AWS:

For creating s3 bucket:

For integration with aws s3 bucket:

Watch Amazon s3 connector with mule esb jitendar bafna/ For integration with aws s3 bucket sivathankamane video

For creating new access key and secret key:

In console.aws.com

Click down arrow next to name

Choose security credentials

If u want delete/create new access key there

Create a new access key, which includes a new secret access key.

* To create a new secret access key for your root account, use the [security credentials page](https://console.aws.amazon.com/iam/home?#security_credential). Expand the **Access Keys** section, and then click **Create New Root Key**.
* To create a new secret access key for an IAM user, open the [IAM console](https://console.aws.amazon.com/iam/home?region=us-east-1#home). Click **Users** in the **Details** pane, click the appropriate IAM user, and then click **Create Access Key** on the **Security Credentials**tab.

Read/Downloadobj from s

Proj-ws -> integrate-aws-createfile

Mule domain project/domain application

It is used to define commonly used properties, and we can use/refer this domain by right clicking project -> properties-> Mule Project ->click drop down button and select mule domain.

We no need to create http config, secure properties, configuration properties everywhere. Just we can use this domain. We can’t use mule domain for commonly used flows, it is only used to store commonly used properties.

Only common global elements can be kept in domain project

No common flows is kept inside domain project

To create mule domain project,

Click new->mule domain project

It don’t have any modules inside, to add that in global elements click manage modules and login to anypoint platform account

1.Add http,database, secure properties connectors(add config.yaml in src/main/resources)

2.go to project where u wan to use this common properties

3. delete already added properties

4.right click project, select properties, select mule project, in that domain option drop down button, select the mule domain u created.

If want to use **Rest api** which I created already, I can use **HTTP Request** to use that by specifying its corresponding

Base path: /rest

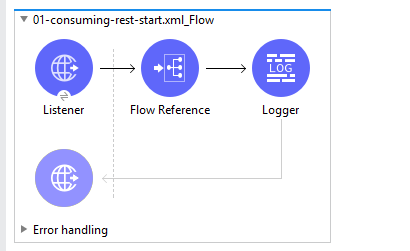
Endpoint: /products

Port:7070

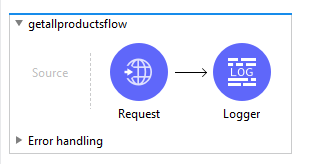
**In postman:**

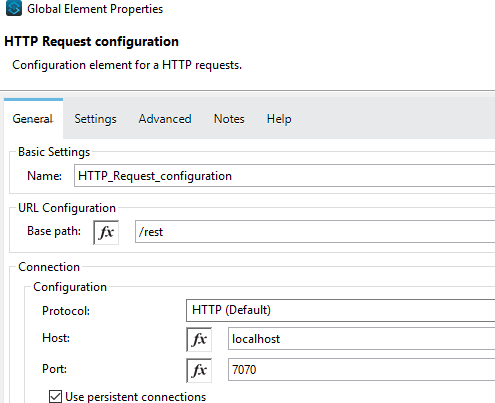
**If we use** [**http://localhost:707/rest/products**](http://localhost:707/rest/products)

**U can access that api resources**

****

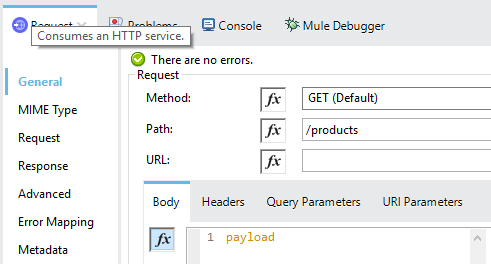
**Flow ref calls getall productsflow**

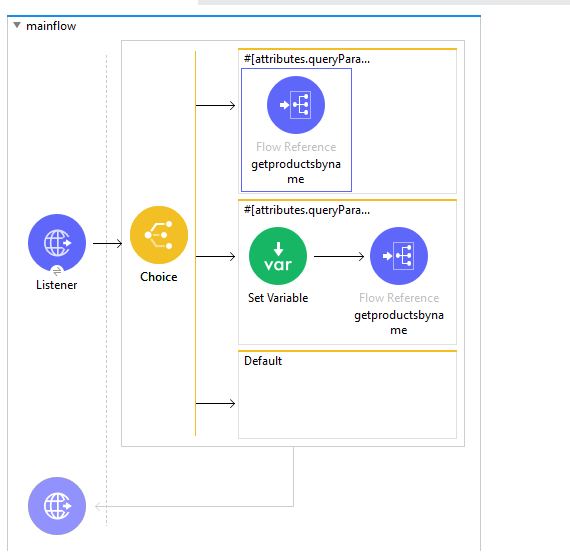
****

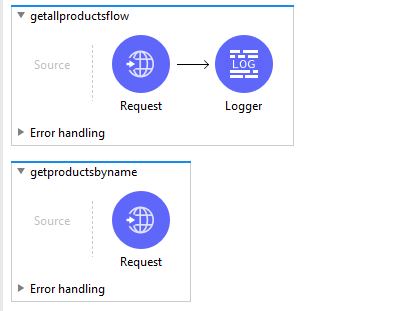
**Request has:** ****

Add new http request configuration, Just change **its base path and port alone**, others default.

