Pentaho Server 8 CE: User/Role List...


Joao Ciocca 15 posts since Dec 20, 2017

Pentaho Server 8 CE: User/Role List could not be obtained. Apr 11, 2018 9:54 PM

So, we've set up an installation of 8.0 CE, followed the docs, server is up and running... but can't login. Admin/password or Suzy/password will result in Login Error. Behind the scenes, catalina.out says:

```
18:27:01,378 INFO  [PeriodicStatusLogger] The system has finished initializing.
[pt_47] Pentaho BI Platform server is ready.
18:27:01,561 ERROR [CompositeUserRoleListService] User/Role List could not be obtained.
java.lang.IllegalStateException: Target of Bean was never resolved:
org.springframework.security.core.userdetails.UserDetailsService
        at org.pentaho.platform.engine.core.system.objfac.spring.BeanBuilder
$1.invoke(BeanBuilder.java:157)
        at com.sun.proxy.$Proxy83.loadUserByUsername(Unknown Source)
        at
org.pentaho.platform.plugin.services.security.userrole.ChainedUserDetailsService.loadUserByUsername(Cha
        at
org.pentaho.platform.plugin.services.security.userrole.PentahoCachingUserDetailsService.loadUserByUser
        at
org.pentaho.platform.security.userroledao.service.UserRoleDaoUserRoleListService.getRolesForUser(UserR
        at
org.pentaho.platform.plugin.services.security.userrole.ExtraRolesUserRoleListServiceDecorator.getRoles
        at
org.pentaho.platform.plugin.services.security.userrole.CompositeUserRoleListService
$7.perform(CompositeUserRoleListService.java:122)
        at
org.pentaho.platform.plugin.services.security.userrole.CompositeUserRoleListService.collectResultsForO
        at
org.pentaho.platform.plugin.services.security.userrole.CompositeUserRoleListService.getRolesForUser(Co
        at
org.pentaho.platform.engine.security.SecurityHelper.createAuthentication(SecurityHelper.java:352)
        at
org.pentaho.platform.engine.security.SecurityHelper.runAsSystem(SecurityHelper.java:415)
        at pt.webdetails.cpf.CpfProperties.loadAsSystem(CpfProperties.java:46)
        at
pt.webdetails.cpf.AbstractCpfProperties.loadSettings(AbstractCpfProperties.java:58)
        at pt.webdetails.cpf.AbstractCpfProperties.<init>(AbstractCpfProperties.java:35)
        at pt.webdetails.cpf.CpfProperties.<init>(CpfProperties.java:32)
        at pt.webdetails.cpf.CpfProperties.getInstance(CpfProperties.java:37)
        at pt.webdetails.cpf.SimpleLifeCycleListener.ready(SimpleLifeCycleListener.java:45)
        at
org.pentaho.platform.web.http.context.PentahoSystemReadyListener.contextInitialized(PentahoSystemReady
        at
org.apache.catalina.core.StandardContext.listenerStart(StandardContext.java:4853)
        at
org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:5314)
        at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:145)
        at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:753)
```

```
        at org.apache.catalina.core.ContainerBase.addChild(ContainerBase.java:729)
        at org.apache.catalina.core.StandardHost.addChild(StandardHost.java:717)
        at org.apache.catalina.startup.HostConfig.deployDirectory(HostConfig.java:1129)
        at org.apache.catalina.startup.HostConfig$DeployDirectory.run(HostConfig.java:1871)
        at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
        at java.util.concurrent.FutureTask.run(FutureTask.java:266)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:748)
```

