

Cloud Consulting Services

Payroll On Cloud

A document for setting the configuration on cloud



I. Maven dependencies added

At pom.xml

```
<dependency>
  <groupId>com.sap.cloud.db.jdbc</groupId>
  <artifactId>ngdbc</artifactId>
  <version>2.4.76</version>
</dependency>

</dependencies>
```

Link: SAP HANA JDBC Driver

Provides connectivity to HANA Databases in Java applications.

II. Spring and hibernate configurations to generate DDL scripts and Tables using entity classes

Hbm2ddl property

- *validate*: validate the schema, makes no changes to the database.
- *update*: update the schema.
- *create*: creates the schema, destroying previous data.
- *create-drop*: drop the schema when the SessionFactory is closed explicitly, typically when the application is stopped.
- *none*: does nothing with the schema, makes no changes to the database

At springConfig.xml

```
class="com.mchange.v2.c3p0.ComboPooledDataSource"
destroy-method="close">
<property name="driverClass" value="com.sap.cloud.db.jdbc" />
<property name="jdbcUrl" value="jdbc:sap://vadbirn.nwtrial.od.sap.biz:30152/?autocommit=false"/>
<property name="user" value="PAYROLLUSER" />
<property name="password" value="PayrollUser123" />

<!-- these are connection pool properties for C3P0 -->
<property name="minPoolSize" value="5" />
<property name="maxPoolSize" value="20" />
<property name="maxIdleTime" value="30000" />
<property name="initialPoolSize" value="5" />

</bean>

<!-- Step 2: Setup Hibernate session factory -->
<bean id="sessionFactory"
class="org.springframework.orm.hibernate5.LocalSessionFactoryBean">
<property name="dataSource" ref="myDataSource" />
<property name="packagesToScan"
value="com.entities" />
<property name="hibernateProperties">
<props>
<prop key="hibernate.dialect">org.hibernate.dialect.SAPDBDialect</prop>
<prop key="hibernate.show_sql">true</prop>
<prop key="hibernate.hbm2ddl.auto">create-drop</prop>

</props>
</property>
</bean>
```

NB: use the SAPDBDialect to create the tables on SAP HANA DB

NB: The below points updated At Sunday 27 September 2020

Configure Eclipse with SAP Cloud Platform Steps

Configure Eclipse with SAP Cloud Platform Tools for Java

You will find all steps with screenshots at the link below:

<https://developers.sap.com/tutorials/hcp-java-eclipse-setup.html>

Creating a trial account on Neo and How to connect Eclipse with it Steps

From the previous link, you will be connected to the neo environment with eclipse.

go into Eclipse's Window > Preferences, then option Server > SAP HANA Cloud and set the SDK location to a version you need. Since Java 8 is the standard now, it might be a good idea to use Tomcat 8, so that the runtime is Java 8 compatible.

The link below will help you to deploy on the Cloud from Eclipse using your trial account:

<https://help.sap.com/viewer/ea72206b834e4ace9cd834feed6c0e09/CLOUD/en-US/60ab35d9edde43a1b38cf48174a3dca2.html>

Creating database schema on the SAP Cloud Platform

The link below will help you

<https://help.sap.com/viewer/d4790b2de2f4429db6f3dff54e4d7b3a/Cloud/en-US/533384eda57e428f98a43815e6a11119.html#loio46af2934d19343ca8250ce288d27ea41>

This video will help you to clone the payroll app, deploy it, and run it on SAP Cockpit with database:

<https://drive.google.com/file/d/1YMU8ecA9fgeGXw9hJ3mzZ9Bq17SpHm6P/view?usp=sharing>

Creating System User and Development user on the Development workbench and its required roles

The video above shows the (PAYROLLUSER) user creation and the roles have given.

NB, The sequence in the database is the id generation on sap Hana schema

It solved inside the entity class with:

@GeneratedValue(generator="increment")

@GenericGenerator(name="increment", strategy = "increment")

Those annotations are written above the id in Entity class