**How to start active MQ**

>> C:\Soft\apache-activemq-5.17.2-bin\apache-activemq-5.17.2\bin>set JAVA\_HOME=C:\Soft\jdk-19\_windows-x64\_bin\jdk-19

>> C:\Soft\apache-activemq-5.17.2-bin\apache-activemq-5.17.2\bin>activemq start

**Agile Method:**

programme management ->we need the ducumentations (every program we can having the decumnetation)

|||

project manajor

|

it is divided into Splints

|

each splints having the TEAM

|

in this team having the one team leader and team memebers.......

|

tha total menthods is called the backlogs

Scrum calls...(time limited their for completion of the code)

GIT/SVM TOOL...

big project having the DEV environment....

DEV having the testing team

issu tracking system?

carbal?

**Dataweave:**

**DATAWEAVE(play ground is tool in mulosoft):**

**-----------------------------------------------**

**dataweave is a mulosoft language**

**1.extracting data**

**2.transforming data**

**\*\*\*\* ARRAYS,NUMBER,STRINGS,OBJECT.**

**The Candidtae name is sande sirisha and her id is 300 and she born on 10-06-2000 and she used to live in hitech city and her organisation is PhiDimensions.**

**"The Candidtae name is" ++ payload.Name ++ "and her id is" ++ payload.ID ++" and she born on" ++ payload.DOB ++ "and she used to live in" ++payload.Location**

**OBJECT**

**----------**

**it is alway enclosed with {} and has a key value pair.**

**ex**

**"name":"siri"**

**{"ID":[1,2,3,4,5]}**

**Array of object**

**----------------**

**{"name":["siri","err"]**

**}**

**ARRAY**

**-------**

**["siri",1,2,3,"sir"]**

**["siri",{"s":"name"}] Array contain the object**

**[1,2,3,3,[2,3,4],"siri"]Array contain the array**

**Extraction:**

**-----------**

**Data is taken from one or more sources or systems. The extraction locates and identifies relevant data, then prepares it for processing or transformation. Extraction allows many different kinds of data to be combined and ultimately mined for business intelligence.**

**Transformation:**

**-------------------**

**Once the data has been successfully extracted, it is ready to be refined. During the transformation phase, data is sorted, organized, and cleansed. For example, duplicate entries will be deleted, missing values removed or enriched, and audits will be performed to produce data that is reliable, consistent, and usable.**

**Loading: The transformed, high quality data is then delivered to a single, unified target location for storage and analysis.**

**What is a Payload in an API?**

**-----------------------------**

**The payload of an API is the data you are interested in transporting to the server when you make an API request. Simply put, it is the body of your HTTP request and response message.**

**map:**

**-----**

**The map function is used to transform the data contained in an array. It does this by iterating over the elements in the array and applying a transformation to each element. The result of the transformation is collected together and output as an array of transformed elements.**

**main Points:**

**--------------**

**The map function transforms data**

**If iterates over the elements in an array**

**It applies a transformation to each element**

**Applies only to array input**

**It can only output an array**

**Java Map Interface**

**-------------------**

**A map contains values on the basis of key, i.e. key and value pair. Each key and value pair is known as an entry. A Map contains unique keys.**

