

Payara 5 – Konfiguration

payara5/glassfish/config/asenv.conf

AS_JAVA="`<java home directory>`"

AS_IMQ_LIB="`../mq/lib`"

AS_IMQ_BIN="`../mq/bin`"

AS_CONFIG="`../config`"

AS_INSTALL="`..`"

AS_DEF_DOMAINS_PATH="`../domains`"

AS_DEF_NODES_PATH="`../nodes`"

AS_DERBY_INSTALL="`../javadb`"

AS_H2_INSTALL="`../h2db`"

AS_JAVA="`/home/maximilian/Dokumente/opt/jdk-8u181-linux-x64/jdk1.8.0_181`"

sudo chmod +x payara5/bin/asadmin

in payara5/bin ./asadmin ausführen

danach start-domain



```
maximilian@maximilian-Aspire-ES1-523: ~/Dokumente/opt/payara-5.193.1/payara5/bin
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
maximilian@maximilian-Aspire-ES1-523:~/Dokumente/opt/payara-5.193.1/payara5/bin$ ./asadmin
Use "exit" to exit and "help" for online help.
Remote commands not fetched
asadmin> start-domain
Waiting for domain1 to start .....
Successfully started the domain : domain1
domain Location: /home/maximilian/Dokumente/opt/payara-5.193.1/payara5/glassfish/domains/domain1
Log File: /home/maximilian/Dokumente/opt/payara-5.193.1/payara5/glassfish/domains/domain1/logs/server.log
Admin Port: 4848
Command start-domain executed successfully.
asadmin>
```

im Browser auf localhost:4848 gehen

set admin password

change admin password

logout

login

user: admin

password: passme

create connection pool:

Nodes

Clusters (Deprecated)

Applications

Lifecycle Modules

Monitoring Data

Resources

Concurrent Resources

Connectors

JDBC

JDBC Resources

JDBC Connection Pools

H2Pool

__TimerPool

derby-connector

JMS Resources

JDBC Connection Pools

To store, organize, and retrieve data, most applications use relational databases. Java EE applications access relational databases through the JDBC API. Before an application can access a database, it must get a connection.

Pools (3)

New...Delete

Select	Pool Name	Resource Type	Classname	Description
<input type="checkbox"/>	H2Pool	javax.sql.DataSource	org.h2.jdbcx.JdbcDataSource	
<input type="checkbox"/>	__TimerPool	javax.sql.XADataSource	org.apache.derby.jdbc.EmbeddedXADataSource	
<input type="checkbox"/>	derby-connector	javax.sql.DataSource	org.apache.derby.jdbc.ClientDataSource40	

New JDBC Connection Pool ->

Pool Name:

derby-connector

Resource Type:

javax.sql.DataSource

Must be specified if the datasource class implements more than 1 of the interface.

Datasource Classname:

org.apache.derby.jdbc.ClientDataSource40

Vendor-specific classname that implements the DataSource and/or XADataSource APIs

Driver Classname:

Vendor-specific classname that implements the java.sql.Driver interface.

Ping:

☒ Enabled

When enabled, the pool is pinged during creation or reconfiguration to identify and warn of any erroneous values for its attributes

Additional Properties ->

Additional Properties (5)

Add PropertyDelete Properties

Select	Name	Value	Description
<input type="checkbox"/>	databaseName	db	
<input type="checkbox"/>	portNumber	1527	
<input type="checkbox"/>	serverName	localhost	
<input type="checkbox"/>	user	app	
<input type="checkbox"/>	password	app	

New JDBC Resource ->

Monitoring Data

Resources

Concurrent Resources

Connectors

JDBC

JDBC Resources

dbDs

jdbc/__TimerPool

jdbc/__default

JDBC Connection Pools

JMS Resources

Resources (3)

New...DeleteEnableDisable

Select	JNDI Name	Logical JNDI Name	Enabled	Connection Pool	Description
<input type="checkbox"/>	dbDs		<input checked="" type="checkbox"/>	derby-connector	
<input type="checkbox"/>	jdbc/__TimerPool		<input checked="" type="checkbox"/>	__TimerPool	
<input type="checkbox"/>	jdbc/__default	java.comp.DefaultDataSource	<input checked="" type="checkbox"/>	H2Pool	

Load Defaults

JNDI Name: dbDs

Pool Name: derby-connector ▼
Use the [JDBC Connection Pools](#) page to create new pools

Deployment Order: 100
Specifies the loading order of the resource at server startup. Lower numbers are loaded first.

