LAB- Using Database Endpoint

In this lab, you will be working on **01-databaseendpoint-start** project under **03-database-externalising-domain** section

In this lab you will understand

a) how to use a database endpoint to fire a select query

b) how to fire various type of queries like Parameterized queries, dynamic queries and template queries.

Before you proceed, please make sure that you have installed mysql database latest version and also any sql client. I would recommend to use SQLYOG as client.

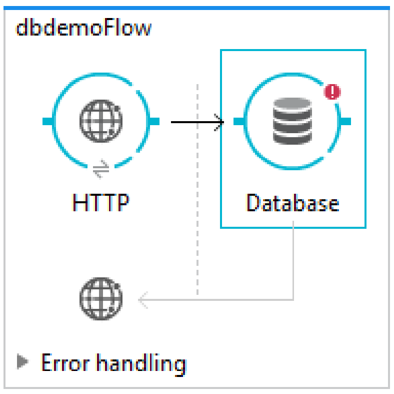
Make sure that you give password as root since all my examples use root/root as username/password

Execute **muletrainingdb.sql** to make sure that database, tables are created and data is populated.

STEP 1

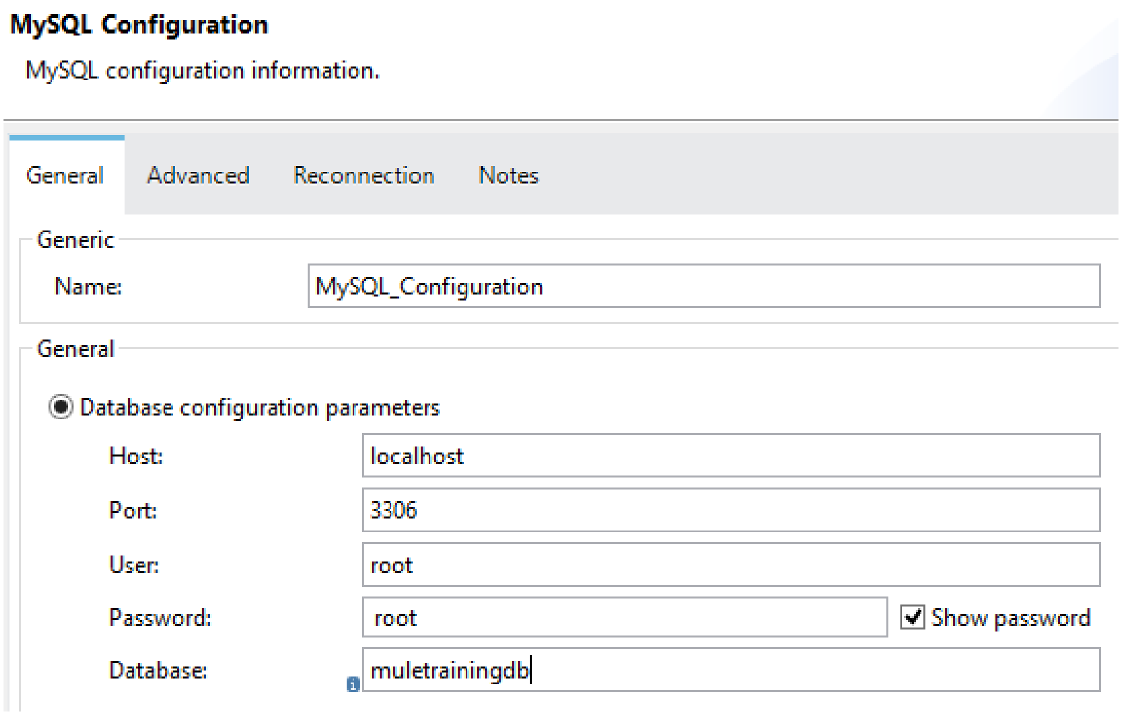
1) Right click on the project , select Build path->Add external archives and select mysql-connector jar.

2) Drag and drop a http and database endpoints as shown below.



Make sure that you configure Http Endpoint for http://localhost:8081/db

2) Create database connector configuration for mysql as shown below :



MySQL Configuration

MySQL configuration information.

