itid-api-fw-resource-feed - Technical Details

Resourcefeeder API is designed to update services with resource and type details based on Environment and Process Types. It delegates resource feed at three levels: Middleware, DataCenter and DBAI. The detailed flow of the same is shown as below:

Command: java –jar itid-api-fw-resource-feed.jar ENV=UAT ProcessType=DBAI

How does it connect and talk to external resources?

Resource feed API includes HTTPClient Utility that delegates all connection management and data exchange via HTTP methods over HTTP protocol maintaining secure connection using SSL authentication.

How does it authenticate with various feeds? Connects over SSL via username password to fetch all data resources under DBAI inventories. DBAI inventories are scrolled for each resource feed page wise and its dumped as pre work done as part of preprocessing.

What are main feeds: There are three main resource feeds- Middleware, Datacenter and DBAI. Each maintains service data along with owner data. This involves- service id, service owner, owner email, host server details and its meta data.

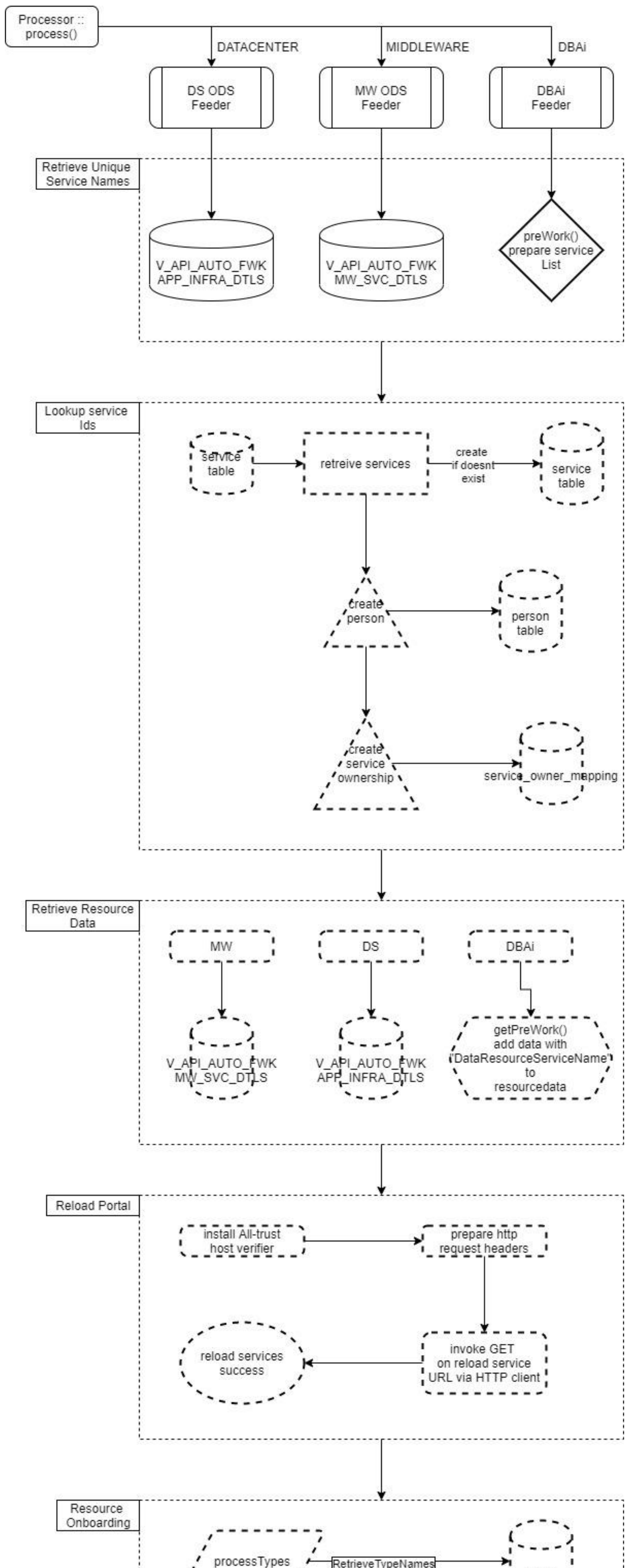
Who maintains feeds at time of writing: Feeds are always processed via environment details and process type data.

How do we test the feed is correct? By validating it against Data resource type, data resource name along with host name and environment details.

