



## **E -VLOŽIŠČE**

**SISTEM ZA VARNO ELEKTRONSKO VROČANJE**

# **Aplikacija ebms-sed (ebMS 3.0 Secure e-delivery)**

## **Namestitev**

**1.0**

## KONTROLA VERZIJ

### ZADNJA VERZIJA:

Verzija	1.0
Datum	02.11.2015
Avtor	Jože Rihtaršič
Odgovornost	Jože Rihtaršič
Zaupnost	/
Datoteka	

### ZGODOVINA:

Verzija	Datum	Avtor	Opis
1.0	02.11.2015	Jože Rihtaršič	Dokument kreiran.

### REVIZIJE:

Revizija	Datum	Avtor	Opis

## ZAŠČITA DOKUMENTA

© 2014 - 2015 Vrhovno sodišče Republike Slovenije

Vse pravice pridržane. Reprodukcijska po delih ali v celoti na kakršni koli način in na katerem koli mediju ni dovoljena brez pisnega dovoljenja avtorja. Omejitve ne veljajo za državne organe Republike Slovenije.

Vsaka kršitev se lahko preganja v skladu z Zakonom o avtorski in sorodnih pravicah in Kazenskim zakonikom Republike Slovenije

## Kazalo vsebine

Namen dokumenta.....	4
Priprava na namestitev.....	4
Namestitev objektov podatkovne baze.....	4
Namestitev wildfly aplikacijskega strežnika.....	4
Modul: org.apache.ws.security.....	4
Nastavitve Wildfly JMS.....	5
Nastavitve wildfly: ebms-sed.....	7
Dodajanje jdbc knjižnice za postgres.....	7
Izgradnja aplikacije.....	8

## Namen dokumenta

Dokument opisuje tehnične značilnosti aplikacije ebms-sed. Namenjen je tehničnemu osebju kot pomoč pri namestitvi in vzdrževanju aplikacije.

## Priprava na namestitev

Aplikacija ebms-sdev za delovanje uporablja java 1.8 (ali novejšo), podatkovno bazo postgresql in aplikacijski strežnik wildfly-9.0.2.Final (ali novejši). Za izgradnjo je potrebno imeti nameščeno aplikacijo maven 3.x

Omenjene aplikacije so dosegljive na naslovu:

- Java SDK: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Podatkovna baza postgresql: <http://www.postgresql.org/download/>
- Aplikacijski strežnik: wildfly: <http://wildfly.org/downloads/>
- Maven: <https://maven.apache.org/download.cgi>

Pred pripravo je potrebno namestiti omenjene aplikacije.

## Namestitev objektov podatkovne baze

Po namestitvi baze izdelaj baznega uporabnika „ebms-sed“. (Uporabniško ime je lahko tudi drugačno.)

Nato poženi skripto za izdelavo objektov podatkovne baze:

```
psql -U ebms-sed -d [my-db] -a -f ebms-sed/scripts/db/postgres_create_db.sql
```

## Namestitev wildfly aplikacijskega strežnika

### **Modul: org.apache.ws.security**

Aplikacija trenutno uporablja apache-cxf knjižnice, ki so nameščeni v wildfly aplikacijskem strežniku. Za uporabo je potreben manjši popravek v namestitveni datoteki:

wildfly-9.0.2.Final/modules/system/layers/base/org/apache/ws/security/main/module.xml

kjer je med odvisne module potrebno dodati vrstico:

```
<module name="javax.mail.api" />
```

modul org.apache.ws.security ima tako odvisne module:

```
<dependencies>
  <module name="javax.api" />
  <module name="javax.mail.api" />
  <module name="javax.xml.bind.api" services="import"/>
  <module name="com.sun.xml.bind" services="import"/>
  <module name="javax.xml.rpc.api" />
  <module name="org.apache.commons.codec" />
  <module name="org.apache.commons.logging" />
```

```

    <module name="org.apache.neethi" />
    <module name="org.apache.santuario.xmlsec" />
    <module name="org.apache.xalan" />
    <module name="org.joda.time" />
    <module name="org.opensaml" />
    <module name="org.slf4j" />
  </dependencies>
</module>

```

## Nastavitve Wildfly JMS

V mapi: wildfly-9.0.2.Final/standalone/configuration

naredi kopijo: standalone.xml

```
cp standalone.xml standalone-ebms.xml
```

V razdelek:

```

<server xmlns="urn:jboss:domain:3.0">
  <extensions> ....