**A Map is useful if you have to search, update or delete elements on the basis of a key.**

**1.it can only contain only array...**

**$**

**----**

**when we write single doler,**

**we getting all employees in the data....**

**ex**

**---**

**payload map{**

**"name\_emplyee":$**

**}**

**$.specification**

**-------------------**

**payload map{**

**"name\_emplyee":$.Name**

**}**

**that we get only employe names...**

**$$**

**----**

**it will give index number of array(0)**

**payload map{**

**"name\_emplyee":$.Name,**

**"unique\_id":$.Name ++"-"++"employee id"++$.ID,**

**"what is $$":$$ >>>> type like this it is showing the index numbers**

**}**

**lamda expresion in map**

**-----------------------**

**INPUT...**

**--------------**

**[**

**{**

**"Name":"Pratik Kumar",**

**"ID":"100",**

**"Father":"Kaushlendra Kumar",**

**"DOB":"29-12-1999",**

**"Location":["Daltonganj","siri","e"**

**]},**

**{**

**"Name":"Piyush Kumari",**

**"ID":"200",**

**"DOB":"09-12-1994",**

**"Location":"Hyderabad"**

**},**

**{**

**"Name":"Sande Sirisha",**

**"ID":"300",**

**"DOB":"10-06-2000",**

**"Location":"Hitech City"**

**},**

**{**

**"Name":"Sande Sirisha",**

**"ID":"300",**

**"DOB":"10-06-2000",**

**"Location":"Hitech City"**

**},**

**{**

**"Name":"Rekha Anabarla",**

**"ID":"400",**

**"DOB":"08-12-1998",**

**"Location":"Hitech City"**

**},**

**{**

**"Name":"Sandya Sandrapangu",**

**"ID":"500",**

**"DOB":"01-11-1999",**

**"Location":"Hitech City"**

**}**

**]**

**SCRIPT**

**---------**

**%dw 2.0**

**output application/csv**

**---**

**payload map ((item, index )->**

**{**

**"name": item.Name,**

**"ID": item.ID,**

**"DOB":item.DOB,**

**"index":index**

**}**

**)**

**OUTPUT**

**--------**

**name,ID,DOB,index**

**Pratik Kumar,100,29-12-1999,0**

**Piyush Kumari,200,09-12-1994,1**

**Sande Sirisha,300,10-06-2000,2**

**Sande Sirisha,300,10-06-2000,3**

**Rekha Anabarla,400,08-12-1998,4**

**Sandya Sandrapangu,500,01-11-1999,5**

**programm..**

**-----------**

**[{**

**"id": "0001",**

**"type": "donut",**

**"name": "Cake",**

**"ppu": 0.55,**

**"ppu1":56,**

**"batter" : [**

**{**

**"type":"irregular",**

**"id":"1000"**

**},**

**{**

**"id": "1001",**

**"type": "Regular"**

**},**

**{**

**"id": "1002",**

**"type": "Chocolate"**

**},**

**{**

**"id": "1003",**

**"type": "Blueberry"**

**},**

**{**

**"id": "1004",**

**"type": "Devil's Food"**

**}**

**],**

**"topping": [{**

**"id": "5001",**

**"type": "None"**

**},**

**{**

**"id": "5002",**

**"type": "Glazed"**

**},**

**{**

**"id": "5005",**

**"type": "Sugar"**

**},**

**{**

**"id": "5007",**

**"type": "Powdered Sugar"**

**},**

**{**

**"id": "5006",**

**"type": "Chocolate with Sprinkles"**

**},**

**{**

**"id": "5003",**

**"type": "Chocolate"**

**},**

**{**

**"id": "5004",**

**"type": "Maple"**

**}**

**]**

**}]**

**%dw 2.0**

**output application/json**

**---**

**payload map ((sandya, index )->**

**{**

**"name": sandya.name,**

**"type":sandya."type",**

**"id": sandya.id,**

**"ppu":sandya.ppu,**

**"inside of the object":sandya.batter."type",**

**"inside of the object":sandya.batter.id,**

**"inside of the object is topping":sandya.topping."type",**

**"inside of the object is topping":sandya.topping.id**

**}**

**)**

**[**

**{**

**"name": "Cake",**

**"type": "donut",**

**"id": "0001",**

**"ppu": 0.55,**

**"inside of the object": [**

**"irregular",**

**"Regular",**

**"Chocolate",**

**"Blueberry",**

**"Devil's Food"**

**],**

**"inside of the object": [**

**"1000",**

**"1001",**

**"1002",**

**"1003",**

**"1004"**

**],**

**"inside of the object is topping": [**

**"None",**

**"Glazed",**

**"Sugar",**

**"Powdered Sugar",**

**"Chocolate with Sprinkles",**

**"Chocolate",**

**"Maple"**

**],**

**"inside of the object is topping": [**

**"5001",**

**"5002",**

**"5005",**

**"5007",**

**"5006",**

**"5003",**

**"5004"**

**]**

**}**

**]**

**BY USING OF THE MAP OBJECT**

**----------------------------**

**%dw 2.0**

**output application/json**

**var employee=**

**[{**

**"firstName": "James",**

**"lastName": "Peter",**

**"age":30,**

**"salary":30000**

**},**

**{**

**"firstName": "Peter",**

**"lastName": "Gonsalves",**

**"age":28,**

**"salary":60000**

**},**

**{**

**"firstName": "Anthony",**

**"lastName": "Watson",**

**"age":27,**

**"salary":50000**

**}]**

**---**

**employee filter $.salary < 40000 map (items,index) -> (**

**items mapObject (**

**if($$ as String == "firstName")**

**{**

**(upper($$)):$,**

**"STATUS":"Active"**

**}**

**else**

**{**

**(upper($$)):$**

**}**

**)**

**)**

**Map Object**

**----------------**

**{**

**"firstName":"Avery",**

**"lastName":"Chance",**

**"age":56,**

**"occupation":"Physicist"**

**}**

**%dw 2.0**

**output application/json**

**---**

**mapObject(payload, ((v,k,idx) ->**

**(upper(k)): upper(v)**

**))**

**output**

**-------**

**{**

**"FIRSTNAME": "AVERY",**

**"LASTNAME": "CHANCE",**

**"AGE": "56",**

**"OCCUPATION": "PHYSICIST"**

**}**

**Dataweave:**

**CREATING DATA**

**---------------**

**Creating data required**

**1.STRING**

**2.ARRAY**

**3.NUMBERS**

**4.BOOLEAN**

**How to find the data type in dataweave**

**---------------------------------------**

**SCRIPTING**

**----------**

**%dw 2.0**

**output JSON**

**---**

**{**

**"siri":typeOf("Hello world!"),**

**"siri1":typeOf({}),**

**"siri3":typeOf([1,2,3,2]),**

**"siri4":typeOf(1),**

**"siri2":typeOf(false)**

**}**

**In this way we get at a time 4 types.**

**we can not create the object only we get the 1 types and remaining we put the comment**

**OUTPUT**

**-------**

**it is in object**

**---------------**

**{**

**"siri": "String",**

**"siri1": "Object",**

**"siri3": "Array",**

**"siri4": "Number",**

**"siri2": "Boolean"**

**}**

**it is in the array**

**------------------**

**%dw 2.0**

**output JSON**

**---**

**[**

**typeOf("Hello world!"),typeOf([]),typeOf(true),typeOf(1)**

**]**

**output**

**------**

**[**

**"String",**

**"Array",**

**"Boolean",**

**"Number"**

**]**

**another format**

**---------------**

**%dw 2.0**

**output yaml**

**---**

**"It is a" ++" " ++ typeOf({})**

**output**

**--------**

**It is a Object**

**-----------------------**

**CSV Format**

**-------------**

**firstName,lastName,age**

**John,Smith,45**

**Jane,Doe,34**

**JSON Format**

**---------------**

**[**

**{**

**"firstName": "John",**

**"lastName": "Smith",**

**"age": 45**

**},**

**{**

**"firstName": "Jane",**

**"lastName": "Doe",**

**"age": 34**

**}**

**]**

**YAML format**

**---------------**

**---**

**-**

**firstName: John**

**lastName: Smith**

**age: 45**

**-**

**firstName: Jane**

**lastName: Doe**

**age: 34**

**XML- FORMAT**

**---------**

**%dw 2.0**

**output xml**

**---**

**{**

**titles: {**

**"title":["Titanic","Avatar"]**

**}**

**}**

**ouput**

**--------**

**<titles>**

**<title>Titanic</title>**

**<title>Avatar</title>**

**</titles>**

**Git hub command**

**GO TO ANY OTHER FILES**

**----------------------**

**click right click....got to GIT BASH HERE.....**

**\*\*\*\*\*ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents\*\*\*\***

**$ pwd (it is showing present working location...)**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents >>>(Documents is present working location)...**