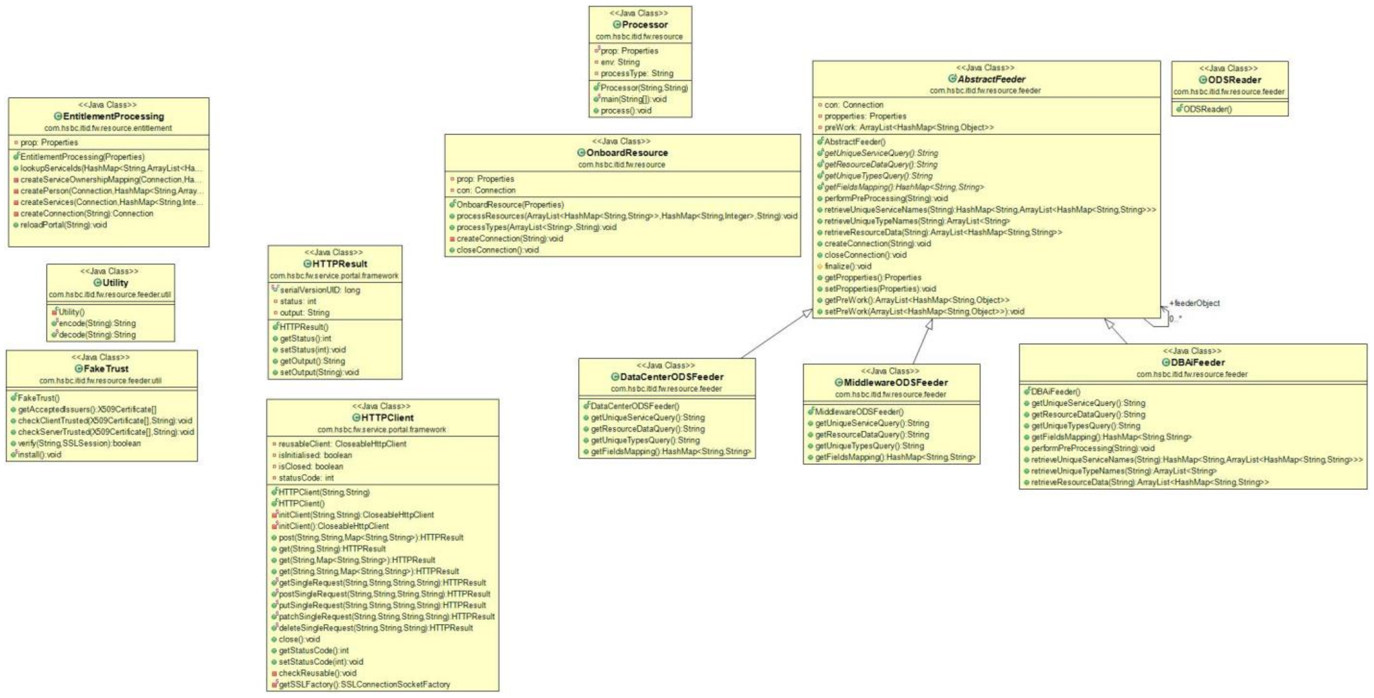
Anything from feeds that is important: Unique service names are mapped to their respective owner data in distinct manner for each feed. For DBAI- service names are mapped to owner data via resource data fetched as part of preprocessing. For Middleware and Datacenter- service names are mapped to owner data via direct DB queries involving retrieval from V_API_AUTO_FWK_MW_SVC_DTLS and V_API_AUTO_FWK_APP_INFRA_DTLS tables respectively.

Entitlements validation and person creation is another task handled as part of resource feed api. Processing involves first a check for service that are doesn't exist in specifc environment and then creating those which doesn't exists. Further a create person action is also executed using service owner data fetched in earlier steps. This processing also creates service to owner mapping in service_owner_mapping table. Finally, services are reloaded to make sure updates are reflected via ITIDSP/PerformAction/reloadservice endpoint.

Views involved, groups service accounts it run it as:


Processor :: process()
DATACENTER
MIDDLEWARE
DBAi
DS ODS Feeder
MW ODS Feeder
DBAi Feeder
Retrieve Unique Service Names
V_API_AUTO_FWK APP_INFRA_DTLS
V_API_AUTO_FWK MW_SVC_DTLS
preWork() prepare service List
Lookup service Ids
service table
retreive services
create if doesnt exist
service table
create person
person table
create service ownership
service_owner_mapping
Retrieve Resource Data
MW
DS
DBAi
V_API_AUTO_FWK MW_SVC_DTLS
V_API_AUTO_FWK APP_INFRA_DTLS
getPreWork() add data with 'DataResourceServiceName' to resourcedata
Reload Portal
install All-trust host verifier
prepare http request headers
invoke GET on reload service URL via HTTP client
reload services success
Resource Onboarding
processTypes
RetrieveTypeNames

Class diagram is shown as below:


