



# Using Tomcat to Host the SAP SQL OData Producer

## *PRINCIPAL AUTHORS*

Philippe Bertrand      [philippe.bertrand@sap.com](mailto:philippe.bertrand@sap.com)

Adam Hurst              [adam.hurst@sap.com](mailto:adam.hurst@sap.com)

## *REVISION HISTORY*

Version 1.0 - September 2015



# TABLE OF CONTENTS

<i>PRINCIPAL AUTHORS</i> .....	1
<i>REVISION HISTORY</i> .....	1
<b>OVERVIEW</b> .....	3
<b>GOAL</b> .....	3
<b>HOSTING THE ODATA PRODUCER</b> .....	4
Installing the OData Producer Servlet .....	4
<b>REMARKS</b> .....	4
<b>APPENDIX</b> .....	5
web.xml.....	5
odata_config.properties.....	5
odata.osdl.....	5

## OVERVIEW

Although the SAP SQL Anywhere 17 database server can host an OData Server, there are times users may want to deploy their OData services on the same webserver as their main web site or on a separate machine from the database server.

This document provides an introduction to hosting the SAP SQL OData Producer in a third-party HTTP server that supports Java Servlets.

Readers are encouraged to read the *Setting up an Apache Server in Conjunction with the SAP SQL OData Server* paper which discusses how to have an Apache HTTP Server proxy to the HTTP server hosting the OData Producer to create a web application with rich OData content and functionality.

**OData** (Open Data Protocol) enables data services over RESTful HTTP. It allows you to perform operations through URIs (Universal Resource Identifiers) to access and modify information.

**The SAP SQL OData Producer** (herein referred to as the OData Producer) is a Java Servlet that uses the JDBC API to connect to a SQL Anywhere, SAP IQ, or SAP ASE database server. It processes OData requests, interfaces with a database, and provides OData responses.

To successfully complete the procedures in this document, you must acquire and install the following software:

- **Java SE Development Kit (JRE) 7 or 8** The Java 7 runtime environment is required by the Tomcat deployment.
- **Apache Tomcat 8.0.20** Tomcat is an open source software implementation of the Java Servlet technology. The OData Producer servlet can be hosted by any servlet container that supports version 3.1 of the Java Servlet API, such as the Tomcat Servlet Container. The rest of this document assumes that \$TOMCAT\_HOME is the root directory of the Tomcat installation.
- **SQL Anywhere 17** SQL Anywhere contains the necessary components to allow the OData Server to interface with SQL Anywhere database servers. It also provides a sample SQL Anywhere database that is used as the data source for the web application. The rest of this document assumes that the \$SQLANY17/Bin64 directory is in your PATH and that the SQLANYSAMP17 environment variable is set to the root of the SQL Anywhere samples (both SQLANY17 and SQLANYSAMP17 are set by the SQL Anywhere installer).

## GOAL

This document provides a procedure for configuring and deploying an OData Producer in a Tomcat Servlet container. Raw OData requests may be done using a browser, which the Tomcat server receives and passes to the OData Producer. The selected producer satisfies each request with the help of the SQL Anywhere database server.

The following diagram describes the back-end setup.



## HOSTING THE ODATA PRODUCER

### Installing the OData Producer Servlet

The following steps illustrate how to install the OData Producer servlet into the Tomcat Servlet Container, and how to configure the servlet to interface with the sample database provided in the SQL Anywhere installation.

1. Create the following directory structures in the Tomcat installation:

`$TOMCAT_HOME/webapps/odata`  
`$TOMCAT_HOME/webapps/odata/WEB-INF`  
`$TOMCAT_HOME/webapps/odata/WEB-INF/lib`

2. Copy the corresponding files listed in the appendix to:

`$TOMCAT_HOME/webapps/odata/WEB-INF/web.xml`  
`$TOMCAT_HOME/webapps/odata/WEB-INF/odata_config.properties`  
`$TOMCAT_HOME/webapps/odata/WEB-INF/odata.osdl`

**web.xml** This Deployment Descriptor file describes how to deploy the servlet within Tomcat.

**odata\_config.properties** This properties file configures the OData Producer. This file defines a number of OData Producer behaviors and provides the details necessary to connect to the SQL Anywhere database server.