Generic

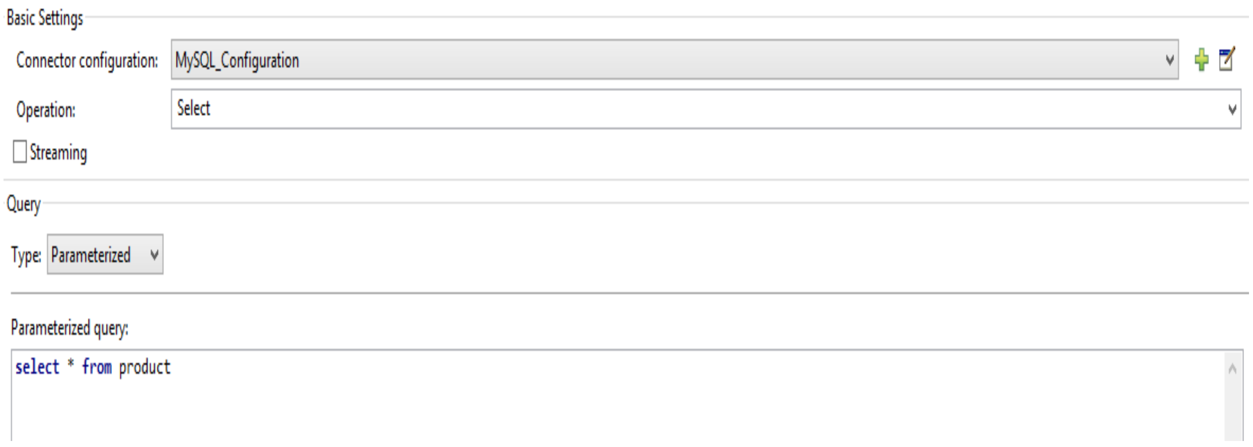
Name: MySQL_Configuration

General

● Database configuration parameters

Host:	localhost
Port:	3306
User:	root
Password:	root ☑ Show password
Database:	muletrainingdb

3) select the operation as "select" and configure query as below :



Basic Settings

Connector configuration: MySQL_Configuration

Operation: Select

☐ Streaming

Query

Type: Parameterized

Parameterized query:

```
select * from product
```

4) After firing select query, database endpoint return List<Map> . To show it on browser , we need to convert List to String.

So, add "Object to String" transformer after database endpoint.

5) Now deploy the application and give request to http://localhost:8081/db and Observe that List of products is displayed.

If we give request to http://localhost:8081/db?brandname=Apple , We want all the products whose brand_name is Apple.

Modify the query to use the query parameter brandname as show below :

```
select * from product where
brand_name=#[message.inboundProperties.'http.query.params'.brandname]
```

Now deploy and test.

6) if you have Observed the drop down, for Query type there are 2 types of query .

a) parameterized query

b) dynamic query

c) from Template

Till now you saw how to configure parameterized query. This will use pre compiled queries.

But if table names or column names are dynamic , we cannot use parameterized queries .

Now select the query type as "Dynamic" and modify the same query as show below :

```
select * from product where
brand_name='#[message.inboundProperties.'http.query.params'.brandname]'
```

