**Rules:**

1. **Write a rule to display the given application in xml format.**

**Code:**  
<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE Rule PUBLIC "sailpoint.dtd" "sailpoint.dtd">

<Rule created="1611123637854" id="c0a8055e82ce1a088182d882c6b90704" language="beanshell" modified="1661490634998" name="Filter">

<Source>

import sailpoint.object.\*;

import sailpoint.object.Filter;

Filter filter = Filter.eq("name","Em");

filter = Filter.ignoreCase(filter);

Application app = context.getUniqueObject( Application.class, filter);

if(null != app)

{

return app;

}

</Source>

</Rule>

**Output:**

<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE Identity PUBLIC "sailpoint.dtd" "sailpoint.dtd">

<Identity correlated="true" created="1661430069651" id="c0a8055e82ce1a088182d4f2f9930675" lastRefresh="1661495939825" modified="1661495939826" name="John">

<Attributes>

<Map>

<entry key="displayName" value="John Doe"/>

<entry key="firstname" value="John"/>

<entry key="lastname" value="Doe"/>

</Map>

</Attributes>

<AttributeMetaDatas>

<AttributeMetaData attribute="firstname" source="Em:firstname"/>

<AttributeMetaData attribute="lastname" source="Em:lastname"/>

</AttributeMetaDatas>

<Links>

<Link created="1661430069652" displayName="John" id="c0a8055e82ce1a088182d4f2f9940676" lastRefresh="1661430069652" modified="1661430069655" identity="1">

<ApplicationRef>

<Reference class="sailpoint.object.Application" id="c0a8055e82ce1a088182d4f291b90669" name="Em"/>

</ApplicationRef>

<Attributes>

<Map>

<entry key="Active" value="Y"/>

<entry key="age" value="23"/>

<entry key="email" value="john.doe@company.com"/>

<entry key="firstname" value="John"/>

<entry key="id" value="1"/>

<entry key="lastname" value="Doe"/>

<entry key="location" value="India"/>

<entry key="userid" value="jdeo"/>

</Map>

</Attributes>

</Link>

</Links>

</Identity>

1. **Write a rule to display the given identity in xml format.**

**Code:**

<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE Rule PUBLIC "sailpoint.dtd" "sailpoint.dtd">

<Rule created="1611123637854" id="c0a8055e82ce1a088182d8cab4d107aa" language="beanshell" name="Filter2">

<Source>

import sailpoint.object.\*;

import sailpoint.object.Filter;

Filter filter = Filter.eq("firstname","Gracia");

Identity id = context.getUniqueObject( Identity.class, filter);

if(null != id)

{

return id;

}

</Source>

</Rule>

**Output:**

<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE Identity PUBLIC "sailpoint.dtd" "sailpoint.dtd">

<Identity correlated="true" created="1661430069730" id="c0a8055e82ce1a088182d4f2f9e2067d" lastRefresh="1661495939889" modified="1661495939890" name="Gracia">

<Attributes>

<Map>

<entry key="displayName" value="Gracia Ford"/>

<entry key="firstname" value="Gracia"/>

<entry key="lastname" value="Ford"/>

</Map>

</Attributes>

<AttributeMetaDatas>

<AttributeMetaData attribute="firstname" source="Em:firstname"/>

<AttributeMetaData attribute="lastname" source="Em:lastname"/>

</AttributeMetaDatas>

<Links>

<Link created="1661430069731" displayName="Gracia" id="c0a8055e82ce1a088182d4f2f9e3067e" lastRefresh="1661430069731" modified="1661430069731" identity="5">

<ApplicationRef>

<Reference class="sailpoint.object.Application" id="c0a8055e82ce1a088182d4f291b90669" name="Em"/>

</ApplicationRef>

<Attributes>

<Map>

<entry key="Active" value="Y"/>

<entry key="age" value="36"/>

<entry key="email" value="gracia.ford@home.com"/>

<entry key="firstname" value="Gracia"/>

<entry key="id" value="5"/>

<entry key="lastname" value="Ford"/>

<entry key="location" value="Canada"/>

<entry key="userid" value="gford"/>

</Map>

</Attributes>

</Link>

</Links>

</Identity>

1. **Write a rule to display the APP id and application name in csv file and also below code generates the csv file.**