And when attempting to login...

```
18:31:14,057 ERROR [UsernamePasswordAuthenticationFilter] An internal error occurred while
trying to authenticate the user.
org.springframework.security.authentication.InternalAuthenticationServiceException: Target
of Bean was never resolved:
org.springframework.security.core.userdetails.UserDetailsService
        at
org.springframework.security.authentication.dao.DaoAuthenticationProvider.retrieveUser(DaoAuthenticati
        at
org.springframework.security.authentication.dao.AbstractUserDetailsAuthenticationProvider.authenticate
        at
org.springframework.security.authentication.ProviderManager.authenticate(ProviderManager.java:174)
        at
org.springframework.security.web.authentication.UsernamePasswordAuthenticationFilter.attemptAuthentica
        at
org.springframework.security.web.authentication.AbstractAuthenticationProcessingFilter.doFilter(Abstra
        at org.springframework.security.web.FilterChainProxy
$VirtualFilterChain.doFilter(FilterChainProxy.java:331)
        at
org.springframework.security.web.authentication.logout.LogoutFilter.doFilter(LogoutFilter.java:121)
        at org.springframework.security.web.FilterChainProxy
$VirtualFilterChain.doFilter(FilterChainProxy.java:331)
        at
org.pentaho.platform.web.http.security.HttpSessionReuseDetectionFilter.doFilter(HttpSessionReuseDetecti
        at org.springframework.security.web.FilterChainProxy
$VirtualFilterChain.doFilter(FilterChainProxy.java:331)
        at
org.springframework.security.web.context.SecurityContextPersistenceFilter.doFilter(SecurityContextPers
        at org.springframework.security.web.FilterChainProxy
$VirtualFilterChain.doFilter(FilterChainProxy.java:331)
        at
org.pentaho.platform.web.http.filters.HttpSessionPentahoSessionIntegrationFilter.doFilter(HttpSessionP
        at org.springframework.security.web.FilterChainProxy
$VirtualFilterChain.doFilter(FilterChainProxy.java:331)
        at
org.springframework.security.web.servletapi.SecurityContextHolderAwareRequestFilter.doFilter(SecurityC
        at org.springframework.security.web.FilterChainProxy
$VirtualFilterChain.doFilter(FilterChainProxy.java:331)
        at
org.springframework.security.web.FilterChainProxy.doFilterInternal(FilterChainProxy.java:214)
        at
org.springframework.security.web.FilterChainProxy.doFilter(FilterChainProxy.java:177)
```

```
        at
org.springframework.web.filter.DelegatingFilterProxy.invokeDelegate(DelegatingFilterProxy.java:346)
        at
org.springframework.web.filter.DelegatingFilterProxy.doFilter(DelegatingFilterProxy.java:262)
        at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:240)
        at
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:207)
        at
org.pentaho.platform.web.http.filters.SystemStatusFilter.doFilter(SystemStatusFilter.java:55)
        at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:240)
        at
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:207)
        at
org.pentaho.platform.web.http.filters.SetCharacterEncodingFilter.doFilter(SetCharacterEncodingFilter.ja
        at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:240)
        at
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:207)
        at
org.pentaho.platform.web.http.filters.WebappRootForwardingFilter.doFilter(WebappRootForwardingFilter.ja
        at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:240)
        at
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:207)
        at
org.pentaho.platform.web.http.filters.PentahoPathDecodingFilter.doFilter(PentahoPathDecodingFilter.java
        at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:240)
        at
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:207)
        at
org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:212)
        at
org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:94)
        at
org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:504)
        at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:141)
        at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:79)
        at
org.apache.catalina.valves.AbstractAccessLogValve.invoke(AbstractAccessLogValve.java:620)
        at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:88)
        at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:502)
        at
org.apache.coyote.http11.AbstractHttp11Processor.process(AbstractHttp11Processor.java:1132)
        at org.apache.coyote.AbstractProtocol
$AbstractConnectionHandler.process(AbstractProtocol.java:684)
        at org.apache.tomcat.util.net.NioEndpoint
$SocketProcessor.doRun(NioEndpoint.java:1539)
        at org.apache.tomcat.util.net.NioEndpoint
$SocketProcessor.run(NioEndpoint.java:1495)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
```

```
        at org.apache.tomcat.util.threads.TaskThread
$WrappingRunnable.run(TaskThread.java:61)
        at java.lang.Thread.run(Thread.java:748)
Caused by: java.lang.IllegalStateException: Target of Bean was never resolved:
org.springframework.security.core.userdetails.UserDetailsService
        at org.pentaho.platform.engine.core.system.objfac.spring.BeanBuilder
$1.invoke(BeanBuilder.java:157)
        at com.sun.proxy.$Proxy83.loadUserByUsername(Unknown Source)
        at
org.pentaho.platform.plugin.services.security.userrole.ChainedUserDetailsService.loadUserByUsername(Cha
        at
org.pentaho.platform.plugin.services.security.userrole.PentahoCachingUserDetailsService.loadUserByUser
        at
org.springframework.security.authentication.dao.DaoAuthenticationProvider.retrieveUser(DaoAuthenticati
        ... 49 more
```