<<Java Class>>
Processor
com.hsbc.itid.fw.resource
prop: Properties
env: String
processType: String
Processor(String,String)
main(String[]):void
process():void
<<Java Class>>
EntitlementProcessing
com.hsbc.itid.fw.resource.entitlement
prop: Properties
EntitlementProcessing(Properties)
lookupServiceIds(HashMap<String,ArrayList<Ha...
createServiceOwnershipMapping(Connection,Ha...
createPerson(Connection,HashMap<String,Array...
createServices(Connection,HashMap<String,Inte...
createConnection(String):Connection
reloadPortal(String):void
<<Java Class>>
OnboardResource
com.hsbc.itid.fw.resource
prop: Properties
con: Connection
OnboardResource(Properties)
processResources(ArrayList<HashMap<String,String>>,HashMap<String,Integer>,String):void
processTypes(ArrayList<String>,String):void
createConnection(String):void
closeConnection():void
<<Java Class>>
AbstractFeeder
com.hsbc.itid.fw.resource.feeder
con: Connection
propperties: Properties
preWork: ArrayList<HashMap<String,Object>>
AbstractFeeder()
getUniqueServiceQuery():String
getResourceDataQuery():String
getUniqueTypesQuery():String
getFieldsMapping():HashMap<String,String>
performPreProcessing(String):void
retrieveUniqueServiceNames(String):HashMap<String,ArrayList<HashMap<String,String>>>
retrieveUniqueTypeNames(String):ArrayList<String>
retrieveResourceData(String):ArrayList<HashMap<String,String>>
createConnection(String):void
closeConnection():void
finalize():void
getPropperties():Properties
setPropperties(Properties):void
getPreWork():ArrayList<HashMap<String,Object>>
setPreWork(ArrayList<HashMap<String,Object>>):void
+feederObject
0..*
<<Java Class>>
ODSReader
com.hsbc.itid.fw.resource.feeder
ODSReader()
<<Java Class>>
Utility
com.hsbc.itid.fw.resource.feeder.util
Utility()
encode(String):String
decode(String):String
<<Java Class>>
HTTPResult
com.hsbc.fw.service.portal.framework
serialVersionUID: long
status: int
output: String
HTTPResult()
getStatus():int
setStatus(int):void
getOutput():String
setOutput(String):void
<<Java Class>>
FakeTrust
com.hsbc.itid.fw.resource.feeder.util
FakeTrust()
getAcceptedIssuers():X509Certificate[]
checkClientTrusted(X509Certificate[],String):void
checkServerTrusted(X509Certificate[],String):void
verify(String,SSLSession):boolean
install():void
<<Java Class>>
DataCenterODSFeeder
com.hsbc.itid.fw.resource.feeder
DataCenterODSFeeder()
getUniqueServiceQuery():String
getResourceDataQuery():String
getUniqueTypesQuery():String
getFieldsMapping():HashMap<String,String>
<<Java Class>>
MiddlewareODSFeeder
com.hsbc.itid.fw.resource.feeder
MiddlewareODSFeeder()
getUniqueServiceQuery():String
getResourceDataQuery():String
getUniqueTypesQuery():String
getFieldsMapping():HashMap<String,String>
<<Java Class>>
DBAiFeeder
com.hsbc.itid.fw.resource.feeder
DBAiFeeder()
getUniqueServiceQuery():String
getResourceDataQuery():String
getUniqueTypesQuery():String
getFieldsMapping():HashMap<String,String>
performPreProcessing(String):void
retrieveUniqueServiceNames(String):HashMap<String,ArrayList<HashMap<String,String>>>
retrieveUniqueTypeNames(String):ArrayList<String>
retrieveResourceData(String):ArrayList<HashMap<String,String>>
<<Java Class>>
HTTPClient
com.hsbc.fw.service.portal.framework
reusableClient: CloseableHttpClient
isInitialised: boolean
isClosed: boolean
statusCode: int
HTTPClient(String,String)
HTTPClient()
initClient(String,String):CloseableHttpClient
initClient():CloseableHttpClient
post(String,String,Map<String,String>):HTTPResult
get(String,String):HTTPResult
get(String,Map<String,String>):HTTPResult
get(String,String,Map<String,String>):HTTPResult
getSingleRequest(String,String,String,String):HTTPResult
postSingleRequest(String,String,String,String):HTTPResult
putSingleRequest(String,String,String,String):HTTPResult
patchSingleRequest(String,String,String,String):HTTPResult
deleteSingleRequest(String,String,String):HTTPResult
close():void
getStatusCode():int
setStatusCode(int):void
checkReusable():void
getSSLFactory():SSLConnectionSocketFactory