**odata.osdl** This OData Service Definition Language file provides the OData Producer with additional information regarding how to expose the schema of the SQL Anywhere sample database.

3. Copy the following files to the `$TOMCAT_HOME/webapps/odata/WEB-INF/lib` directory:

`$SQLANY17/java/dbodata.jar`  
`$SQLANY17/java/sajdbc4.jar`

**dbodata.jar** This file contains the OData Producer servlet.

**sajdbc4.jar** This file provides the JDBC drive that the OData Producer uses to interface with the SQL Anywhere database server.

4. Start the Tomcat Servlet Container by running `$TOMCAT_HOME/bin/startup.bat`.
5. Verify that the servlet loaded successfully by using a web browser to visit

[http://localhost:8080/odata/\\$metadata](http://localhost:8080/odata/$metadata).

An XML document describing the OData service schema appears.

### REMARKS

As shown in this document, the OData Producer servlet can be hosted by any servlet container that supports version 3.1 of the Java Servlet API, such as the Tomcat Servlet Container and the Jetty HTTP Server (9.2.4). Most servlet containers can be configured to host multiple instances of the OData Producer, with each producer publishing its own OData resources, possibly from different database servers. In this scenario, the servlet container configures each servlet to process request for a given base URI which corresponds to the OData Producer's service root (the ServiceRoot configuration option).

In all production environments, the OData Producer relies on the containing HTTP server for security. Refer to the HTTP server's documentation for instructions on how to configure the HTTP server such that OData request are received only through HTTPS requests.

Configuring and deploying a web application with rich OData content is discussed in the paper *Setting up an Apache Server in Conjunction with the SAP SQL OData Server*.

### APPENDIX

#### web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  version="3.1"
  metadata-complete="true">

  <display-name>SQLAnywhere OData Producer</display-name>
  <description>
    This is an OData producer for the SQLAnywhere demo database.
  </description>

  <servlet>
    <servlet-name>ODataServlet</servlet-name>
    <servlet-class>com.sap.odata.producer.servlets.ODataServlet</servlet-class>
    <init-param>
      <param-name>odataConfigFile</param-name>
      <param-value>/WEB-INF/odata_config.properties</param-value>
    </init-param>
  </servlet>

  <servlet-mapping>
    <servlet-name>ODataServlet</servlet-name>
    <url-pattern>*/</url-pattern>
  </servlet-mapping>

</web-app>
```

#### odata\_config.properties

```
# OData Producer options
# -----
Model = /WEB-INF/odata.osdl

# Connect to a SQLAnywhere Database
DbProduct = sqlanywhere

# Enable database authentication. This prompts for userID and password
Authentication = database

# Use the SQLAnywhere demo database as the data source
DbConnectionString = AUTOSTOP=no;DSN=SQL Anywhere 17 Demo

# ServiceRoot should match the webapps directory and url-pattern, without
# trailing /* from the web.xml file. In this example, the app is in
# $TOMCAT_HOME/webapps/odata/WEB-INF and url-pattern is /* so the service
# root is /odata (if the url-pattern was /great/* then the root would be
# /odata/great)
ServiceRoot = /odata
```

#### odata.osdl

```
service namespace "HttpsSample" {
  "GROUPO"."Employees";
  "GROUPO"."Departments";
  "GROUPO"."Customers";
  "GROUPO"."SalesOrders";
  "GROUPO"."SalesOrderItems";
  "GROUPO"."Products";
}
```

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. Please see <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> for additional trademark information and notices. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors.

National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP SE or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP SE or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.