I've retraced all setup steps, everything is correct, the server is connected to the postgres database, according to netstat, but nothing. I've googled around but found nothing helpful. Ideas?
Tags: login_issues, login_failed, pentaho ce

 Joao Ciocca *15 posts since Dec 20, 2017*

Re: Pentaho Server 8 CE: User/Role List could not be obtained. Apr 13, 2018 12:47 PM

Figured out: the problem was caused by disabling the HSQLDB connections, without configuring a proper new auth method. We were disabling it and enabling postgreSQL following the documentation, but there's nothing on the docs that say the default users (admin, suzy, joe) don't get loaded on the postgreSQL create scripts - and we just assumed they were.

Once we switched back HSQLDB, everything worked out fine.

 Alexis Araya *3 posts since Sep 13, 2018*

Re: Pentaho Server 8 CE: User/Role List could not be obtained. Sep 13, 2018 3:30 PM

 Joao Ciocca *15 posts since Dec 20, 2017*

Re: Pentaho Server 8 CE: User/Role List could not be obtained. Sep 13, 2018 5:42 PM

sorry mate, we were thinking of integrating auth with our company's SSO mechanism, but so far, no progress - so we've just changed the default passwords for those default users and kept life going.

 Elijah Abdulrahman *3 posts since Jan 14, 2019*

Re: Pentaho Server 8 CE: User/Role List could not be obtained. Jan 15, 2019 5:50 PM

What did you edit to change the passwords? I see the script files in the data/hsqldb which have hashed passwords. Did you change these?

Thank you.

Gian Paolo Perillo *3 posts since Sep 26, 2018*

Re: Pentaho Server 8 CE: User/Role List could not be obtained. Sep 26, 2018 4:07 PM

Hi Joao, please could you specify files you have modified to switch back to HSQLDB?
We have just terminated Pentaho CE Server 8.1 installation but we have to stop our task due this issue.
Thanks in advance
Gian Paolo

Daniele Sibio *1 posts since Nov 5, 2018*

Re: Pentaho Server 8 CE: User/Role List could not be obtained. Nov 5, 2018 3:54 PM

Does Someone figured out how to resolve this issue?

I'm on RedHat/Tomcat/MySQL 5.7 pentaho 8.1 and i got the same error.

Cannot login on the Console with Admin or Suzy and the "User/Roles cannot be obtained .." in the logs file.
We also modified

SQL code

```
1
```

```
CREATE DATABASE IF NOT EXISTS `hibernate`
2 DEFAULT CHARACTER SET latin1;
3
4 USE hibernate;
GRANT ALL ON hibernate.* TO
'hibuser'@'localhost' identified by 'pass-
word';
commit;
```

