



Guideline



Project Acronym: PEPPOL
Grant Agreement number: 224974

Project Title: Pan-European Public Procurement Online



Developer Guide to the Java Access Point components for Oxalis

Version: 1.00 Status: In Use



Editors:

Jens Aabol, Difi



	Project co-funded by the European Commission within the ICT Policy Support Programme				
	Dissemination Level				
Р	Public	Х			
С	Confidential, only for members of the consortium and the Commission Services				





Revision History

Version	Date	Editor	Org	Description
1.00	28.03.2012	Jens Aabol	Difi	Initial version

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Statement of copyright



This deliverable is released under the terms of the **Creative Commons Licence** accessed through the following link: http://creativecommons.org/licenses/by/3.0/.

In short, it is free to

Share — to copy, distribute and transmit the work

Remix — to adapt the work

Under the following conditions

Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).





Contributors

Organizations

Difi – Agency for Public Management and eGovernment

Persons

Jens Aabol, Difi





Table of Contents

1	Introduction							
-		Objective and Scope						
2	Audi	ence	5					
		Components						
3 Installing Oxalis, PEPPOL access point example implementation								
	3.1	Implementation guideline	7					
		Using a JNDI data source						





1 Introduction

1.1 Objective and Scope

This document is the guideline for how to develop a PEPPOL access point when using the Difi code Oxalis. Oxalis is only made for Java installations.

Oxalis is made as open source and is located on https://github.com/difi/oxalis for downloading. The PEPPOL sample code is the basic code for Oxalis but Oxalis is a more plug and play version then the sample code.

You must run Apache and Tomcat together with Maven and a WSDL setup. All components are available as free downloads on the internet.

Additional to the Oxalis code you need to retrieve the PEPPOL certificate from the Regional Authority and confirm with the Regional Authority when to start testing and production.

On the Github site you will find the latest version for Oxalis.

Oxalis is an open source code and contributing to the code is desired. Governess for Oxalis is for the time being administrated by the Regional Authority – Difi in Norway.

2 Audience

The audience for this document is organizations in need for a short introduction to the PEPPOL BIS. These may include the following PEPPOL Stakeholders:

Service Providers

More specific it is the following roles:

- ICT Architects
- ICT Developers



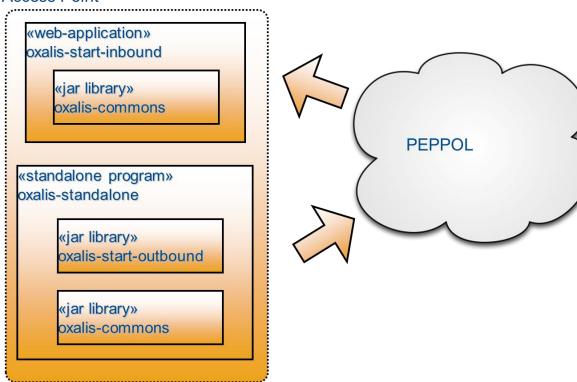


2.1 Components

Oxalis contains of an **inbound part** for messages that are received from other access points. These messages are put on an **input catalogue** for further internal services. The catalogue for the input messages can be changed to a database solution (an example is within the Oxalis download code).

Oxalis contains of an **outbound part** for messages that are to be sent to another accesspoint. Oxalis makes a **lookup to the SML/SMP** for retrieving the access point and documents of the receiver (script location: oxalis-start-outbound/src/main/assembly using the code fetch-metadata.sh). Oxalis sends the message to the correct access point.