**Code:**

<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE Rule PUBLIC "sailpoint.dtd" "sailpoint.dtd">

<Rule created="1611123637854" id="c0a8055e82ce1a088182d87176f706df" language="beanshell" name="ApplicationcsvRule">

<Source>

import sailpoint.object.Identity;

import sailpoint.object.Application;

import sailpoint.api.SailPointContext;

import java.util.ArrayList;

import java.util.List;

import java.io.\*;

List app = context.getObjects(Application.class);

List l = new ArrayList();

for(Application application : app) {

l.add(application.getId() + "," + application.getName());

}

String fileLocation = "D:\\Test1.csv";

  File f = new File(fileLocation);

BufferedWriter bfr = new BufferedWriter(new FileWriter(f) );

bfr.append("AppID");

bfr.append(",");

bfr.append("ApplicationName");

bfr.append(",");

bfr.append("\r\n");

try{

for(int i=0; i&lt;l.size();i++)

{

bfr.write(l.get(i)+","+"\n");

}

System.out.print("\r\n");

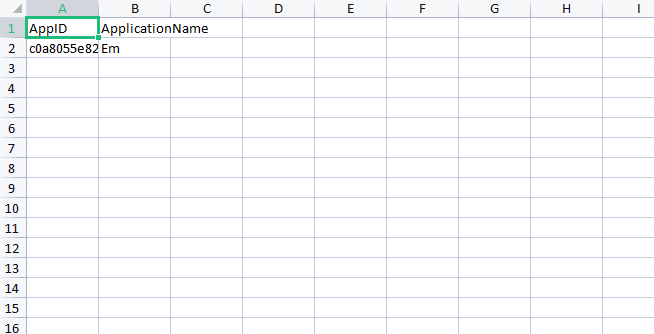
bfr.close() ; }

catch(Exception e){

}

</Source>

</Rule>

**Output:**  


**4)Write a rule to display the list of applications available in sailpoint iiq.**

**Code:**<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE Rule PUBLIC "sailpoint.dtd" "sailpoint.dtd">

<Rule created="1611123637854" id="c0a8055e821a19b381821a2bf8ee0040" language="beanshell" modified="1660041018443" name="ApplicationRule">

<Source>

import sailpoint.object.\*;

import sailpoint.object.Identity;

import sailpoint.object.Application;

import java.util.\*;

List list =context.getObjects(Application.class);

// return list;

List name = new ArrayList();

for(Application id : list){

name.add(id.getName());

}

return name;

</Source>

</Rule>

Output:  


**5)Write a rule to display the list of identities available in sail point iiq.**

**Code:**

<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE Rule PUBLIC "sailpoint.dtd" "sailpoint.dtd">

<Rule created="1611123637854" id="c0a8055e821a19b381821a2bf8ee0040" language="beanshell" modified="1661516515576" name="IdentityRule">

<Source>

import sailpoint.object.\*;

import sailpoint.object.Identity;

import java.util.\*;

List list =context.getObjects(Identity.class);

// return list;

List name = new ArrayList();

for(Identity id : list){

name.add(id.getName());

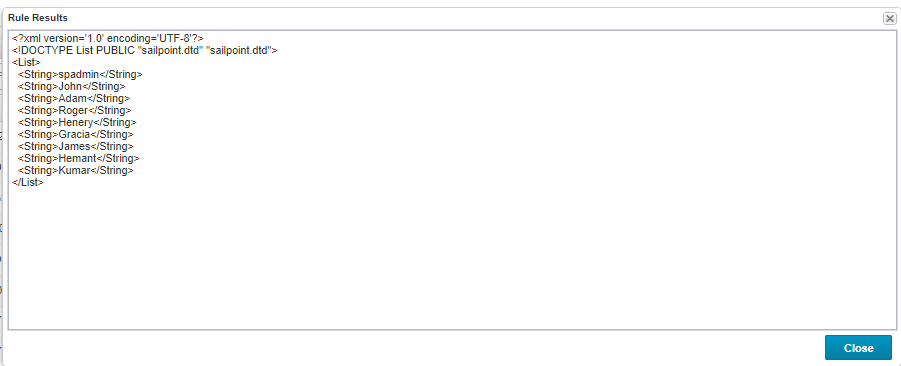
}

return name;

</Source>

</Rule>

**Output:**



**6)**