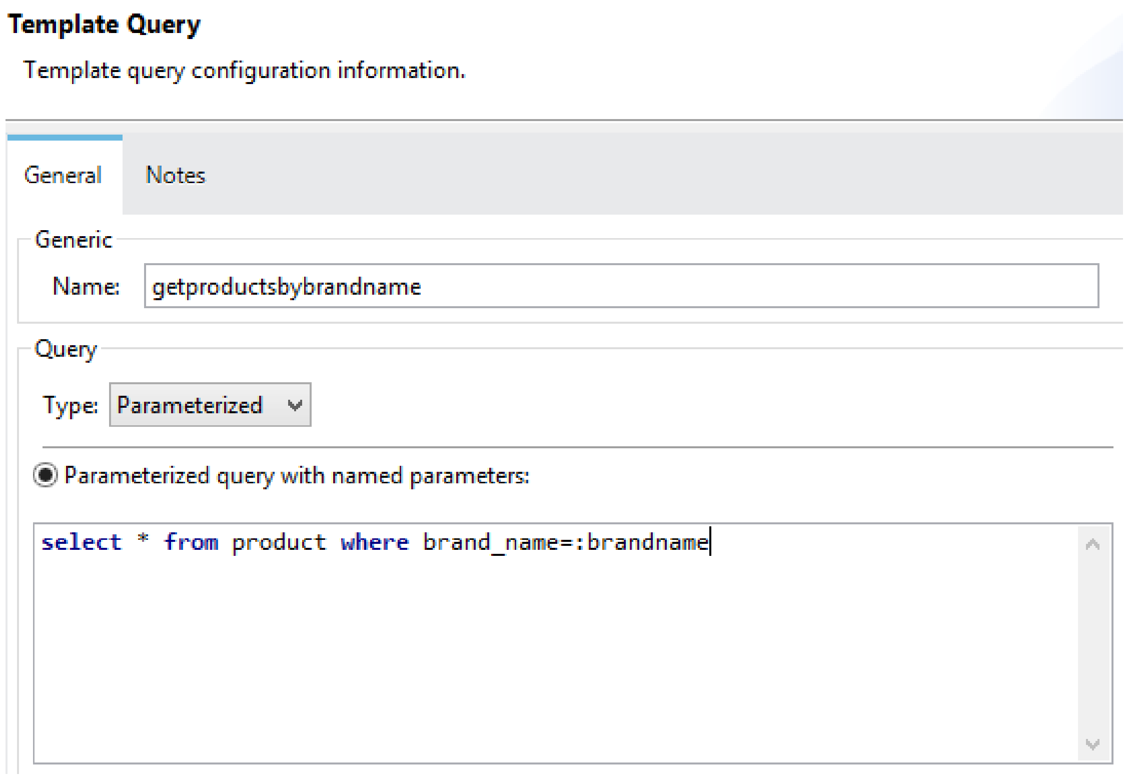
Observe that the expression is enclosed in single quote.

Now deploy and test.

7) Now change the query type as "from Template".

Create a new Template query with name "getproductsbybrandname".(Can you tell what is the advantage of template query over parameterized query ? See the video if you can't answer)

Give the query as shown below :



Template Query

Template query configuration information.

Generic

Name: getproductsbybrandname

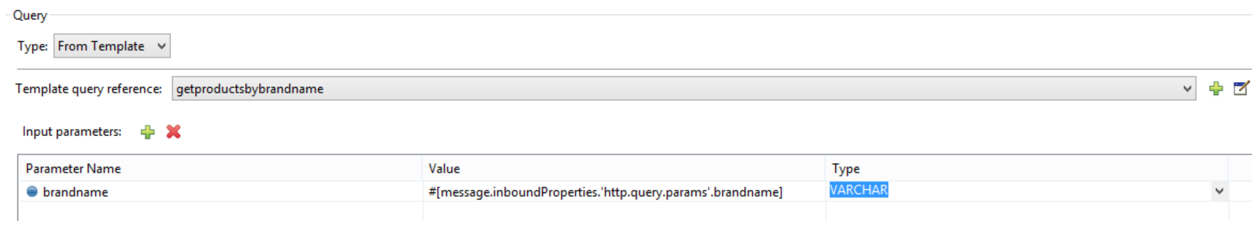
Query

Type: Parameterized

(●) Parameterized query with named parameters:

```
select * from product where brand_name=:brandname
```

8) Now give the parameter value for brand name as shown below :



Query

Type: From Template

Template query reference: getproductsbybrandname

Input parameters:

Parameter Name	Value	Type
brandname	#[message.inboundProperties.'http.query.params'.brandname]	VARCHAR