**$ git init**

**\*\*\*Initialized empty Git repository in C:/Users/ssande/OneDrive - Phidimensions\_O365/Documents/.git/\*\*\*\*\***

**IN THIS WAY IT IS SHOWING ALL DOCUMENTS**

**----------------------------------------**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git status**

**On branch master**

**No commits yet**

**Untracked files:**

**(use "git add <file>..." to include in what will be committed)**

**123 LINUX.docx**

**DATA CENTER.docx**

**DATA CENTER.pdf**

**DIFFERENCE BETWEEN LINUX AND UNIX.docx**

**ADD THE FILE**

**-------------**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git add mouniting.txt**

**CHEKE THE STATUS**

**----------------**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git status**

**COMMIT THE FILE**

**---------------**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git commit -m "first commit"**

**TO GIVE THE WHAT IS THE LOCATION OF REPROCITRY**

**----------------------------------------------**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git remote add origin https://github.com/sandeshirisha/sande.git**

**UCAN PUSH THE IN A REFROSITRY**

**----------------------------**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git push -u origin master**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git status**

**AFTER MODIFIED THE FILES**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**add the file\*\*\*\*\*\*\***

**---------------------**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git add mouniting.txt**

**COMMIT THE FILE**

**--------------------**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git commit -m "second commit"**

**[master 09a4d52] second commit**

**1 file changed, 4 insertions(+)**

**PUSH THE FILE**

**---------------**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git push**

**How to clone from git hub**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents**

**$ pwd**

**/c/Users/ssande/OneDrive - Phidimensions\_O365/Documents**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents**

**$ git init**

**Initialized empty Git repository in C:/Users/ssande/OneDrive - Phidimensions\_O365/Documents/.git/**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git status**

**On branch master**

**No commits yet**

**Untracked files:**

**(use "git add <file>..." to include in what will be committed)**

**123 LINUX.docx**

**DATA CENTER.docx**

**DATA CENTER.pdf**

**DIFFERENCE BETWEEN LINUX AND UNIX.docx**

**DIFFERENCE BETWEEN LINUX AND UNIX.pdf**

**DIFFERENCE BETWEEN WINDOWS AND LINUX.docx**

**DIFFERENCE BETWEEN WINDOWS AND LINUX.pdf**

**Hipervizer.txt**

**desktop.ini**

**mouniting.txt**

**sande123.pptx**

**siri.c**

**sirill.txt**

**nothing added to commit but untracked files present (use "git add" to track)**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git add mouniting.txt**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git status**

**On branch master**

**No commits yet**

**Changes to be committed:**

**(use "git rm --cached <file>..." to unstage)**

**new file: mouniting.txt**

**Untracked files:**

**(use "git add <file>..." to include in what will be committed)**

**123 LINUX.docx**

**DATA CENTER.docx**

**DATA CENTER.pdf**

**DIFFERENCE BETWEEN LINUX AND UNIX.docx**

**DIFFERENCE BETWEEN LINUX AND UNIX.pdf**

**DIFFERENCE BETWEEN WINDOWS AND LINUX.docx**

**DIFFERENCE BETWEEN WINDOWS AND LINUX.pdf**

**Hipervizer.txt**

**desktop.ini**

**sande123.pptx**

**siri.c**

**sirill.txt**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git commit -m "first commit"**

**[master (root-commit) e70262b] first commit**

**1 file changed, 344 insertions(+)**

**create mode 100644 mouniting.txt**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git remote add origin https://github.com/sandeshirisha/sande.git**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git push -u origin master**

**Enumerating objects: 3, done.**

**Counting objects: 100% (3/3), done.**

**Delta compression using up to 8 threads**

**Compressing objects: 100% (2/2), done.**

**Writing objects: 100% (3/3), 2.60 KiB | 2.60 MiB/s, done.**

**Total 3 (delta 0), reused 0 (delta 0), pack-reused 0**

**To https://github.com/sandeshirisha/sande.git**

**\* [new branch] master -> master**

**branch 'master' set up to track 'origin/master'.**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git status**

**On branch master**

**Your branch is up to date with 'origin/master'.**

**Changes not staged for commit:**

**(use "git add <file>..." to update what will be committed)**

**(use "git restore <file>..." to discard changes in working directory)**

**modified: mouniting.txt**

**Untracked files:**

**(use "git add <file>..." to include in what will be committed)**

**123 LINUX.docx**

**DATA CENTER.docx**

**DATA CENTER.pdf**

**DIFFERENCE BETWEEN LINUX AND UNIX.docx**

**DIFFERENCE BETWEEN LINUX AND UNIX.pdf**

**DIFFERENCE BETWEEN WINDOWS AND LINUX.docx**

**DIFFERENCE BETWEEN WINDOWS AND LINUX.pdf**

**Hipervizer.txt**

**desktop.ini**

**sande123.pptx**

**siri.c**

**sirill.txt**

**no changes added to commit (use "git add" and/or "git commit -a")**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git add mouniting.txt**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git commit -m "second commit"**