Next we create **quartz** database and the user **pentaho_user**. You only need to execute the SQL script **create_quartz_mysql.sql** included at biserver-ce\data\mysql5

```
1
```

```
CREATE DATABASE IF NOT EXISTS `quartz`
2 DEFAULT CHARACTER SET latin1;
3
4
5 grant all on quartz.* to
6 'pentaho_user'@'localhost' identified by
7 'password';
8
```

```
USE `quartz`;

DROP TABLE IF EXISTS QRTZ5_JOB_LISTENERS;
DROP TABLE IF EXISTS QRTZ5_TRIGGER_LISTENERS;
DROP TABLE IF EXISTS QRTZ5_FIRED_TRIGGERS;
DROP TABLE IF EXISTS
QRTZ5_PAUSED_TRIGGER_GRPS;
DROP TABLE IF EXISTS QRTZ5_SCHEDULER_STATE;
DROP TABLE IF EXISTS QRTZ5_LOCKS;
DROP TABLE IF EXISTS QRTZ5_SIMPLE_TRIGGERS;
DROP TABLE IF EXISTS QRTZ5_CRON_TRIGGERS;
DROP TABLE IF EXISTS QRTZ5_BLOB_TRIGGERS;
DROP TABLE IF EXISTS QRTZ5_TRIGGERS;
DROP TABLE IF EXISTS QRTZ5_JOB_DETAILS;
DROP TABLE IF EXISTS QRTZ5_CALENDARS;

CREATE TABLE QRTZ5_JOB_DETAILS
  (
    JOB_NAME  VARCHAR(200) NOT NULL,
    JOB_GROUP VARCHAR(200) NOT NULL,
    DESCRIPTION VARCHAR(250) NULL,
    JOB_CLASS_NAME   VARCHAR(250) NOT NULL,
    IS_DURABLE VARCHAR(1) NOT NULL,
    IS_VOLATILE VARCHAR(1) NOT NULL,
    IS_STATEFUL VARCHAR(1) NOT NULL,
    REQUESTS_RECOVERY VARCHAR(1) NOT NULL,
    JOB_DATA BLOB NULL,
    PRIMARY KEY (JOB_NAME,JOB_GROUP)
);

CREATE TABLE QRTZ5_JOB_LISTENERS
  (
    JOB_NAME  VARCHAR(200) NOT NULL,
    JOB_GROUP VARCHAR(200) NOT NULL,
    JOB_LISTENER VARCHAR(200) NOT NULL,
    PRIMARY KEY
(JOB_NAME,JOB_GROUP,JOB_LISTENER),
    FOREIGN KEY (JOB_NAME,JOB_GROUP)
        REFERENCES
QRTZ5_JOB_DETAILS(JOB_NAME,JOB_GROUP)
);

CREATE TABLE QRTZ5_TRIGGERS
```

```
52   (
53     TRIGGER_NAME VARCHAR(200) NOT NULL,
54     TRIGGER_GROUP VARCHAR(200) NOT NULL,
55     JOB_NAME  VARCHAR(200) NOT NULL,
56     JOB_GROUP VARCHAR(200) NOT NULL,
57     IS_VOLATILE VARCHAR(1) NOT NULL,
58     DESCRIPTION VARCHAR(250) NULL,
59     NEXT_FIRE_TIME BIGINT(13) NULL,
60     PREV_FIRE_TIME BIGINT(13) NULL,
61     PRIORITY INTEGER NULL,
62     TRIGGER_STATE VARCHAR(16) NOT NULL,
63     TRIGGER_TYPE VARCHAR(8) NOT NULL,
64     START_TIME BIGINT(13) NOT NULL,
65     END_TIME BIGINT(13) NULL,
66     CALENDAR_NAME VARCHAR(200) NULL,
67     MISFIRE_INSTR SMALLINT(2) NULL,
68     JOB_DATA BLOB NULL,
69     PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP),
70     FOREIGN KEY (JOB_NAME,JOB_GROUP)
71         REFERENCES
72 QRTZ5_JOB_DETAILS(JOB_NAME,JOB_GROUP)
73 );
74
75 CREATE TABLE QRTZ5_SIMPLE_TRIGGERS
76   (
77     TRIGGER_NAME VARCHAR(200) NOT NULL,
78     TRIGGER_GROUP VARCHAR(200) NOT NULL,
79     REPEAT_COUNT BIGINT(7) NOT NULL,
80     REPEAT_INTERVAL BIGINT(12) NOT NULL,
81     TIMES_TRIGGERED BIGINT(10) NOT NULL,
82     PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP),
83     FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP)
84         REFERENCES
85 QRTZ5_TRIGGERS(TRIGGER_NAME,TRIGGER_GROUP)
86 );
87
88 CREATE TABLE QRTZ5_CRON_TRIGGERS
89   (
90     TRIGGER_NAME VARCHAR(200) NOT NULL,
91     TRIGGER_GROUP VARCHAR(200) NOT NULL,
92     CRON_EXPRESSION VARCHAR(200) NOT NULL,
93     TIME_ZONE_ID VARCHAR(80),
94     PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP),
```

```
95      FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP)
96          REFERENCES
97  QRTZ5_TRIGGERS(TRIGGER_NAME,TRIGGER_GROUP)
98  );
99
100 CREATE TABLE QRTZ5_BLOB_TRIGGERS
101   (
102     TRIGGER_NAME VARCHAR(200) NOT NULL,
103     TRIGGER_GROUP VARCHAR(200) NOT NULL,
104     BLOB_DATA BLOB NULL,
105     PRIMARY KEY (TRIGGER_NAME,TRIGGER_GROUP),
106     FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP)
107         REFERENCES
108 QRTZ5_TRIGGERS(TRIGGER_NAME,TRIGGER_GROUP)
109 );
110
111 CREATE TABLE QRTZ5_TRIGGER_LISTENERS
112   (
113     TRIGGER_NAME  VARCHAR(200) NOT NULL,
114     TRIGGER_GROUP VARCHAR(200) NOT NULL,
115     TRIGGER_LISTENER VARCHAR(200) NOT NULL,
116     PRIMARY KEY
117 (TRIGGER_NAME,TRIGGER_GROUP,TRIGGER_LISTENER),
118     FOREIGN KEY (TRIGGER_NAME,TRIGGER_GROUP)
119         REFERENCES
120 QRTZ5_TRIGGERS(TRIGGER_NAME,TRIGGER_GROUP)
121 );
122
123 CREATE TABLE QRTZ5_CALENDARS
124   (
125     CALENDAR_NAME  VARCHAR(200) NOT NULL,
126     CALENDAR BLOB NOT NULL,
127     PRIMARY KEY (CALENDAR_NAME)
128 );
129
130 CREATE TABLE QRTZ5_PAUSED_TRIGGER_GRPS
131   (
132     TRIGGER_GROUP  VARCHAR(200) NOT NULL,
133     PRIMARY KEY (TRIGGER_GROUP)
134 );
135
136 CREATE TABLE QRTZ5_FIRED_TRIGGERS
137   (
```

```
    ENTRY_ID VARCHAR(95) NOT NULL,
    TRIGGER_NAME VARCHAR(200) NOT NULL,
    TRIGGER_GROUP VARCHAR(200) NOT NULL,
    IS_VOLATILE VARCHAR(1) NOT NULL,
    INSTANCE_NAME VARCHAR(200) NOT NULL,
    FIRED_TIME BIGINT(13) NOT NULL,
    PRIORITY INTEGER NOT NULL,
    STATE VARCHAR(16) NOT NULL,
    JOB_NAME VARCHAR(200) NULL,
    JOB_GROUP VARCHAR(200) NULL,
    IS_STATEFUL VARCHAR(1) NULL,
    REQUESTS_RECOVERY VARCHAR(1) NULL,
    PRIMARY KEY (ENTRY_ID)
);

CREATE TABLE QRTZ5_SCHEDULER_STATE
  (
    INSTANCE_NAME VARCHAR(200) NOT NULL,
    LAST_CHECKIN_TIME BIGINT(13) NOT NULL,
    CHECKIN_INTERVAL BIGINT(13) NOT NULL,
    PRIMARY KEY (INSTANCE_NAME)
);

CREATE TABLE QRTZ5_LOCKS
  (
    LOCK_NAME  VARCHAR(40) NOT NULL,
    PRIMARY KEY (LOCK_NAME)
);

INSERT INTO QRTZ5_LOCKS
values('TRIGGER_ACCESS');
INSERT INTO QRTZ5_LOCKS values('JOB_ACCESS');
INSERT INTO QRTZ5_LOCKS
values('CALENDAR_ACCESS');
INSERT INTO QRTZ5_LOCKS
values('STATE_ACCESS');
INSERT INTO QRTZ5_LOCKS
values('MISFIRE_ACCESS');
commit;
```