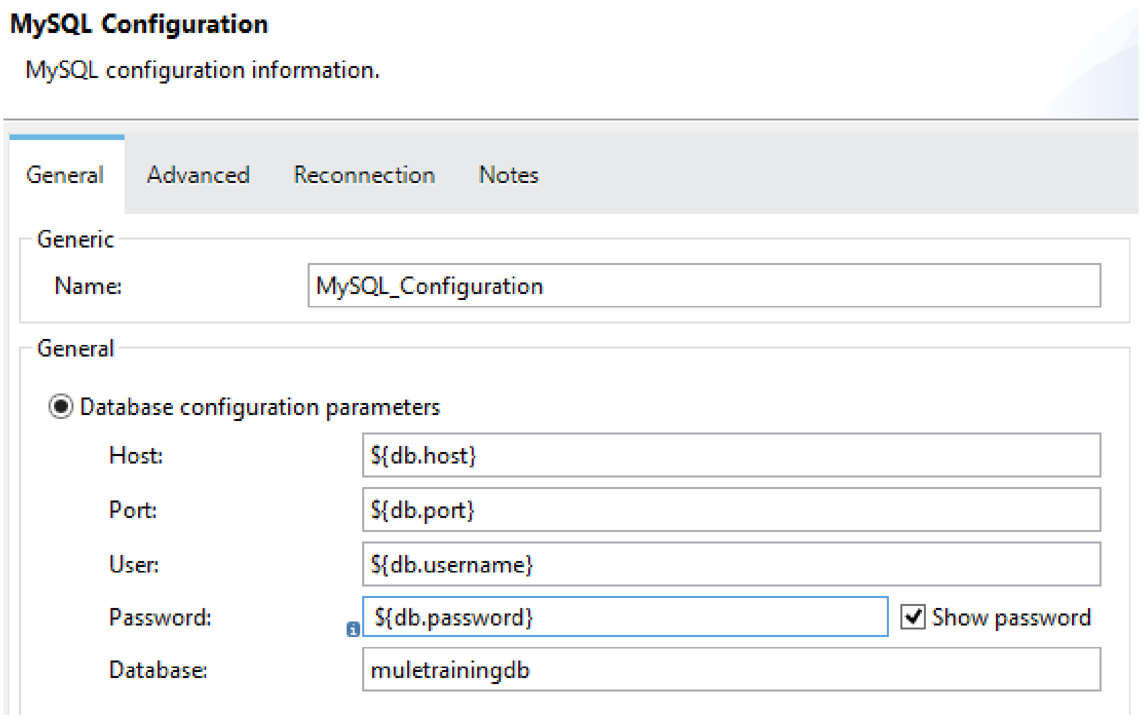
9) Deploy this application and test it by giving a request to

http://localhost:8081/db?brandname=Apple

STEP2

In this step, you will be working on **02-externalising-start** project

1) Edit the Database Connector configuration as shown below :



MySQL Configuration

MySQL configuration information.

Generic

Name: MySQL_Configuration

General

● Database configuration parameters

Host: ${db.host}

Port: ${db.port}

User: ${db.username}

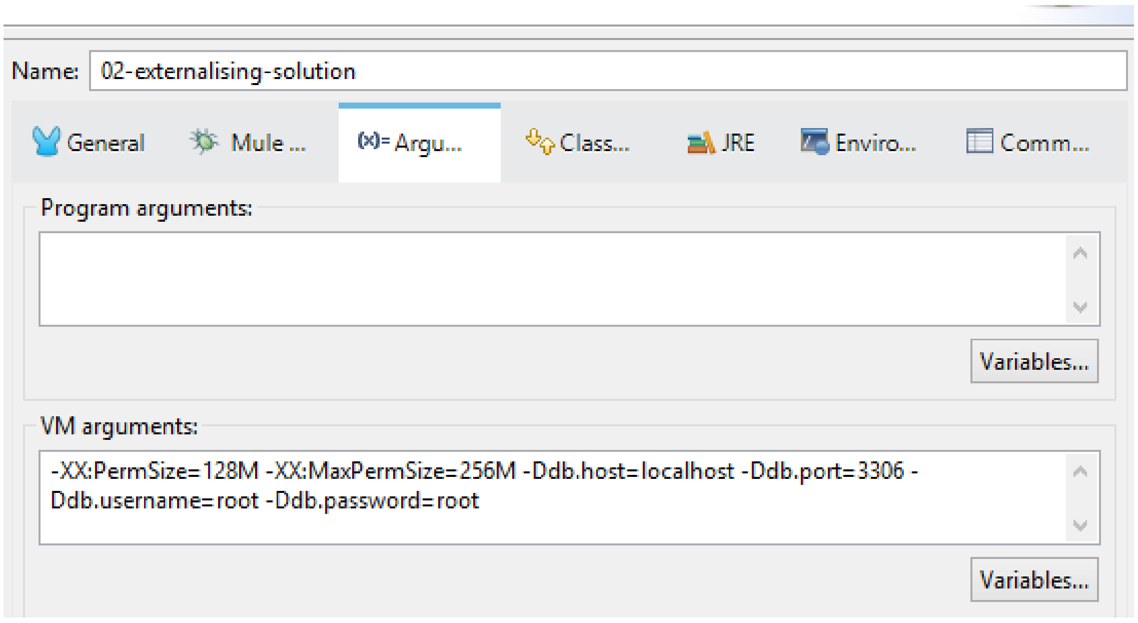
Password: ${db.password} ☑ Show password