Access Point







3 Installing Oxalis, PEPPOL access point example implementation.

3.1 Implementation guideline.

- 1. Download and install Java JDK version 6 or higher (Note: you must install the JDK, the JRE is not sufficient).
- 2. Download and install Apache Tomcat, we recommend version 7.0.22 or higher.
- 3. Enable SSL (https) in Tomcat by installing your SSL certificate. This guide assumes that you enable SSL on port 8443.
- 4. Download and install Apache Ant.
- 5. If you are running on a Windows machine, Set the following user variables (My Computer -> Properties -> Advanced -> Environment Variables)
- 6. JAVA_HOME should point to the Java JDK installation (ex. C:\Program Files\Java\idk1.6.0 30
- 7. CATALINA_HOME should point to the Apache Tomcat installation (ex. C:\Program Files\Apache Software Foundation\Tomcat 7.0)
- 8. TOMCAT_HOME should be identical to CATALINA_HOME

 ANT_HOME should point to the Apache Ant installation (ex. C:\Documents and Settings\Administrator\Desktop\apache-ant-1.8.2)
- 9. Add Apache Ant to your path (for Windows):

```
set PATH=%PATH%;%ANT HOME%/bin
```

 Download and install the Java Net Metro package. This package extends your JDK with a complete SOAP Web Services stack:

sudo ant -Dtomcat.home=\$TOMCAT_HOME -f metro-on-tomcat.xml install

If you're on a Windows machine, run this command:

ant -Dtomcat.home="%TOMCAT_HOME%" -f metro-on-tomcat.xml install

11. Install your keystore. We suggest shoving it into the Tomcat installation. Whatever you do, don't forget to specify the location in the Oxalis configuration file, as specified below.

This is how we would perform this task on a Linux or Mac OS X machine:

sudo -u tomcat cp ~/keystore.jks \$TOMCAT_HOME/conf/keystore

- 12. Modify \$TOMCAT/conf/catalina.properties to ensure that the Oxalis specific stuff is loaded by the Tomcat commons class loader by modifying the line starting with shared.loader=:
- 13. shared.loader=\${catalina.base}/shared/lib,\${catalina.home}/shared/lib/*.jar, \
- 14. \${catalina.base}/shared/lib/oxalis,\${catalina.base}/shared/lib/oxalis/*.jar
- 15. Copy the supplied sample-oxalis-web.properties til to \$TOMCAT_HOME/shared/lib/oxalis/oxalis-web.properties. This is how you would do it on a Linux or Mac OS X machine:

cp sample-oxalis-web.properties \$TOMCAT HOME/shared/lib/oxalis/oxalis-web.properties





16. Copy the .jar-files required into the Oxalis directory of Tomcat. This is how you would do that on a Linux or Mac OS X machine:

sudo -u tomcat cp *.jar \$TOMCAT_HOME/shared/lib/oxalis

17. Edit the Oxalis configuration file:

sudo -u tomcat vi \$TOMCAT_HOME/shared/lib/oxalis-web.properties

- 18. Start Tomcat
- 19. You should now be able to see the WSDL by going to the address: https://localhost:8443/oxalis/accessPointService?wsdl

3.2 Using a JNDI data source

If you intend to roll your own persistence mechanism by implementing the MessageRepository interface, this is how you configure your JDBC data source:

```
1. Add this XML fragment to $TOMCAT HOME/conf/server.xml in order to create a global data source:
```

```
<Resource name="jdbc/sr"</pre>
            auth="Container"
3.
            type="javax.sql.DataSource"
4.
5.
            maxActive="100"
6.
            maxIdle="30"
7.
            maxWait="10000"
8 .
            username="xxxxxxxxxx"
                                    <---- Don't forget to change this
            password="******* <----- Don't forget to change this
9.
10.
                    driverClassName="com.mysql.jdbc.Driver"
11.
    url="jdbc:mysql://localhost:3306/sendregning?autoReconnect=true"
12.
                   removeAbandoned="true"
13.
                    removeAbandonedTimeout="60"
14.
                    logAbandoned="true"
            />
15.
```

16. Link the global data source to the JNDI ENC name used by Oxalis:

```
17. ResourceLink name="jdbc/peppol-ap" global="jdbc/sr"
    type="javax.sql.DataSource"/>
```

18. The jar file containing your implementation of <code>MessageRepository</code>, should be copied into the <code>\$TOMCAT_HOME/shared/lib/oxalis</code> directory. Don't forget to include the file <code>META-INF/services/eu.peppol.start.persistence.MessageRepository</code> in the root of your .jar file in order to let Oxalis discover your implementation, using the standard Java services location mechanism.