Finally we create **jackrabbit** database and the user **jcr_user**. You only need to execute the SQL script **create_jcr_mysql.sql** included at biserver-ce\data\mysql5

```
CREATE DATABASE IF NOT EXISTS `jackrabbit`
DEFAULT CHARACTER SET latin1;

grant all on jackrabbit.* to
'jcr_user'@'localhost' identified by 'pass-
word';
commit;
```

Configuring JDBC Security

This section describes how to configure the Pentaho BI Platform JDBC security to use a MySQL server, this means the Pentaho BI Platform will now point to the hibernate database on the MySQL server instead of the packaged HSQL in memory database.

1. applicationContext-spring-security-hibernate.properties.

Edit the file pentaho-solutions\system**applicationContext-spring-security-hibernate.properties**.

Original code

```
jdbc.driver=org.hsqldb.jdbcDriver

jdbc.url=jdbc:hsqldb:hsql://local-
host:9001/hibernate
jdbc.username=hibuser
jdbc.password=password
hibernate.dialect=org.hibernate.dialect.HSQLDialect
```

Make the changes necessary to get the snippet of code below

```
jdbc.driver=com.mysql.jdbc.Driver

jdbc.url=jdbc:mysql://localhost:3306/hiber-
nate
jdbc.username=hibuser
jdbc.password=password
hibernate.dialect=org.hibernate.dialect.MySQLDialect
```

2. hibernate-settings.xml

Edit the file pentaho-solutions\system\hibernate**hibernate-settings.xml**.