**[master 09a4d52] second commit**

**1 file changed, 4 insertions(+)**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**$ git push**

**Enumerating objects: 5, done.**

**Counting objects: 100% (5/5), done.**

**Delta compression using up to 8 threads**

**Compressing objects: 100% (2/2), done.**

**Writing objects: 100% (3/3), 295 bytes | 295.00 KiB/s, done.**

**Total 3 (delta 1), reused 0 (delta 0), pack-reused 0**

**remote: Resolving deltas: 100% (1/1), completed with 1 local object.**

**To https://github.com/sandeshirisha/sande.git**

**e70262b..09a4d52 master -> master**

**ssande@HGT-LAP-258 MINGW64 ~/OneDrive - Phidimensions\_O365/Documents (master)**

**C:\GIT-JAVA>git clone https://github.com/sandeshirisha/sandeshirisha.git**

**Cloning into 'sandeshirisha'...**

**remote: Enumerating objects: 9, done.**

**Receiving objects: 100% (9/9), done., done.**

**remote: Compressing objects: 100% (6/6), done.**

**remote: Total 9 (delta 0), reused 3 (delta 0), pack-reused 0**

**C:\GIT-JAVA>cd eis**

**The system cannot find the path specified.**

**C:\GIT-JAVA>git clone https://github.com/rbandaruuniphi/eis.git**

**Cloning into 'eis'...**

**remote: Enumerating objects: 21, done.**

**remote: Counting objects: 100% (21/21), done.**

**remote: Compressing objects: 100% (14/14), done.**

**remote: Total 21 (delta 4), reused 0 (delta 0), pack-reused 0**

**Receiving objects: 100% (21/21), 4.42 KiB | 4.42 MiB/s, done.**

**Resolving deltas: 100% (4/4), done.**

**C:\GIT-JAVA>cd eis**

**C:\GIT-JAVA\eis>dir**

**Volume in drive C has no label.**

**Volume Serial Number is C65C-0EA7**

**Directory of C:\GIT-JAVA\eis**

**06-10-2022 02:23 <DIR> .**

**06-10-2022 02:23 <DIR> ..**

**06-10-2022 02:23 17 Anji**

**06-10-2022 02:23 25 Pratik**

**06-10-2022 02:23 29 raveendar**

**06-10-2022 02:23 61 README.md**

**06-10-2022 02:23 18 rekha**

**06-10-2022 02:23 19 Sandya**

**06-10-2022 02:23 21 shirisha**

**7 File(s) 190 bytes**

**2 Dir(s) 447,106,990,080 bytes free**

**C:\GIT-JAVA\eis>**

**How to download java in the linux**

**check java in the linux**

**----------------------------**

**$ java -version**

**it is not install**

**jrk installation process**

**------------------------**

**$ sudo apt install default-jre**

**check jre is found are not**

**---------------------------**

**$ java -version**

**Install JDK process**

**-------------------**

**first check JDK is not found are not**

**------------------------------------**

**command is - $ javac -version**

**it is not found the install the jdk**

**-----------------------------------**

**sudo apt install default-jdk**

**after installation process**

**---------------------------**

**just check $ javac -version**

**How to use error handling in the batch process**

%dw 2.0

**output** **application/json**

---

{

"errorMessage": **(Batch::getStepExceptions()).Batch\_Step.localizedMessage**, this the command to stored the all error in the batch process while running in the project

"id": payload."Line Item Number"[0]

}

**How to use the update function in the dataweave**

%dw 2.0

**output** **application/json**

---

flatten(payload) map (

$ **update** {

**case** GENERICATTRIBUTE32 **at** .GENERICATTRIBUTE32-> p('secure::fileName.TXSTA') ++ vars.fileName

}

)

**Data function**

(now()) **as** String {format : "yyyyMMdd\_HHmmss\_MMMMdd" } ++ "\_" ++ p('secure::fileName.formate')

**Notes:**

When we are working in the batch process in the project while that entering in the batch aggregator it is command like object code.

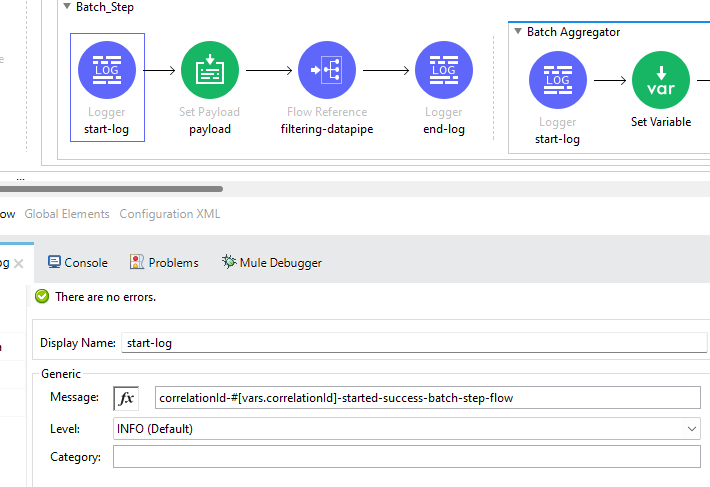
**Ex: { [] [] []**

**} like this that case we use the platen function to convert array flatten(payload)**

**It is giving like : [ {},{}…..{}**

**]**

**How create category in the logger:**

****

**Changes some configuration in the log4j2.xml**

**Step1:**

go to log4j2.xml set configuration inside of the logger

<Loggers>

<!-- Http Logger shows wire traffic on DEBUG -->

<!--AsyncLogger name="org.mule.service.http.impl.service.HttpMessageLogger"

level="DEBUG"/ -->

<AsyncLogger name=*"org.mule.service.http"* level=*"WARN"* />

<AsyncLogger name=*"org.mule.extension.http"* level=*"WARN"* />

<AsyncLogger name=*"com.bacthJob.DPtransaction"* level=*"DEBUG"*>

</logger>

This configuration help to creating logger level. the default logger level is 1. Error 2. Info 3. *WARN*

We have to create another logger level we can use like <AsyncLogger name=*"com.bacthJob.DPtransaction"* level=*"DEBUG"*> inside of the loggers

com.batchjob.DPtransaction indicated like category name

step 2:

if you want create logger message like file manner use this configuration. Add this configuration inside of appender.