```

Dodaj dodatek:

```
<extension module="org.jboss.as.messaging"/>
```

Nato dodaj modul urn:jboss:domain:messaging:3.0:

```

<subsystem xmlns="urn:jboss:domain:messaging:3.0">
  <hornetq-server>
    <journal-file-size>102400</journal-file-size>
    <connectors>
      <http-connector name="http-connector" socket-binding="http">
        <param key="http-upgrade-endpoint" value="http-acceptor"/>
      </http-connector>
      <http-connector name="http-connector-throughput" socket-binding="http">
        <param key="http-upgrade-endpoint" value="http-acceptor-throughput"/>
        <param key="batch-delay" value="50"/>
      </http-connector>
      <in-vm-connector name="in-vm" server-id="0"/>
    </connectors>
    <acceptors>
      <http-acceptor name="http-acceptor" http-listener="default"/>
      <http-acceptor name="http-acceptor-throughput" http-listener="default">
        <param key="batch-delay" value="50"/>
        <param key="direct-deliver" value="false"/>
      </http-acceptor>
      <in-vm-acceptor name="in-vm" server-id="0"/>
    </acceptors>
    <security-settings>
      <security-setting match="#">
        <permission type="send" roles="guest"/>
        <permission type="consume" roles="guest"/>
        <permission type="createNonDurableQueue" roles="guest"/>
        <permission type="deleteNonDurableQueue" roles="guest"/>
      </security-setting>
    </security-settings>
  </hornetq-server>
</subsystem>

```

```

</security-settings>
<address-settings>
  <!--default for catch all-->
  <address-setting match="#">
    <dead-letter-address>jms.queue.DLQ</dead-letter-address>
    <expiry-address>jms.queue.ExpiryQueue</expiry-address>
    <max-size-bytes>10485760</max-size-bytes>
    <page-size-bytes>2097152</page-size-bytes>
    <message-counter-history-day-limit>10</message-counter-history-day-limit>
  </address-setting>
</address-settings>
<jms-connection-factories>
  <connection-factory name="InVmConnectionFactory">
    <connectors>
      <connector-ref connector-name="in-vm"/>
    </connectors>
    <entries>
      <entry name="java:/ConnectionFactory"/>
    </entries>
  </connection-factory>
  <connection-factory name="RemoteConnectionFactory">
    <connectors>
      <connector-ref connector-name="http-connector"/>
    </connectors>
    <entries>
      <entry name="java:jboss/exported/jms/RemoteConnectionFactory"/>
    </entries>
  </connection-factory>
  <pooled-connection-factory name="hornetq-ra">
    <transaction mode="xa"/>
    <connectors>
      <connector-ref connector-name="in-vm"/>
    </connectors>
    <entries>
      <entry name="java:/JmsXA"/>
      <!-- Global JNDI entry used to provide a default JMS Connection factory to EE application
-->
      <entry name="java:jboss/DefaultJMSConnectionFactory"/>
    </entries>
  </pooled-connection-factory>
</jms-connection-factories>
<jms-destinations>
  <jms-queue name="MSHQueue">
    <entry name="java:/jms/queue/MSHQueue"/>
  </jms-queue>
  <jms-queue name="ExpiryQueue">
    <entry name="java:/jms/queue/ExpiryQueue"/>
  </jms-queue>
  <jms-queue name="DLQ">
    <entry name="java:/jms/queue/DLQ"/>
  </jms-queue>
</jms-destinations>