Database: muletrainingdb

2) How to pass the values for place holders?

You can pass them as Java environment variables while running mule application.

Right click on project -->Run as -> Run Configurations.. and configure as below and run

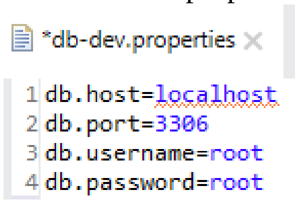

Name: 02-externalising-solution

```
-XX:PermSize=128M -XX:MaxPermSize=256M -Ddb.host=localhost -Ddb.port=3306 -
Ddb.username=root -Ddb.password=root
```

Give a request to http://localhost:8081/db?brandname=Apple and make sure that you get results

3)
Instead of passing multiple environment variables, we can externalize all the properties to a .properties files.

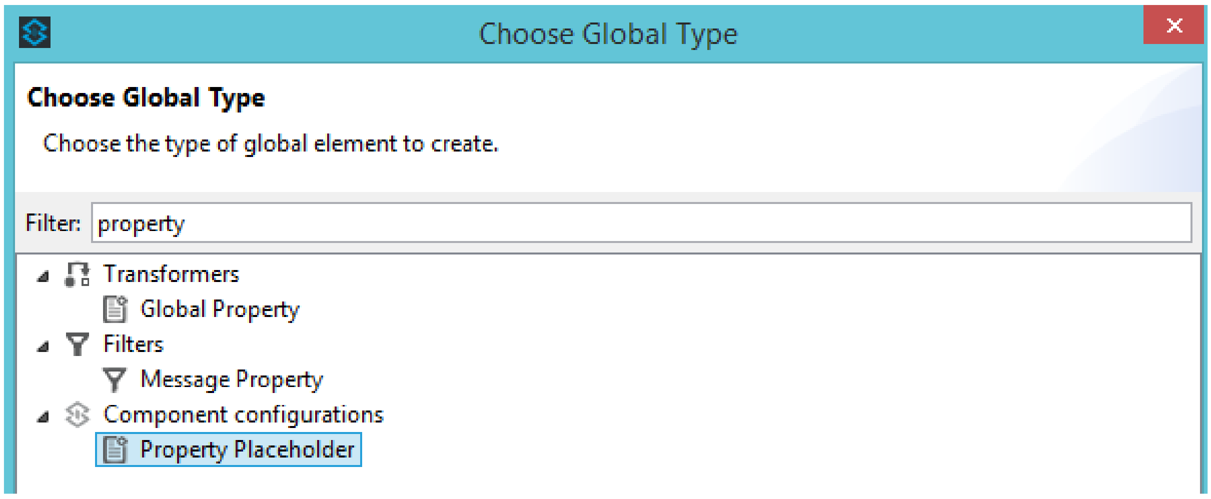
Create db-dev.properties as shown below :



```
db.host=localhost
db.port=3306
db.username=root
db.password=root
```