<RollingFile name=*"com.bacthJob.DPtransactionappender"*

fileName=*"${sys:mule.home}${sys:file.separator}logs${sys:file.separator}com.bacthJob.DPtransactionappender.log"*

filePattern=*"${sys:mule.home}${sys:file.separator}logs${sys:file.separator}batch-process-%i.log"*>

<PatternLayout

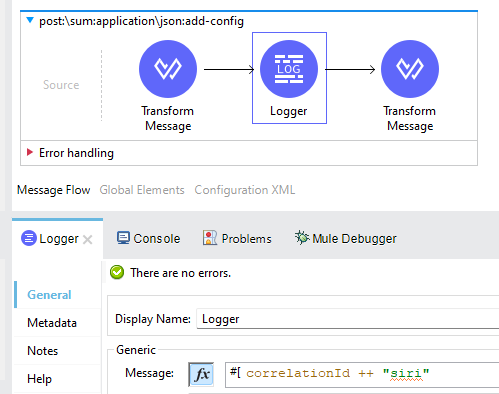
pattern=*"%-5p %d [%t] [processor: %X{processorPath}; event: %X{correlationId}] %c: %m%n"* />

<SizeBasedTriggeringPolicy size=*"10 MB"* />

<DefaultRolloverStrategy max=*"10"* />

**</RollingFile>**

**How to create correlation id with out creating of variable**

 this is configuration set correlation Id

**How to create Pipeline**

Go to this link

<https://medium.com/another-integration-blog/mule-4-continuous-integration-using-azure-devops-b8cf6f6083ba>

https://dzone.com/articles/configuring-jenkins-pipeline-for-deploying-mulesof

**step 1:** set the could hub configuration in the pom.xml file

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-clean-plugin</artifactId>

<version>3.2.0</version>

</plugin>

<plugin>

<groupId>org.mule.tools.maven</groupId>

<artifactId>mule-maven-plugin</artifactId>

<version>${mule.maven.plugin.version}</version>

<extensions>true</extensions>

<configuration>

<cloudHubDeployment>

<username>${username}</username>

<password>${password}</password>

<workers>${workers}</workers>

<workerType>${worker.type}</workerType>

<environment>${environment}</environment>

<muleVersion>${mule.version}</muleVersion>

<applicationName>${application.name}</applicationName>

<objectStoreV2>true</objectStoreV2>

<properties>

<maven.compiler.target>1.8</maven.compiler.target>

<maven.compiler.source>1.8</maven.compiler.source>

</properties>

</cloudHubDeployment>

<classifier>---- select project type ----</classifier>

</configuration>

</plugin>

</plugins>

</build>

Step 2: commit the over all code in the azure repo with out any errors

By using of git commands

Step 3: create pipe line the by using of yaml file

That goal is “goals:  'package deploy -DmuleDeploy -Dusername=$(username) -Dpassword=$(password) -Dworkers=$(workers) -Dworker.type=$(worker.type) -Denvironment=$(environment) -Dmule.version=$(mule.version) -Dapplication.name=$(application.name)'”

**How to create RMAL in the efficient**

**How to create Library**

#%RAML 1.0 Library

resourceTypes:

  emp:

    post:

      description: This API will create Employee Information

      body:

        application/json:

          type: !include ../dataType/empInformationRequest.raml

          example: !include ../example/empInformationRequest.json

      responses:

        200:

          body:

            application/json:

              type: !include ../dataType/common.raml

              example: !include ../example/common.json

    put:

      body:

        application/json:

          type: !include ../dataType/empInformationRequest.raml

          example: !include ../example/empInformationRequest.json

      responses:

        200:

          body:

            application/json:

              type: !include ../dataType/common.raml

              example: !include ../example/updateSuccessResponse.json

    patch:

      body:

        application/json:

          type: !include ../dataType/empInformationRequest.raml

          example: !include ../example/empInformationRequest.json

      responses:

        200:

          body:

            application/json:

              type: !include ../dataType/common.raml

              example: !include ../example/updateSuccessResponse.json

**how to call Library in the root file**

resourceTypes:

   resourceType:

     type: empResources

uses:

   empResources: /libraries/emp-info.raml

/empInfo:

   /emp:

     type: {empResources.emp}

**traits**

#%RAML 1.0 Trait

responses:

  400:

    body:

      application/json:

        type: !include ../dataType/errorResponse.raml

        example: !include ../example/400.json

  404:

    body:

      application/json:

        type: !include ../dataType/errorResponse.raml

        example: !include ../example/404.json

  405:

    body:

      application/json:

        type: !include ../dataType/errorResponse.raml

        example: !include ../example/405.json

**How to call traits in the root file:**

traits:

  errorResponse: !include traits/errorResponse.raml

/empInfo:

  /emp:

    is:

      - errorResponse

**How to call secure file in the root file:**

#%RAML 1.0 SecurityScheme

type: x-client-enforcement

describedBy:

  headers:

    client\_id:

      type: string

      required: true

      displayName:  Client Id

    client\_secret:

      type: string

      required: true

      displayName: Client Secret

    correlation\_Id:

      type: string

      required: true

      displayName: Correlation Id

  responses:

    401:

      body:

        application/json:

          properties:

            error:

              type: string

              description: A description of the error.

              example: Client Credentials Invalid

          example: |

            {

              "error": "Client Credentials Invalid"

            }

**securitySchemes:**

**customSecuritySchemes:** !include /securityScheme/customSecurityScheme.raml

**securedBy:**

  - customSecuritySchemes

/empInfo:

  /emp:

    is:

      - errorResponse

    type: {empResources.emp}

**How to create custom connector in the mulesoft :**

<https://www.salesforce.com/blog/custom-connector-mule-sdk/>

**How to deploy mulesoft code in the could hub by using of maven command**

[**https://dzone.com/articles/mule-maven-plugin-for-deploying-mulesoft-applicati**](https://dzone.com/articles/mule-maven-plugin-for-deploying-mulesoft-applicati)

<https://developer.mulesoft.com/tutorials-and-howtos/quick-start/how-to-deploy-using-maven-anypoint-studio-mule-application/>

