

<resource-ref>

<res-ref-name>jdbc/testDB</res-ref-name>

<res-type>javax.sql.DataSource</res-type>

<res-auth>Container</res-auth>

<mapped-name>java:/DefaultDS</mapped-name>

</resource-ref>

</web-app>

Server da JNDI'a bind edildiği gerçek ismidir.
Eğer kod içerisindeki isim ile bind edildiği isim aynı ise
web.xml'deki bu tanıma ihtiyaç yoktur

İletişim



www.harezmi.com.tr

www.java-egitimleri.com



info@harezmi.com.tr

info@java-egitimleri.com



[@HarezmiBilisim](https://twitter.com/HarezmiBilisim)

[@JavaEgitimleri](https://twitter.com/JavaEgitimleri)