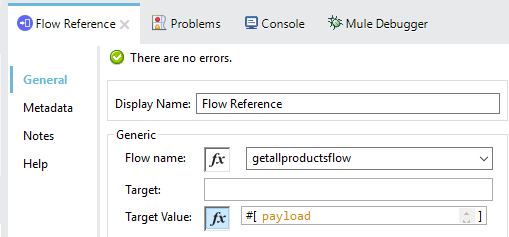
In request properties

First add http req config, then **change GET method and path /products and content payload**





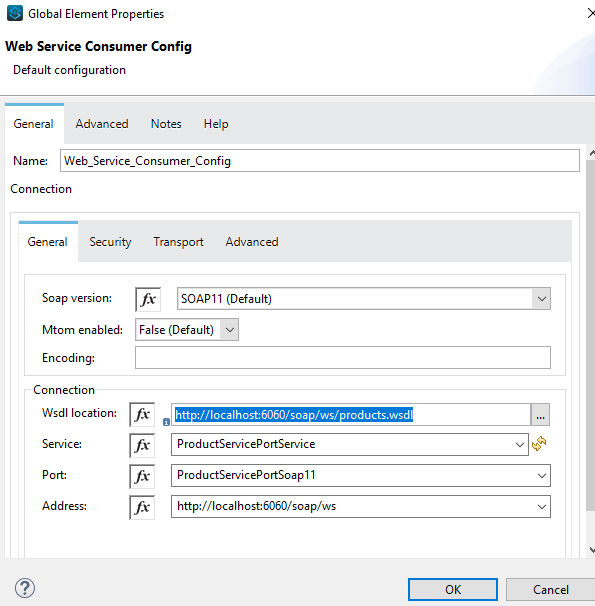
When we r passing flow reference we can set flow name where to go ,and also pass **target value** to pass, payload(EX)



**SOAP:**

If we want to consume soap webservices which we created with wsdl with various endpoints, we have to add module **web service consume and add consume**

Before add web service consumer config, start soap.bat to start the service. Then add connector configuration give the address of wsdl location, other fields automatically populated.



Once we added connector config,

Then in the properties in the next line, if we click operations, it automatically populates all the operations

**For encapsulation,**

For example if u r giving ${http.port} instead of 8081

Add config.yaml in src/main/resources

http:

port: “8081”

then in global elements click create

then click global configurations

then click configuration properties

then give the fle name u created with properties in resources folder yaml/properties file

the main thing is to connect yaml file with that, for that we need to configure

add config.yaml in configuration properties

${http.port} will be successful

But if we want to keep it secure, u have to add secure prop from exchange

Give the config.yaml in file name, if u want to use it directly with encrypted string value

If u want to encrypt whole file, then delete configuration property, instead in the file name give a new file name dev-secure-dev.yaml(which u can create by typing

java -cp secure-properties-tool.jar com.mulesoft.tools.SecurePropertiesTool file encrypt Blowfish CBC abcdefghijlmnop db-dev.yaml db-secure-dev.yaml)

new file is created in the location where u executed cmd

copy and save it to src/main/resources

now to refer encrypted data${secure::http.port}

is used.

To encrypt password using Blowfish:

java -cp secure-properties-tool.jar com.mulesoft.tools.SecurePropertiesTool string encrypt Blowfish CBC Mymulesoft root

secretkey generated: 2ZwyE/jBalQ=

Java –jar secure-properties-tool.jar string encrypt Blowfish CBC MyMuleSoft “JaiSai@01”

If it did’t work try java –cp secure-properties-tool.jar syntax

Here Mymulesoft is the key

Root is the password to encrypt

C:\demo>java -cp secure-properties-tool.jar com.mulesoft.tools.SecurePropertiesTool string encrypt Blowfish CBC Mymulesoft JaiSai@01

BkXOHFFT7ersufE4N+ZzsQ==

C:\demo>java -cp secure-properties-tool.jar com.mulesoft.tools.SecurePropertiesTool string encrypt Blowfish CBC Mymulesoft vjOPoO2FzDGMYB9z7fQjbLY3