**How to deploy by using of Jenkins**

[**https://dzone.com/articles/configuring-jenkins-pipeline-for-deploying-mulesof**](https://dzone.com/articles/configuring-jenkins-pipeline-for-deploying-mulesof)

**Monday**

**Learn about ESB, SOAP and rest services**

**Tue:**

**Jenkins, dataweave functions**

**\*\*\*\*Learn about SNOW FLAKE**

<https://docs.mulesoft.com/snowflake-connector/1.1/>

\*\*How to deploy the code by using of Jenkins

<https://www.youtube.com/watch?v=EVxk5O05aHw>

while deploying in the could 2.0

* You need to have anypoint-exchange-v3 as a repository Id with proper organization Id in the URL
* Check **settings.xml** file to ensure it is available for all builds run on a given machine.
* Check the Maven configuration setup from Windows->preferences->Maven
* If you want to publish private assets then make sure you have to set up **distributionManagement in pom.xml with proper org id**

Ssl

SSL certificates are used for securing the transport layer and creating a secure communication between the service caller and the server. SSL channel authenticate the identity of the caller with server and encrypt the data exchanged between caller and server. In Organization it’s a standard to have only SSL based services.

Please go through <https://docs.oracle.com/cd/E19798-01/821-1841/gjrgy/> to understand more about keystore generation.

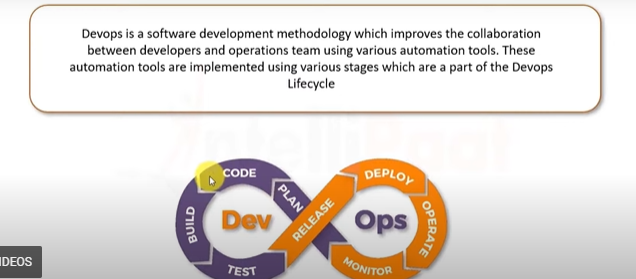
Azure [DevOps](https://intellipaat.com/blog/what-is-azure-devops/)

==================

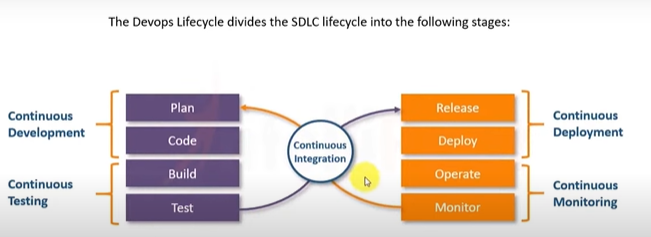
Why we use Devops

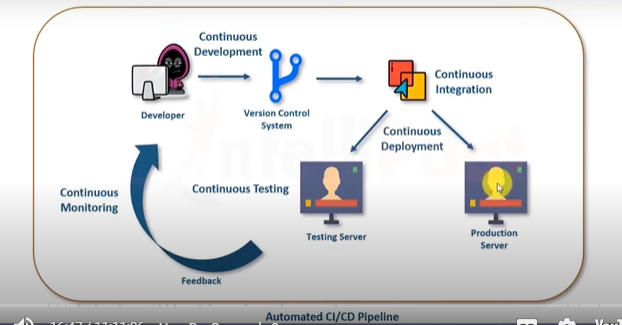
1. More productivity
2. Less time invested in the planning
3. Frequent releases, with continues feedback makes achieving targets easy
4. Team is divided into specialized goals