Original code

```
<config-file>system/hiberna-
te/hsql.hibernate.cfg.xml</config-file>
```

Make the changes necessary to get the snippet of code below

```
<config-file>system/hiberna-
te/mysql5.hibernate.cfg.xml</config-file>
```

3. mysql5.hibernate.cfg.xml

Edit the file pentaho-solutions\system\hibernate**mysql5.hibernate.cfg.xml** .
You do not need to make any changes to this file if you would like to use the default user hibuser. However, if you would like to specify your custom user, change connection.username and password properties.
Original code

```
<property
name="connection.driver_class">com.mysql.jdbc.Driver</
property>

<property
name="connection.url">jdbc:mysql://local-
host:3306/hibernate</property>
<property
name="dialect">org.hibernate.dialect.MySQL5InnoDBDialect
property>
<property
name="connection.username">hibuser</proper-
ty>
<property
name="connection.password">password</proper-
ty>
```

4. quartz.properties

Edit the file pentaho-solutions\system\quartz**quartz.properties** .
Original code

```
org.quartz.jobStore.driverDelegateClass =
org.quartz.impl.jdbcjobstore.PostgreSQLDelegate
```

Make the changes necessary to get the snippet of code below

```
org.quartz.jobStore.driverDelegateClass =
org.quartz.impl.jdbcjobstore.StdJDBCDelegate
```

Configuring Hibernate and Quartz

Hibernate and Quartz need to specifically use the hibernate and quartz databases which were created on the MySQL server. To do so modifications need to be executed in context.xml file .

5. context.xml

Edit the file tomcat\webapps\pentaho\META-INF**context.xml**.

Remember deleting tomcat\conf\Catalina\localhost**pentaho.xml** , Pentaho creates on startup pentaho.xml as a copy of **context.xml**.

Original code

```
<Resource name="jdbc/Hibernate"
auth="Container" type="javax.sql.DataSource"

factory="org.apache.commons.dbcp.BasicDataSourceFactory"
maxActive="20" maxIdle="5"
maxWait="10000" username="hibuser"
password="password"
driverClassName="org.hsqldb.jdbcDriver"
url="jdbc:hsqldb:hsql://localhost/hibernate"
validationQuery="select count(*) from
INFORMATION_SCHEMA.SYSTEM_SEQUENCES" />

<Resource name="jdbc/Quartz" auth="Container"
type="javax.sql.DataSource"
factory="org.apache.commons.dbcp.BasicDataSourceFactory"
maxActive="20" maxIdle="5"
maxWait="10000" username="pentaho_user"
password="password"
driverClassName="org.hsqldb.jdbcDriver"
url="jdbc:hsqldb:hsql://localhost/quartz"
validationQuery="select count(*) from
INFORMATION_SCHEMA.SYSTEM_SEQUENCES"/>
```

Make the changes necessary to get the snippet of code below

```
<Resource name="jdbc/Hibernate"
auth="Container" type="javax.sql.DataSource"

factory="org.apache.commons.dbcp.BasicDataSourceFactory"
maxActive="20" maxIdle="5"
maxWait="10000" username="hibuser"
password="password"

```

```
driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost:3306/hibernate"
validationQuery="select 1" />

<Resource name="jdbc/Quartz" auth="Container"
type="javax.sql.DataSource"
factory="org.apache.commons.dbcp.BasicDataSourceFactory"
maxActive="20" maxIdle="5"
maxWait="10000" username="pentaho_user"
password="password"
driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost:3306/quartz"
validationQuery="select 1"/>
```

6. repository.xml

Comment the original code in the FileSystem part

```
<!--

Replace the following "FileSystem" XML node
to use supported databases as
the repository file system.  Change the url,
user, password and other parameters
to suit your db installation.  The schemaOb-
jectPrefix should
be a unique prefix that will be prepended to
the table names.
NOTE: The database must be pre-created in
and match the parameters.  See Jackrabbit
documentation for further explanation.
-->
<FileSystem>
<param name="path" value="${rep.home}/reposi-
tory"/>
</FileSystem>
```