Description:

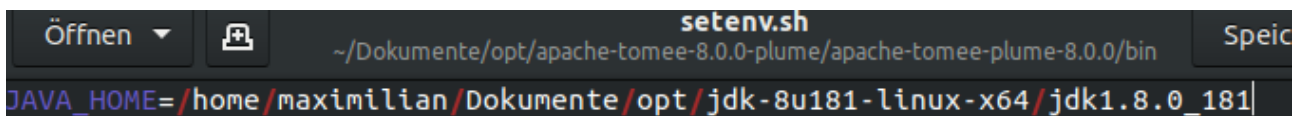
Status: ☒ Enabled

persistence.xml ->

```
<!--<jta-data-source>dbDS</jta-data-source>
<properties>
  <property name="eclipselink.ddl-generation" value="create-tables"/>
</properties-->
```

Glassfish Server → Payara5 Pfad auswählen → Domain auf domain1 setzen → Fix Deployment

TomEE



in bin Ordner setenv.sh File erstellen und JAVA_HOME setzen

im bin Ordner ./startup.sh ausführen

```
maximilian@maximilian-Aspire-E51-523:~/Dokumente/opt/apache-tomee-8.0.0-plume/apache-tomee-plume-8.0.0/bin$ ./star
tup.sh
Using CATALINA_BASE:   /home/maximilian/Dokumente/opt/apache-tomee-8.0.0-plume/apache-tomee-plume-8.0.0
Using CATALINA_HOME:   /home/maximilian/Dokumente/opt/apache-tomee-8.0.0-plume/apache-tomee-plume-8.0.0
Using CATALINA_TMPDIR: /home/maximilian/Dokumente/opt/apache-tomee-8.0.0-plume/apache-tomee-plume-8.0.0/temp
Using JRE_HOME:        /home/maximilian/Dokumente/opt/jdk-8u181-linux-x64/jdk1.8.0_181
Using CLASSPATH:       /home/maximilian/Dokumente/opt/apache-tomee-8.0.0-plume/apache-tomee-plume-8.0.0/bin/bootst
rap.jar:/home/maximilian/Dokumente/opt/apache-tomee-8.0.0-plume/apache-tomee-plume-8.0.0/bin/tomcat-juli.jar
Tomcat started.
```

wenn kein Fehler auftritt, dann ./shutdown.sh ausführen

in conf/tomcat-users.xml folgende Zeilen hinzufügen:

```
<tomcat-users xmlns="http://tomcat.apache.org/xml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
  version="1.0">
  <role rolename="manager-gui" />
  <user username="admin"
    password="passme"
    roles="tomee-admin,manager-gui,admin-gui" />
```

derbyclient.jar (derbydb) in lib Ordner kopieren
in conf/tomee.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<tomee>
  <!-- see http://tomee.apache.org/containers-and-resources.html -->

  <!-- activate next line to be able to deploy applications in apps -->
  <!-- <Deployments dir="apps" /> -->
    <Resource id="jdbc/dbDS" type="DataSource">
      #dbDS Datasource

      JdbcDriver org.apache.derby.jdbc.ClientDriver
      JdbcUrl jdbc:derby://localhost:1527/db;create=true
      UserName app
      Password app
    </Resource>
</tomee>
```

persistence.xml (kein Hibernate)

```
<jta-data-source>jdbc/dbDS</jta-data-source>
<properties>
  <property name="javax.persistence.schema-generation.database.action" value="drop-and-create"/>
</properties>
```

Edit Configuration → TomEE Server Local → Pfad auswählen → Fix Deployment

Glassfish

in glassfish5/glassfish/config/asenv.conf JAVA_HOME setzen:

```
AS_IMQ_LIB="../../../mq/lib"
AS_IMQ_BIN="../../../mq/bin"
AS_CONFIG="../../../config"
AS_INSTALL=".."
AS_DEF_DOMAINS_PATH="../../domains"
AS_DEF_NODES_PATH="../../nodes"
AS_DERBY_INSTALL="../../../javadb"
AS_JAVA="/home/maximilian/Dokumente/opt/jdk-8u181-linux-x64/jdk1.8.0_181"
```

in glassfish5/bin ./asadmin ausführen

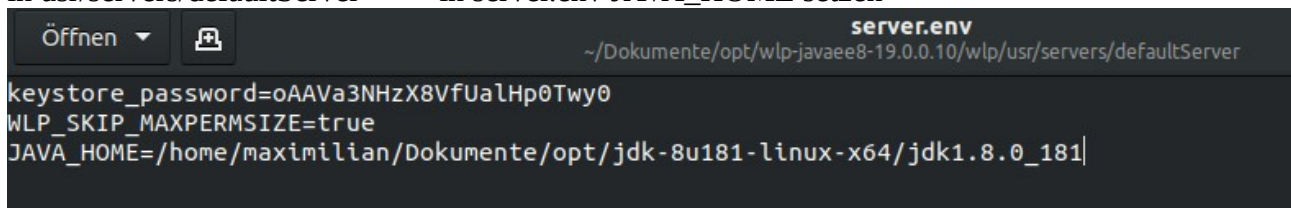
```
maximilian@maximilian-Aspire-ES1-523:~/Dokumente/opt/glassfish-5.0/glassfish5$ cd bin/
maximilian@maximilian-Aspire-ES1-523:~/Dokumente/opt/glassfish-5.0/glassfish5/bin$ ls
asadmin asadmin.bat
maximilian@maximilian-Aspire-ES1-523:~/Dokumente/opt/glassfish-5.0/glassfish5/bin$ ./asadmin
Use "exit" to exit and "help" for online help.
asadmin> start-domain
Waiting for domain1 to start .....
Successfully started the domain : domain1
domain Location: /home/maximilian/Dokumente/opt/glassfish-5.0/glassfish5/glassfish/domains/domain1
Log File: /home/maximilian/Dokumente/opt/glassfish-5.0/glassfish5/glassfish/domains/domain1/logs/server.log
Admin Port: 4848
Command start-domain executed successfully.
asadmin> █
```

ab dann gleich wie Payara5 Server Config

Websphere Liberty

`./server create -->` im bin Ordner ausführen

in `usr/servers/defaultServer` --> in `server.env` `JAVA_HOME` setzen

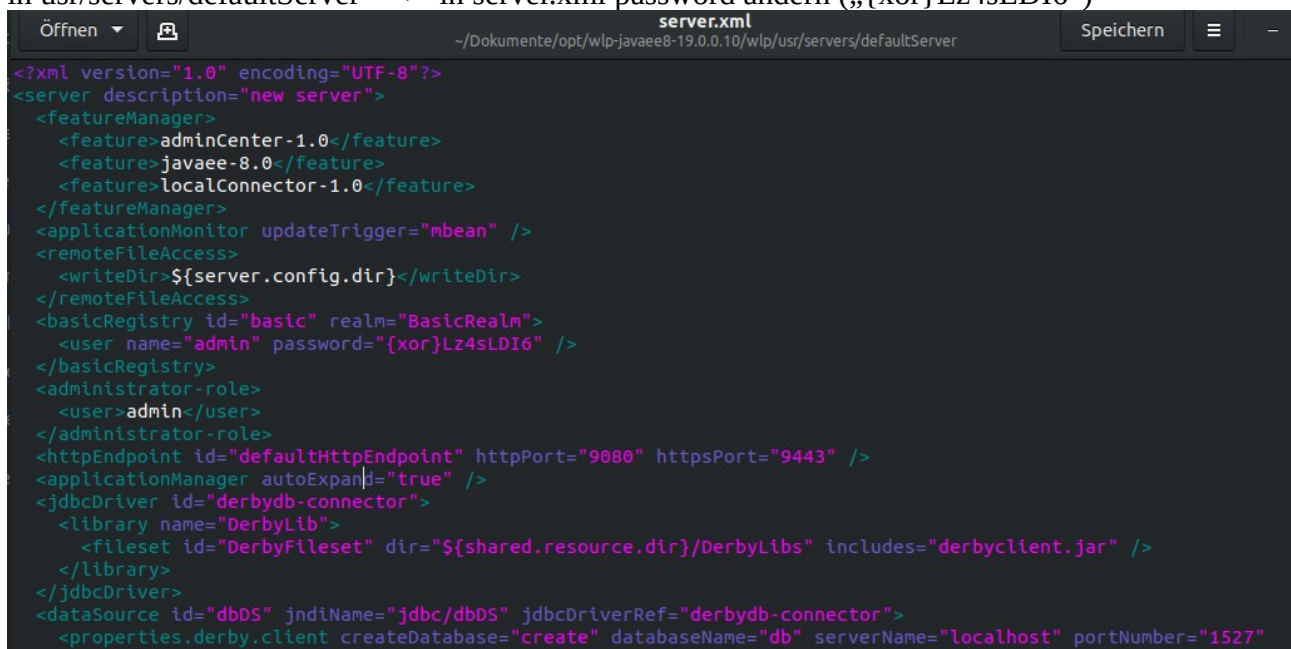