Why the devo’s

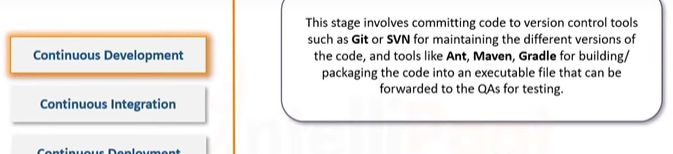


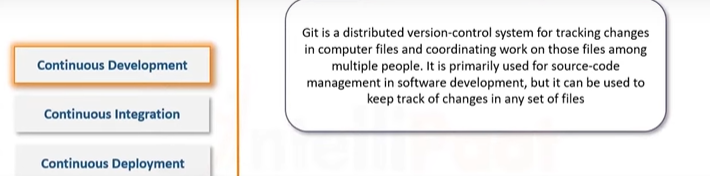
**How devo’s works**

****

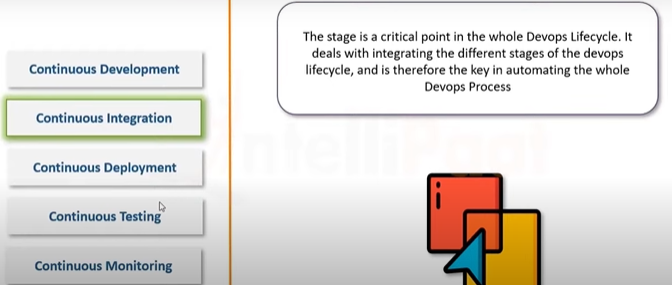
****

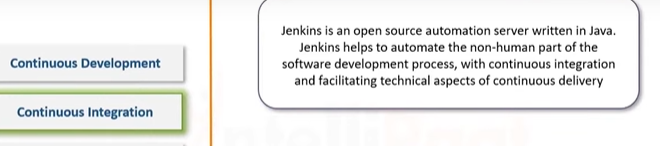
Continuous development



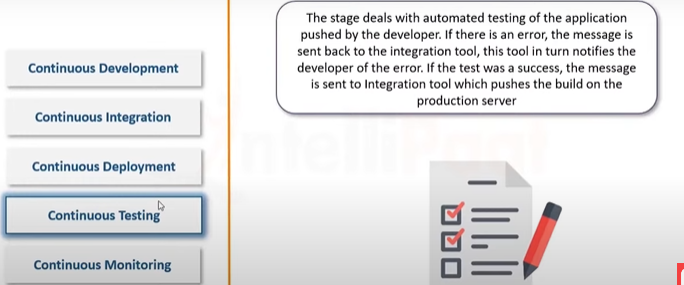


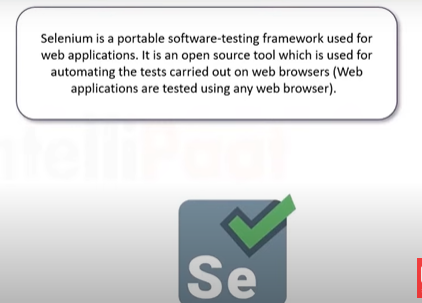
Continuous integration tool





Continuous testing

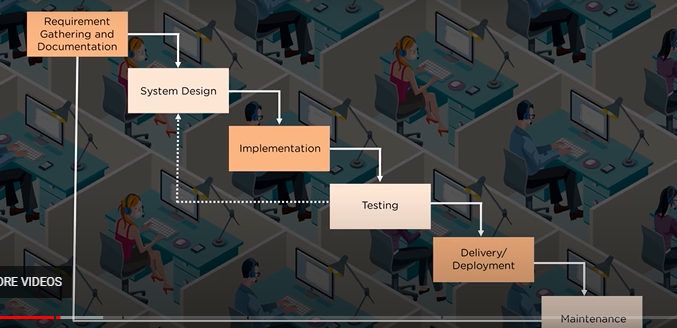


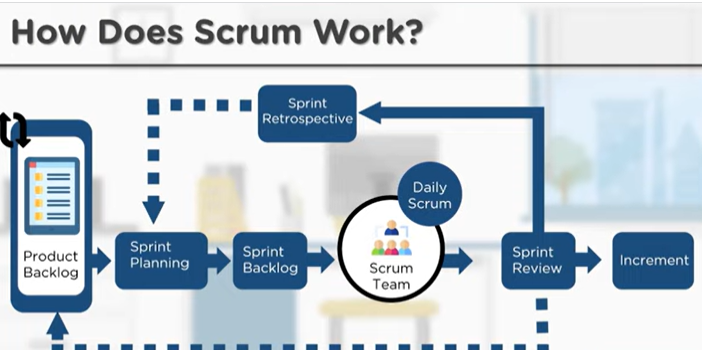


Continuous monitoring

Dataweave learning super web site

https://www.prostdev.com/post/dataweave-2-0-core-functions-cheatsheet





If you want download local standalone server

<https://developer.mulesoft.com/download-mule-esb-runtime/>

salesforce

select Id,Name,BillingStreet,BillingCity,BillingState,BillingCountry,BillingPostalCode from Account where Id = '0015g00001QR0fVAAT'

**Anypoint Partner Manager**

Anypoint Partner Manager enables you to perform bidirectional Business-to-Business (B2B) message exchanges using APIs and Electronic Data Interchange (EDI) formats, and to conduct business transactions such as buying and selling products with your partners. Transactions are typically transmitted over transports such as HTTPS and SFTP, and exchanges such as AS2. Partner Manager supports X12, JSON, and XML message formats.

**Partner Manager client**

bidirectional Business-to-Business exchanges using APIs and Electronic Data Interchange (EDI) formats, and to conduct business transactions such as buying and selling products with your partners

**as2**

MuleSoft’s AS2 connector allows Anypoint customers to send and receive business transactions electronically with added security over HTTP or HTTPS protocols, from within their Mule Applications. The AS2 connector available within the Anypoint platform delivers below outcomes:

* Enables secured B2B communication with trading partner community from the unified Anypoint platform; Organizations do not need to have personnel with different skill set.
* Supports large file processing through the usage of streaming.
* Enables AS2 communications from the Cloud.

use Partner Manager to:

* Perform technical onboarding of your partners.
* Set up, manage, and monitor business message flows between your systems and your partners' systems.
* Track and report on business transactions to achieve greater operational efficiency in B2B transactions.
* View meta information, validation errors, and all messages included with each transmission payload.

Transmission payloads can contain one or more business messages.

Before you begin using Partner Manager, you must have:

* An Anypoint Platform user account with the Partner Manager entitlement
* The Partner Manager Administrator role, which can assign permission levels to all users who need access to B2B message flows in their environments (sandbox or production)
* The required Mule vCore capacity to manage your B2B transaction workload needs
* Storage for the B2B transmission payloads you send or receive, such as a database, Amazon S3, Azure Blob Storage, and so on
* Licenses for the AS2, EDIFACT, or X12 connectors used in your B2B message flows
* Anypoint Studio 7.4 or later
* Familiarity with EDI message formats
* The ability to create DataWeave maps to use for performing data transformations

## **Partner Manager Concepts**

To use Partner Manager, you must understand the following concepts:

### **Host**

Your Anypoint Platform organization or business group. A Partner Manager implementation has a single host for which you define a host profile.

### **Partners**

The partners with whom you send and receive B2B transmissions. You define a partner profile for each partner in your B2B ecosystem.

### **Payload Storage API**

An API that provides Partner Manager with a standardized way to connect to your solution for storing transmission content.

### **Message Flows**

The paths along which a B2B message travels from its source, through a DataWeave translation map, to its target. There are two types of message flows:

* Inbound Message Flows

Inbound message flows receive messages from partners, including messages that go through third-party connections. When you create an inbound message flow, you can validate and transform the inbound messages and then send them to your backend applications.

* Outbound message flows

Outbound message flows receive messages from your backend applications, validate and transform these messages to the format your partners expect, and then send them to your partners, either directly or through third-party connections.

Message flows consist of [endpoints](https://docs.mulesoft.com/partner-manager/2.x/#endpoints), [message types](https://docs.mulesoft.com/partner-manager/2.x/#message-types), [sender and receiver identifiers](https://docs.mulesoft.com/partner-manager/2.x/#identifiers), and [translation maps](https://docs.mulesoft.com/partner-manager/2.x/#translation-maps) (optional).