```

```
</hornetq-server>
</subsystem>
```

V modul `<subsystem xmlns="urn:jboss:domain:ejb3:1.2">`

Dodaj:

```
<mdb>
  <resource-adapter-ref resource-adapter-name="{ejb.resource-adapter-name:hornetq-ra.rar}"/>
  <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
</mdb>
```

## ***Nastavitve wildfly: ebms-sed***

Aplikacija ima trenutno dve nastavitvi:

- sed.home.dir: ebms-sed „domača mapa“, kjer so nastavitve p-mode in cache mape:
- org.sed.msh.maxWorkers: število obdelovalnih vrst za izhodno pošto.

Najprej določi SED\_HOME mapo in vanjo posnami pripravljene kofiguracijo:

ebms-sed/documentation/samples/sed-home

v datoteko:

wildfly-9.0.2.Final/bin/standalone.conf

dodaj nastavitvi:

# SED HOME DIR

JAVA\_OPTS="\$JAVA\_OPTS -Dsed.home.dir=[POT DO SED\_HOME MAPE]/sed-home"

#MAX WORKERS FOR outgoing messages

JAVA\_OPTS="\$JAVA\_OPTS -Dorg.sed.msh.maxWorkers=2"

## ***Dodajanje jdbc knjižnice za postgres***

Na spletni strani „<https://jdbc.postgresql.org/download.html>“ pridobi primereno jdbc knjižnico za nameščeno postgresql bazo.

Postgresql driver se bo namestil s pomočjo jboss-cli ukazov, zato v mapi:

wildfly-9.0.2.Final/bin

določi wildfly admin uporabnika tako da požnem skripto (in izbereš userType: a) Management User):

./add-user.sh oz. ./add-user.bat

Nato poženi aplikacijski strežnik.

./standalone.sh -c standalone-ebms

oziroma:

./standalone.bat -c standalone-ebms

Po končanem štartu strežnika zaženini

./jboss-cli.sh oz. ./jboss-cli.bat

V ukazno vrstico nato vpiši `connect` in [enter]

Za namestitev ojdbc knjižnice prekopirajo ukaz:

```
[standalone@localhost:9990 /] module add --name=org.postgres -  
resources=[potDoMapeZKnjiznico]/postgresql-9.4-1204.jdbc42.jar  
--dependencies=javax.api,javax.transaction.api
```

Nato določi ojdbc knjižnico

```
[standalone@localhost:9990 /]/subsystem=datasources/jdbc-  
driver=postgres:add(driver-name="postgres",driver-module-  
name="org.postgres",driver-class-name=org.postgresql.Driver
```

Nato določi povezavo na bazo:

```
[standalone@localhost:9990 /]data-source add --jndi-  
name=java:/dsEBMS_SED --name=ebMSSEDPool --connection-  
url=jdbc:postgresql://localhost/db_sed --driver-name=postgres  
--user-name=[USERNAME] --password=[Geslo]
```

## Izgradnja aplikacije

V mapi `ebms-sed`

poženi maven ukaz za izgradnjo aplikacije:

```
mvn clean install
```

Pri tem se izgradita modula:

```
ebms-sed/ebms-msh-module/ebms-msh-ear/target/ebms-msh.ear  
in
```

```
ebms-sed-module/ebms-sed-ws/target/ebms-sed-ws.war
```

Modula posnami v wildfly namestitveno map:

```
wildfly-9.0.2.Final/standalone/deployments
```

Po upešni nemestivti sta vidni dva sklopa spletnih storitev:

**ebMS 3.0 spletne storitve:**

```
http://localhost:8080/ebms-msh/ebms?wsdl
```

in interne spletne storitve za pošiljanje in prevzem dohodne pošte

```
http://localhost:8080/ebms-sed-ws/SEDMailBoxWS?wsdl
```