```
server.env
~/Dokumente/opt/wlp-javaee8-19.0.0.10/wlp/usr/servers/defaultServer

keystore_password=oAAVa3NHZx8VfUalHp0Twy0
WLP_SKIP_MAXPERMSIZE=true
JAVA_HOME=/home/maximilian/Dokumente/opt/jdk-8u181-linux-x64/jdk1.8.0_181
```

`./installUtility install adminCenter-1.0 -->` in bin Ordner ausführen

in `usr/servers/defaultServer` --> in `server.xml` password ändern („{xor}Lz4sLDI6“)



```
server.xml
~/Dokumente/opt/wlp-javaee8-19.0.0.10/wlp/usr/servers/defaultServer

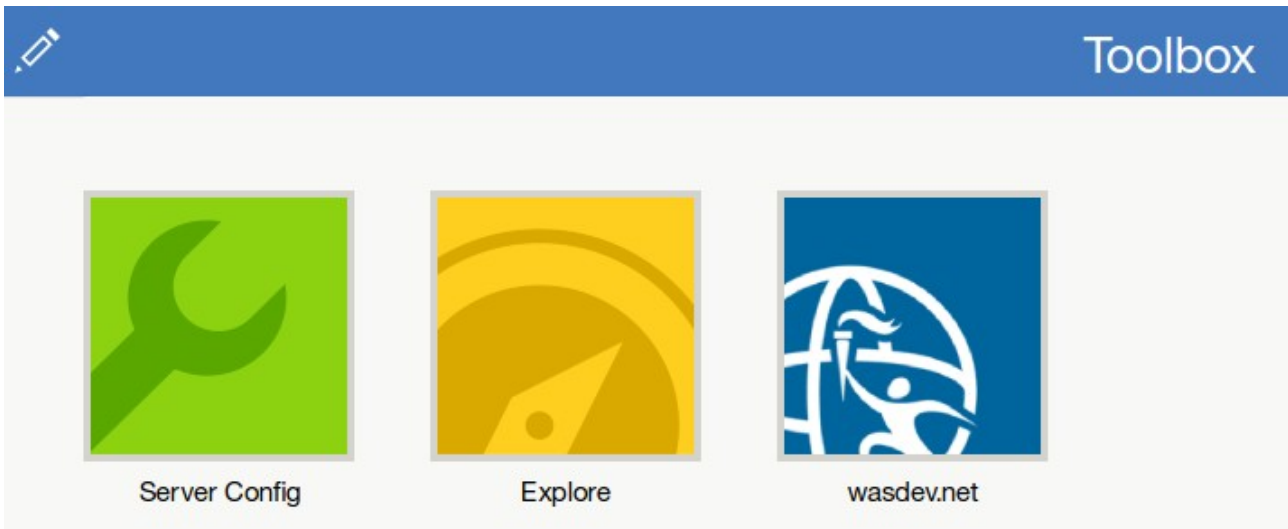
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">
  <featureManager>
    <feature>adminCenter-1.0</feature>
    <feature>javaee-8.0</feature>
    <feature>localConnector-1.0</feature>
  </featureManager>
  <applicationMonitor updateTrigger="mbean" />
  <remoteFileAccess>
    <writeDir>${server.config.dir}</writeDir>
  </remoteFileAccess>
  <basicRegistry id="basic" realm="BasicRealm">
    <user name="admin" password="{xor}Lz4sLDI6" />
  </basicRegistry>
  <administrator-role>
    <user>admin</user>
  </administrator-role>
  <httpEndpoint id="defaultHttpEndpoint" httpPort="9080" httpsPort="9443" />
  <applicationManager autoExpand="true" />
  <jdbcDriver id="derbydb-connector">
    <library name="DerbyLib">
      <fileset id="DerbyFileset" dir="${shared.resource.dir}/DerbyLibs" includes="derbyclient.jar" />
    </library>
  </jdbcDriver>
  <dataSource id="dbDS" jndiName="jdbc/dbDS" jdbcDriverRef="derbydb-connector">