5)similarly create db-prod.properties with different values

4) Click on "Global Elements" and Click on Create button. Type Property and select "Property placeholder" as shown below :


Choose Global Type

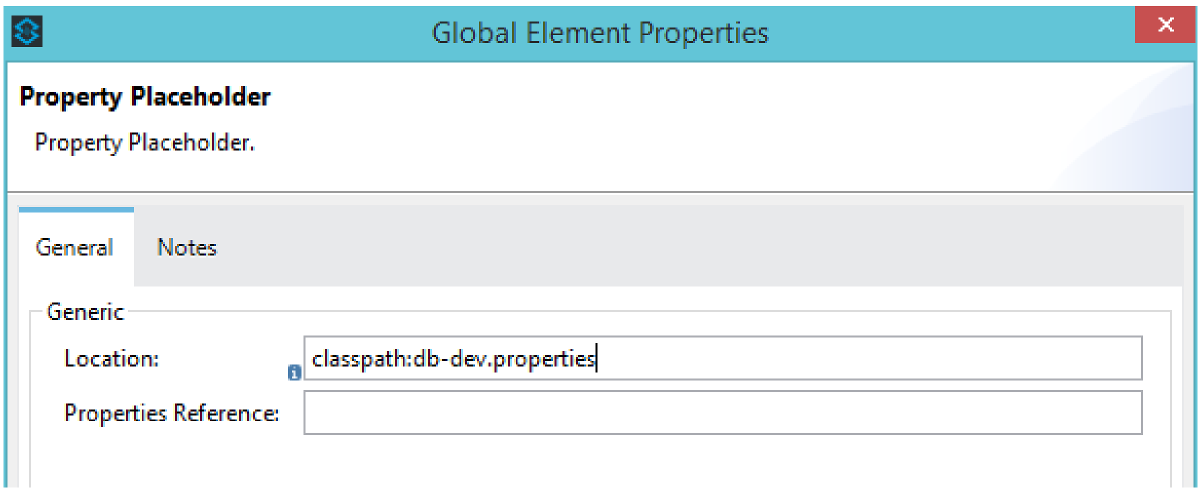
Choose Global Type

Choose the type of global element to create.

Filter: property

- Transformers
  - Global Property
- Filters
  - Message Property
- Component configurations
  - Property Placeholder

5) Configure Property place holder as shown below :


Global Element Properties

Property Placeholder

Property Placeholder.

Generic

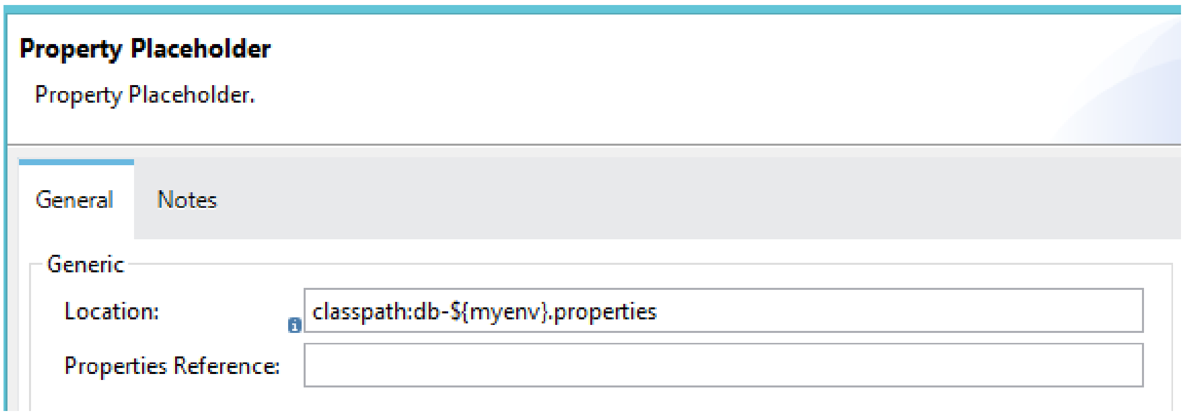
Location: classpath:db-dev.properties

Properties Reference:

WAY2LEARN

6) Now deploy the application and test by giving request to
http://localhost:8081/db?brandname=Apple

7) Now modify the property placeholder as below :


Property Placeholder

Property Placeholder.