### Endpoints

Define the transport protocol configurations necessary for:

* Receiving messages from external partners or internal backend applications
* Sending messages to external partners or backend applications

You can configure the following types of endpoints:

* Receive from Partners

Source endpoint in an inbound message flow that receives inbound B2B messages from partners either directly or via third-party connections

* Target at Host

Target endpoint in an inbound message flow through which transformed messages are sent to the backend applications

* Source at Host

Source endpoint in an outbound message flow that receives application messages from the backend applications

Send to Partners

Target endpoint in an outbound message flow through which transformed B2B messages are sent to the partners either directly or via third-party connections

### Message Types

Define the structure of the transactions that Partner Manager receives or sends. Partner Manager supports the following message types:

* Receive from Partners

Defines the structure of the transactions that partners or third-party connections send to the host through inbound message flows. You can define [custom message attributes](https://docs.mulesoft.com/partner-manager/2.x/#custom-attributes) for this message type.

* Target at Host

Defines the structure of the transactions that the host sends to the backend through inbound message flows.

Source at Host

Defines the structure of the transactions that the backend sends to the host through outbound message flows. You can define custom message attributes for this message type. You can define [custom message attributes](https://docs.mulesoft.com/partner-manager/2.x/#custom-attributes) for this message type.

* Send to Partners

Defines the structure of the transactions that the host sends to partners or third-party connections through outbound message flows.

### Sender and Receiver Identifiers

Identifiers in AS2, EDIFACT, and X12 transactions that identify the message senders and receivers. For X12 messages, the identifiers in the ISA and GS headers provide information that identifies the appropriate flow to process the transaction.

### Custom Message Attributes

User-defined attributes that you can associate with a Receive from Partners or Source at Host message type. Using these attributes provides visibility into the full lifecycle of your B2B workflow.

### Translation Maps

DataWeave maps transform business transactions between your partner’s message formats and your organization’s enterprise application message formats. You create the maps in Studio and import them into Partner Manager when you configure message flows.

### Templates

Prebuilt templates that convert your B2B message flow configurations into runtime applications. When MuleSoft upgrades a template, you can upgrade your message flows to use them to take advantage of new capabilities that enable better integration with your partner’s ecosystem.

Resources path

Web site

<https://help.mulesoft.com/s/question/0D52T00005WbgC6SAJ/how-to-include-resource-type-from-fragment-in-another-raml>

reusable fragment -raml

--------------------------

<https://github.com/mulesoft-catalyst/reusable-raml-fragments>

## **api Autodiscovery**

<https://dzone.com/articles/using-api-autodiscovery-in-anypoint-platform#:~:text=Once%20you%20save%20the%20configuration,is%20the%20API%20Autodicovery%20tag>.

RAML:

When ever are getting error with “#” symbol include with fields

Example : field : Name # (that ur getting syntax error in the RAML)

Resolve the this syntax using double quotes

Like: “Name #”

Reusable raml link

<https://github.com/mulesoft-catalyst/reusable-raml-fragments>

secure keys:

secure-key-command:☹encrypt single value

first download: java -cp secure-properties-tool.jar and use download path like this “C:\Users\ssande\Downloads

C:\Users\ssande\Downloads>java -cp secure-properties-tool.jar com.mulesoft.tools.SecurePropertiesTool string encrypt AES CBC "oW-VG&gR46\*i$tbh" "PW72WCSij7fd9SDtEbAAkYpQ"

0g80G8pBfNwH67WHLXGfp26IrNkxQov4AGpc+n7+6UA=

AES-aligartham

CBS-mode

O to h-secure.key

P to Q-values

secure-key-command:☹decrypt single value

C:\Users\ssande\Downloads>java -cp secure-properties-tool.jar com.mulesoft.tools.SecurePropertiesTool string decrypt AES CBC "oW-VG&gR46\*i$tbh" "lW13xCeK2mVQMF4ZIOVBWg=="

Pull&Shoot1(decrypt value)

Error (important)

When ur getting error on the http request in the mulesoft:

Error

"HTTP GET on resource '[https://api.jillamy.com/dev-netsuite-communities-sapi-v1:443/api/in/945'](https://api.jillamy.com/dev-netsuite-communities-sapi-v1:443/api/in/945%27) failed: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target."

Use this link :

https://stackoverflow.com/questions/76179285/command-line-flag-to-disable-ssl-certification-validation-in-java

Solution

Use some commands in the run as arguments(pass that command run time arguments)

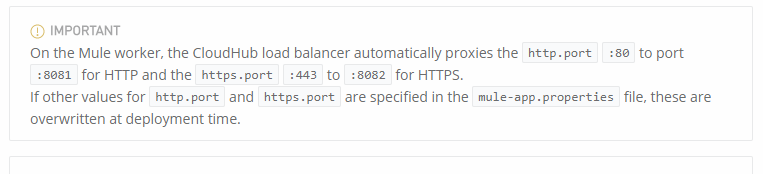
-Djdk.internal.httpclient.disableHostnameVerification=true

-Dcom.sun.security.enableAIAcaIssuers=true

-Dtrust\_all\_cert=true

-Dcom.sun.net.ssl.checkRevocation=false

http port number



How set keys in the properties file

Dev.properties file

Mention like keys

http.port=8081

[db.password=siri@123](mailto:db.password=siri@123)

like this

you can call in the config like

${ http.port}

**How to clone the code by using url desktop**

* 1. Get the clone url
  2. Go to git desktop
  3. Open the files
  4. Choose clone option in the git desktop menu
  5. Choose url option in the cloning
  6. Cloning is done
* **Note**: git desktop and studio path should be same
* **Importing**
* **-----------------------**
* Go to studio
* Go to file menu
* Choose import option for same file located what we in the git desktop path location
* Import successful

**Workers with 0.1 vCores and 0.2 vCores:**

* Provide limited CPU and I/O for apps with smaller workloads
* Can burst to higher CPU speeds for a short time

This ability helps to improve application startup times and to process infrequent, large workloads. If you need consistent performance, use workers with more vCores.