

Guideline



Acronym: PEPPOL
Grant Agreement number: 224974
Title: Pan-European Public Procurement Online



PEPPOL Transport Infrastructure Java Web Registration Developer Guide

Version: 1.0



Authors:

Christian Uldall Pedersen (NITA/Accenture)
Jens Jakob Andersen (NITA)
Klaus Vilstrup Pedersen (DIFI)



Project co-funded by the European Commission within the ICT Policy Support Programme

Dissemination Level

P	Public	X
C	Confidential, only for members of the consortium and the Commission Services	

Revision History

Version	Date	Author	Organisation	Description
1.0	20100215	Christian Uldall Pedersen	NITA/Accenture	First version (pending EC approval)

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This deliverable contains original unpublished work except where clearly indicated otherwise.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

Organisations

DIFI (Direktoratet for forvaltning og IKT)¹, Norway, www.difi.no
NITA (IT- og Telestyrelsen)², Denmark, www.itst.dk

Persons

Christian Uldall Pedersen (NITA/Accenture) (editor)
Jens Jakob Andersen, NITA
Klaus Vilstrup Pedersen, DIFI

¹ English: Agency for Public Management and eGovernment

² English: National IT- and Telecom Agency



Table of Content

1	Introduction	5
1.1	Objective	5
1.2	Scope	5
2	Prerequisites.....	5
2.1	Environment.....	5
2.2	Resources and libraries.....	6
3	Getting and Compiling the Source Code	6
4	Deploying the site.....	6
5	Configuring the website.....	7
5.1	Handling user access	7
5.2	Choosing language	7
5.3	Configuring profile roles	7
5.4	Other configuration	7

1 Introduction

1.1 Objective

This document is a developer guide to the web registration site for registering business identifiers on an SMP.

The site is currently located at:

<http://ec2-174-129-190-34.compute-1.amazonaws.com/RegistrationSite/welcome/welcome.do>

Note that the above link must be copied into your browser. It will not work if it is just clicked.