General | Notes

Generic

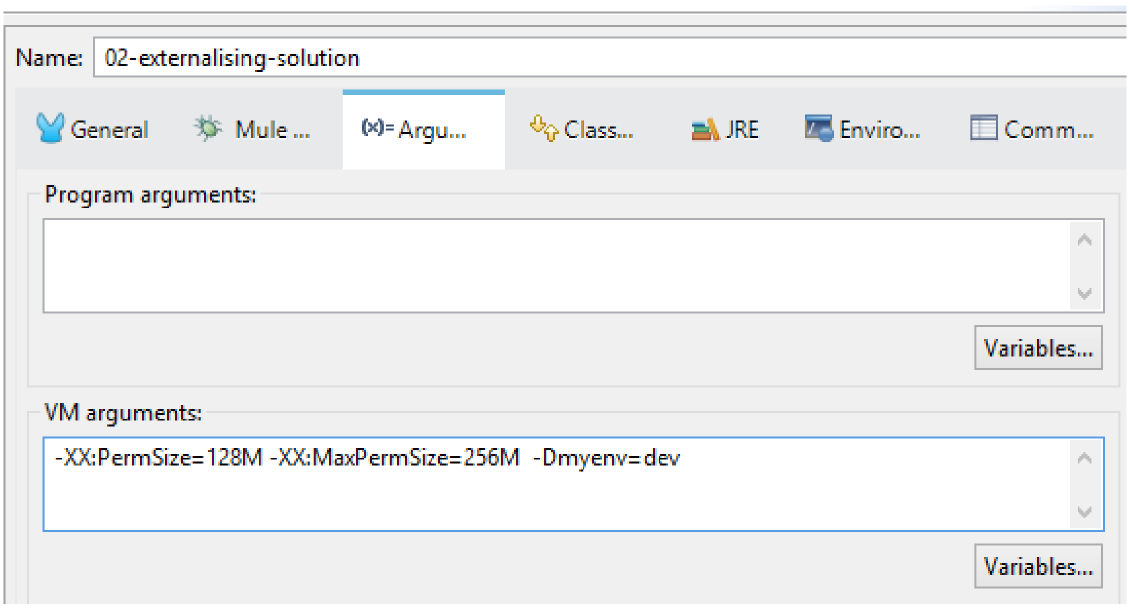
Location: classpath:db-${myenv}.properties

Properties Reference:

8)

You can pass the value of myenv as Java environment variables while running mule application.

Right click on project -->Run as -> Run Configurations.. and configure as below and run


Name: 02-externalising-solution

General | Mule ... | Argu... | Class... | JRE | Enviro... | Comm...

Program arguments:

Variables...

VM arguments:

```
-XX:PermSize=128M -XX:MaxPermSize=256M  -Dmyenv=dev
```

Variables...

WAY2LEARN

STEP3

In this step, you will understand how to share configurations among multiple mule applications using domains

1) Create a new Mule Domain Project with name " 03-way2learndomain-start"

Add the mysql driver jar into build path of the project.

2) Open " externalizing-usingdomain.xml" inside "03-databaseendpoint-using-domain-start" project .

Cut (I am saying cut. Not copy. Be careful) db:mysql-config and context:property-placeholder tags and paste them in **mule-domain-config.xml** inside 03-way2learndomain-start".

You will see that there are errors in this xml file in domain project because namespaces are missing in the xml.