    <properties.derby.client createDatabase="create" databaseName="db" serverName="localhost" portNumber="1527" />
  </dataSource>
</server>
```

`./server start -->` in bin Ordner ausführen

`localhost:9443/adminCenter -->` im Browser öffnen

Benutzername: admin

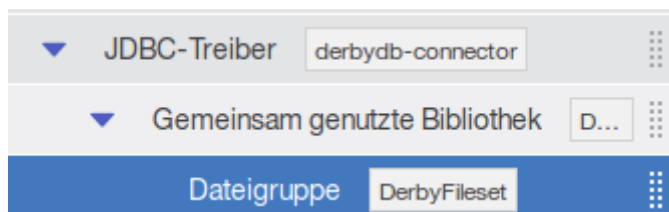
Passwort: passme



Server Config → server.xml

Untergeordnetes Element hinzufügen:

1. JDBC-Treiber
2. id: derbydb-connector
3. Untergeordnetes Element hinzufügen
 1. Gemeinsam genutzte Bibliothek
 2. Name: DerbyLib
 3. Untergeordnetes Element hinzufügen
 1. Dateigruppe
 2. id: DerbyFileset
 3. Basisverzeichnis: \${shared.resource.dir}/DerbyLibs
 4. Einschlussmuster: derbyclient.jar



Untergeordnetes Element hinzufügen:

1. Datenquelle
2. id: dbDS
3. JNDI Name: jdbc/dbDS
4. JDBC-Treiberreferenz: derbydb-connector

5. Untergeordnetes Element hinzufügen
 1. Eigenschaften für Derby Network Client

Datenbank erstellen



Auswählen

JDBC-Treibereigenschaft: createDatabase

Datenbankname



JDBC-Treibereigenschaft: databaseName

Servername



Server, auf dem die Datenbank ausgeführt wird

Portnummer



Port, an dem Datenbankverbindungen abgerufen werden.

Weitere Eigenschaften

Weitere JDBC-Anbiitereigenschaften für die Datenquelle.

```
<jta-data-source>jdbc/dbDS</jta-data-source>
<properties>
  <property name="javax.persistence.schema-generation.database.action"
    value="drop-and-create"/>
</properties>
```

Edit Configuration → WebSphere Server → Pfad auswählen → Fix Deployment