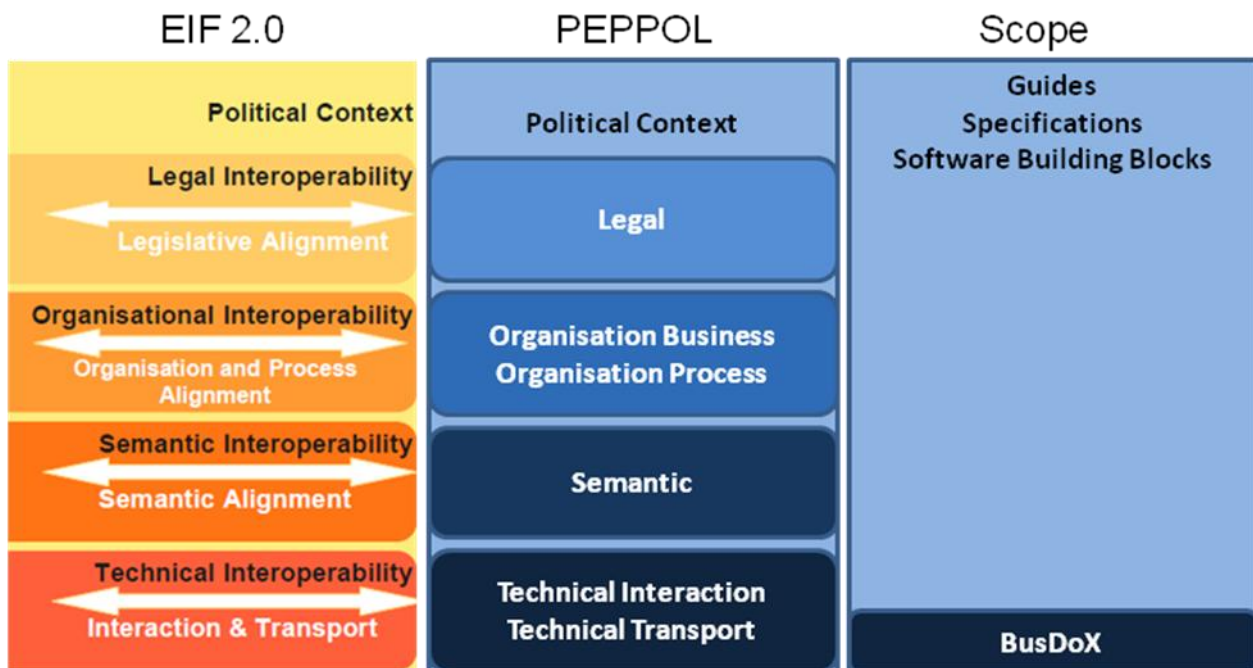
Username: peppol_user

Password: Test1234

The following browsers has been tested: Firefox 3, Internet Explorer 7 + 8, Safari 4

1.2 Scope

This guideline relates to the Technical Transport Layer i.e. BusDox specifications. The BusDox specifications can be used in many interoperability settings. In the PEPPOL context, it provides transport for procurement documents as specified in the PEPPOL Profiles.



2 Prerequisites

2.1 Environment

- JDK 6 or later
- Tomcat 6 server



Recommendations for development:

- Eclipse 3.2 or higher - <http://www.eclipse.org/>

Recommended tools for contributors to the online community

- Subclipse – Subversion plugin for Eclipse - <http://subclipse.tigris.org>
- TortoiseSVN –Subversion client for Windows - <http://tortoisesvn.net/>

The code has been tested on the following operating systems:

- Windows Server 2003
- Ubuntu 9.10

2.2 Resources and libraries

Required run-time libraries:

- Spring Framework 3 M3
- Spring Security
- Bouncy castle
- Apache Commons Codec, Collections, FileUpload, IO, Logging, Validator
- Google GDate Core

Required compile-time libraries (given these are present on the tomcat server):

- Servlet API

All of the above mentioned prerequisites are already present in the code repository.

3 Getting and Compiling the Source Code

The source code is located at:

<https://svn.forge.osor.eu/svn/peppol/java/ServiceMetadataRegistration/>, under the **v1.0** tags.

The component is added to the repository as an Eclipse project. The easiest way to build the project is therefore to import the project into Eclipse.

4 Deploying the site

The site can be deployed in two ways:

- Running the site through Eclipse. This can be done by right clicking the project root and choosing “Run as -> Run on server”



- Creating a war file by choosing “Export -> WAR file” and deploying this WAR file on Tomcat.

5 Configuring the website

5.1 Handling user access

Users are configured in the /WebContent/WEB-INF/Forms-security.xml file. Users are added in the <authentication-provider> element. An example of this element:

```
<authentication-provider>  
    <user-service>  
        <user name="peppol_user" password="Test1234" authorities="ROLE_USER"  
    />  
    </user-service>  
</authentication-provider>
```

Note that the users configured here must also be present in the corresponding SMP.

5.2 Choosing language

The language of the website can be changed by changing the *localeResolver* bean in the following configuration file: /WebContent/WEB-INF/regsiter-servlet.xml. An example of the *localeResolver* bean:

```
<bean id="localeResolver"  
class="org.springframework.web.servlet.i18n.FixedLocaleResolver">  
    <property name="defaultLocale" value="en" />  
</bean>
```

The language configuration files are located in the /cfg/ folder. New languages can be added here.

5.3 Configuring profile roles

Profile roles are configured using three configuration files located in the /res/ folder:

- CategoriesMapping.xml: Defines the categories which profile roles are places in.
- ProfileNameMapping.xml: Defines the human readable description for each profile role.
- ProfileRoleToServiceDefinitionMapping.xml: Defines the mapping between profile roles and services.

5.4 Other configuration

Other important configurable parameters are:

- /cfg/KeyTypes.xml: Contains the business identifiers which users should be able to choose on the web site.
- /cfg/config.properties:
 - service_smp: The URL to the SMP service.



- `registration_max_no`: The number of profile oriented registrations each user can create. This parameter is enforced by the web site, not the SMP service.
- `profile_conformance_claim`: The transport protocol to use when creating registrations.