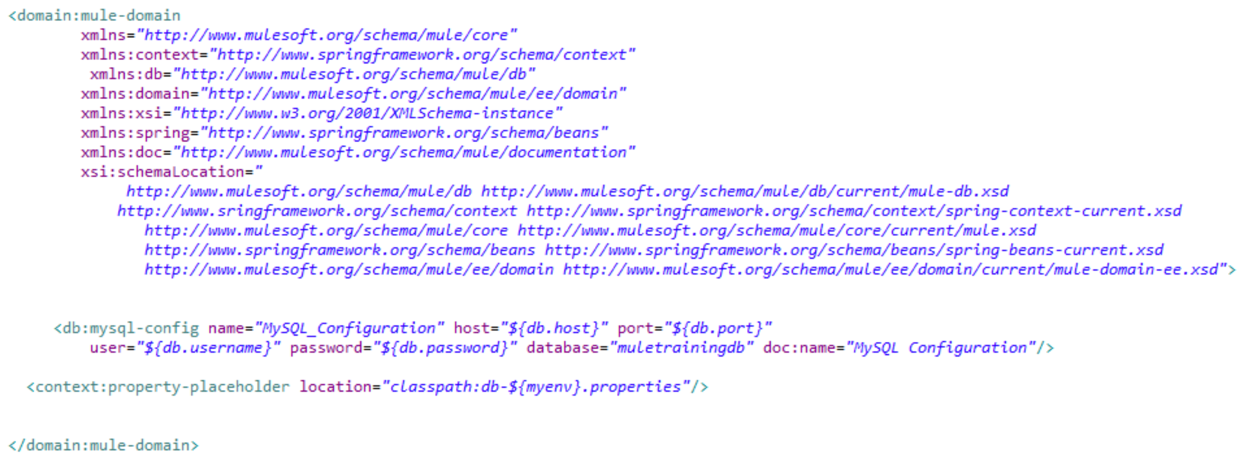
Copy the below lines in to the root tag of mule-domain-config.xml

```
xmlns:context="http://www.springframework.org/schema/context"

 xmlns:db="http://www.mulesoft.org/schema/mule/db"
```

Also copy the below schema locations

```
http://www.mulesoft.org/schema/mule/db
http://www.mulesoft.org/schema/mule/db/current/mule-db.xsd

http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-current.xsd
```

Make sure that the file finally looks like below :



```
<domain:mule-domain
        xmlns="http://www.mulesoft.org/schema/mule/core"
        xmlns:context="http://www.springframework.org/schema/context"
         xmlns:db="http://www.mulesoft.org/schema/mule/db"
        xmlns:domain="http://www.mulesoft.org/schema/mule/ee/domain"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:spring="http://www.springframework.org/schema/beans"
        xmlns:doc="http://www.mulesoft.org/schema/mule/documentation"
        xsi:schemaLocation="
             http://www.mulesoft.org/schema/mule/db http://www.mulesoft.org/schema/mule/db/current/mule-db.xsd
            http://www.sringframework.org/schema/context http://www.springframework.org/schema/context/spring-context-current.xsd
               http://www.mulesoft.org/schema/mule/core http://www.mulesoft.org/schema/mule/core/current/mule.xsd
               http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-current.xsd
               http://www.mulesoft.org/schema/mule/ee/domain http://www.mulesoft.org/schema/mule/ee/domain/current/mule-domain-ee.xsd">


     <db:mysql-config name="MySQL_Configuration" host="${db.host}" port="${db.port}"
         user="${db.username}" password="${db.password}" database="muletrainingdb" doc:name="MySQL Configuration"/>

   <context:property-placeholder location="classpath:db-${myenv}.properties"/>


</domain:mule-domain>
```

3) Now copy db-dev.properties and db-prod.properties from **"03-databaseendpoint-using-domain-start"** project to src/main/resources of **"03-way2learndomain-start" project.**

4) Now you will see an error in **"03-databaseendpoint-using-domain-start"** project becase you have cut "MySQL_configuration" tag from the xml and pasted in domain.

Now to clear that error, we need to configure **"03-databaseendpoint-using-domain-start"** project to belong to **"03-way2learndomain-start "** domain

Open mule-deploy.properties in **"03-databaseendpoint-using-domain-start"** project and observe that domain is "default".

change that domain to **03-way2learndomain-start.**

5) Now deploy **"03-databaseendpoint-using-domain-start" .** But dont forget to pass value of myenv as dev.

Test the application by making request to http://localhost:8081/db?brandname=Apple

This is the end of the Exercise