Make this code active on FileSystem part of the code

```
<FileSystem>

```

```
<param name="driver"
value="com.mysql.jdbc.Driver"/>
<param name="url" value="jdbc:mysql://local-
host:3306/jackrabbit"/>
<param name="user" value="jcr_user"/>
<param name="password" value="password"/>
<param name="schema" value="mysql"/>
<param name="schemaObjectPrefix"
value="fs_repos_"/>
</FileSystem>
```

Comment the original code in the DataStore part

```
<!--

data store configuration
-->
<!--
Replace the following "DataStore" XML node
to use supported databases as the data
store for the repository.  Change the url,
user, password and other parameters
to suit your db installation.  The schemaOb-
jectPrefix should
be a unique prefix that will be prepended to
the table names.
NOTE: The database must be pre-created in
and match the parameters.  See Jackrabbit
documentation for further explanation.
-->
<DataStore
class="org.apache.jackrabbit.core.data.FileDataStore"/
>
```

Make this code active on DataStore part of the code

```
<DataStore
class="org.apache.jackrabbit.core.data.db.DbDataStore">

    <param name="url" value="jdbc:mysql://lo-
calhost:3306/jackrabbit"/>
    <param name="user" value="jcr_user"/>
    <param name="password" value="password"/>
```

```
    <param name="databaseType" value="mysql"/>
    <param name="driver"
value="com.mysql.jdbc.Driver"/>
    <param name="minRecordLength" va-
lue="1024"/>
    <param name="maxConnections" value="3"/>
    <param name="copyWhenReading"
value="true"/>
    <param name="tablePrefix" value=""/>
    <param name="schemaObjectPrefix"
value="ds_repos_"/>
</DataStore>
```

Below the security part comment the original code in the FileSystem Workspace part

```
<!--

virtual file system of the workspace:
class: FQN of class implementing the File-
System interface
-->
<!--
Replace the following "FileSystem" XML node
to use supported databases as
the repository file system.  Change the url,
user, password and other parameters
to suit your db installation.  The schemaOb-
jectPrefix should
be a unique prefix that will be prepended to
the table names.
NOTE: The database must be pre-created in
and match the parameters.  See Jackrabbit
documentation for further explanation.
-->
<FileSystem>
<param name="path" value="${wsp.home}"/>
</FileSystem>
```

Make this code active on FileSystem WorkSpace part of the code

```
<FileSystem>

```

```
<param name="driver"
value="com.mysql.jdbc.Driver"/>
<param name="url" value="jdbc:mysql://local-
host:3306/jackrabbit"/>
<param name="user" value="jcr_user"/>
<param name="password" value="password"/>
<param name="schema" value="mysql"/>
<param name="schemaObjectPrefix"
value="fs_ws_"/>
</FileSystem>
```

Below FileSystem Workspace part you will find the PersistenceManager part

```
<!--

persistence manager of the workspace:
class: FQN of class implementing the Persis-
tenceManager interface
-->
<!--
Replace the following "PersistenceManager"
XML node to use a supported database as the
persistenanceManager store.  Change the url,
user, password and parameters
to suit your db installation.  The schemaOb-
jectPrefix should
be a unique prefix that will be prepended to
the table names.
NOTE: The database must be pre-created in
and match the parameters.  See Jackrabbit
documentation for further explanation.
-->

<PersistenceManager>
<param name="url" value="jdbc:h2:${wsp.home}/
db"/>
<param name="schemaObjectPrefix"
value="${wsp.name}_"/>
</PersistenceManager>
```

Make this code active on PersistenceManager part of the code

```
```

```
<PersistenceManager>

<param name="url" value="jdbc:mysql://local-
host:3306/jackrabbit"/>
<param name="user" value="jcr_user" />
<param name="password" value="password" />
<param name="schema" value="mysql"/>
<param name="schemaObjectPrefix"
value="${wsp.name}_pm_ws_"/>
</PersistenceManager>
```

Below you will find FileSystem Versioning part

```
1
```

```
<!--

Configures the filesystem to use for versio-
ning for the respective
persistence manager
-->
<!--
Replace the following "FileSystem" XML node
to use a supported database as
the repository file system.  Change the url,
user, password and other parameters
to suit your db installation.  The schemaOb-
jectPrefix should
be a unique prefix that will be prepended to
the table names.
NOTE: The database must be pre-created in
and match the parameters.  See Jackrabbit
documentation for further explanation.
-->
<FileSystem>
<param name="path" value="${rep.home}/versi-
on" />
</FileSystem>
```