SUbCIyoR8A2UpJOl7p6Z7cLlXGUL3uwER8Z/qkcUWPc=

sfdc.username=sujijothy@gmail.com

sfdc.pwd=![BkXOHFFT7ersufE4N+ZzsQ==]

sfdc.token=![SUbCIyoR8A2UpJOl7p6Z7cLlXGUL3uwER8Z/qkcUWPc=]

sfdc.loginurl=https://login.salesforce.com/services/Soap/u/48.0

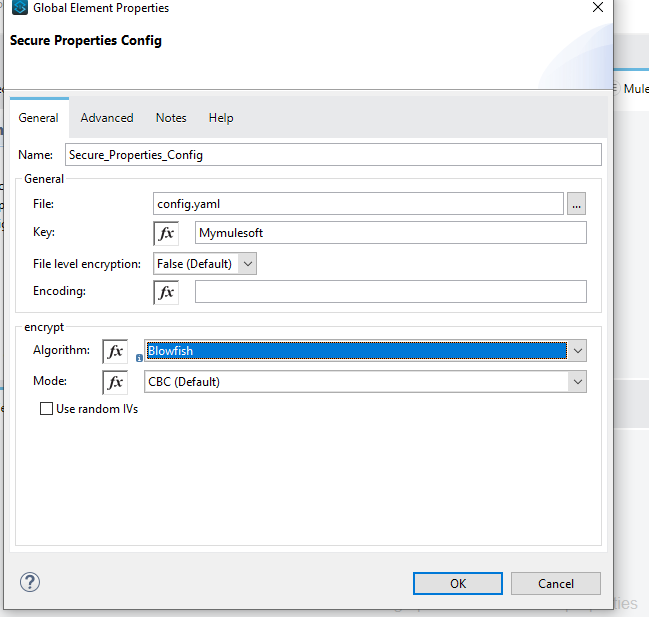
once encrypted then give

instead of ${db.pwd} we r including ${secure::…………….}

${secure::db.pwd}

For doing this we need to add secure config, to add that click search from exchange and add secure property

Then go to global elements and click connector configuration and choose secure properties config



db.host=localhost

db.user=root

db.pwd=![2ZwyE/jBalQ=]

db.database=emp

new salesforce token:

sfdc.pwd=![BkXOHFFT7ersufE4N+ZzsQ==]

sfdc.token=![KTq7erxaAMWTQowiEvqESJ+T2fD5cDCunXyPgfob2To=]

sfdc.loginurl=https://login.salesforce.com/services/Soap/u/48.0

Mule Server version: 4.3.0

Summary has list of resource list

API endpoints

/flights:

  is: [client-id-required]

  description: flight records from the database

  get:

    description: retrieves records from the database

    queryParameters:

      destination:

        required: false

        enum:

          - LAX

          - SFO

          - CLE

    responses:

      200:

        body:

          application/json:

            type: AmericanFlight[]

            examples:

              output: !include /exchange\_modules/68ef9520-24e9-4cf2-b2f5-620025690913/training-american-flights-example/1.0.1/AmericanFlightsExample.raml

post:

    description: add a recordto the database

    body:

      application/json:

        type: AmericanFlight

        examples:

          input: !include examples/AmericanFlightNoIDExample.raml

    responses:

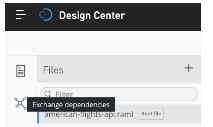
      201:

        body:

          application/json:

            example:

              responsemessage: Flight record added, nothng really happened

To add examples/datatypes from exchange u have to click exchange dependencies 

Click +sign and search for datatypes u built already and ready in exchange

Which u created in fragment can be used.

Once u added, click show in files. Then copy path to clipboard and add that to code

Types:

CustomErrorMessage: !include paste the path

An **API proxy** is your interface to developers that want to use your backend services. Rather than having them consume those services directly, they access an Edge **API proxy** that **you create**. With a **proxy**, **you** can provide value-added features such as: Security

It is important to note that API Proxies require an existing API while some **API Gateways** can assist in building a new API.

create folder IN box CONNECTOR

https://api.box.com/2.0/folders/

https://docs.mulesoft.com/box-connector/4.0/box-connector-examples

box request code token:

https://www.box.com/api/oauth2/authorize?response\_type=code&client\_id=fv12kstas83vyy21yhvq1ctifx0dfruq&state=authenticated

BEFORE RUNNING PGM:

http://localhost:8081/authorize

TO AUTHORIZE

DATAWEAVE:

%dw 2.0

Application/java

{

}

Here empty curly braces shows that it is map