Make this code active on FileSystem Versioning part

```
1
```

```
<FileSystem>

<param name="driver"
value="com.mysql.jdbc.Driver"/>
```

```
<param name="url" value="jdbc:mysql://local-
host:3306/jackrabbit"/>
<param name="user" value="jcr_user"/>
<param name="password" value="password"/>
<param name="schema" value="mysql"/>
<param name="schemaObjectPrefix"
value="fs_ver_"/>
</FileSystem>
```

Below you will find PersistenceManager Versioning part

1

```
<!--

Configures the persistence manager to be
used for persisting version state.
Please note that the current versioning im-
plementation is based on
a 'normal' persistence manager, but this
could change in future
implementations.
-->
<!--
Replace the following "PersistenceManager"
XML node to use a supported database as the
persistenanceManager store.  Change the url,
user, password and parameters
to suit your db installation.  The schemaOb-
jectPrefix should
be a unique prefix that will be prepended to
the table names.
NOTE: The database must be pre-created in
and match the parameters.  See Jackrabbit
documentation for further explanation.
-->
<PersistenceManager>
<param name="url" value="jdbc:h2:${rep.home}/
version/db"/>
<param name="schemaObjectPrefix"
value="version_"/>
</PersistenceManager>
```

Make this code active on PersistenceManager Versioning part

```
<PersistenceManager>

<param name="url" value="jdbc:mysql://local-
host:3306/jackrabbit"/>
<param name="user" value="jcr_user" />
<param name="password" value="password" />
<param name="schema" value="mysql"/>
<param name="schemaObjectPrefix"
value="pm_ver_"/>
</PersistenceManager>
```

Quit HSQL Hypersonic automatic startup

By default Hypersonic database starts up automatically – to avoid this comment or delete locate the following snippets of code from **web.xml**:

```
<!-- [BEGIN HSQLDB DATABASES] -->

<context-param>
<param-name>hsqldb-databases</param-name>
<param-value>sampledata@../../data/hs-
qldb/sampledata,hibernate@../../data/hs-
qldb/hibernate,quartz@../../data/hs-
qldb/quartz</param-value>
</context-param>
<!-- [END HSQLDB DATABASES] -->
```

Second section you need to comment or eliminate

```
<!-- [BEGIN HSQLDB STARTER] -->

<listener>
<liste-
ner-class>org.pentaho.platform.web.http.context.HsqldbSt
listener-class>
</listener>
<!-- [END HSQLDB STARTER] -->
```

Took the guide online

Marcus Berglund *1 posts since Jan 2, 2019*

Re: Pentaho Server 8 CE: User/Role List could not be obtained. Jan 2, 2019 12:35 PM

As far as I can tell, no one has managed to resolve this issue (?)
The only proposed workaround in this thread (including the accepted answer) is to use HSQLDB instead.

Does this mean that Pentaho PUC (CE) no longer can be said to work with Postgresql/MySQL ?
If so - is there an official statement to this fact ?

... Or if Pentaho PUC / Pentaho Server 8.1 is supposed to work with MySQL or Postgresql - perhaps I can raise a bug somewhere.
I'm a bit new around here so advice on where to report a bug would be most welcome !

EUGENIO SA *1 posts since Jun 19, 2019*

Re: Pentaho Server 8 CE: User/Role List could not be obtained. Jun 28, 2019 10:45 PM

Hello,
The steps below had worked for me:

1. Unzip the installation file (archive type includes Tomcat) to a fresh installation. Follow the steps in online guide. DO NOT start-pentaho.bat yet.
2. In step 4 of online guide, "Modify Jackrabbit Repository Information for MySQL", uncomment MySQL lines AND comment related to local database. It will force jackrabbit to populate MySQL instead of local HSQLDB in pentaho's first execution.
3. In file ".../webapps/pentaho/WEB-INF/web.xml", comment lines related to [BEGIN HSQLDB DATABASES] and [BEGIN HSQLDB STARTER].
4. Do start-pentaho.bat.

In my test, default users and roles were created in MySQL Jackrabbit. Of course, all depending of HSQLDB will be unavailable.
Best Regards,
Eugenio.