External Stage (DRP stage) Local Setup.  
  
Prerequisites

**- VPN (pritunnel) profile for access LGC2 DB**

**- Checkout config repository(**<https://github.com/ICTASL/udi-poc-mosip-config/tree/1.1.5-drp-mosip-config>**)**

**- Checkout registration-processor repository(<https://github.com/ICTASL/udi-poc-registration/tree/1.1.5> )**

- Backend Service (Auth Service) - <https://github.com/ICTASL/udi-poc-drp-backend/tree/development>  
 - DRP Angular Client - <https://github.com/ICTASL/udi-poc-drp-frontend/tree/development>

**Step 1: Setup spring cloud-config-server**

* Pull the docker image from here.

<https://hub.docker.com/r/hyness/spring-cloud-config-server>

* Checkout config repository from here.

<https://github.com/ICTASL/udi-poc-mosip-config/tree/1.1.5-drp-mosip-config-viraj>

or <https://github.com/ICTASL/udi-poc-mosip-config/tree/1.1.5-drp-mosip-config>

* Then execute bellow command (Set Spring Config Repo path to yours).

**docker run -it --name=spring-cloud-config-server -p 51000:8888 -v C:/Users/Pcadmin/IdeaProjects/udi-poc-mosip-config:/config -e SPRING\_CLOUD\_CONFIG\_SERVER\_GIT\_URI=file:/config -e spring.cloud.config.server.git.searchPaths=sandbox\* -e spring.cloud.config.server.git.force-pull:false hyness/spring-cloud-config-server**

A picture containing text

Description automatically generated

* Test config server on Chrome.  
  <http://localhost:51000/registration-processor-mz/default/1.1.5-drp-mosip-config>

Text

Description automatically generated

**Step 2: Build registration-processor source code  
  
mvn clean install -Dgpg.skip -DskipTests**

(Copy & store the registration-processor-common-camel-bridge-1.1.5.jar, registration-processor-securezone-notification-stage-1.1.5.jar and the registration-processor-external-stage-1.1.5.jar to a easily accessible folder)

**Step 3: Run required camel services locally.**

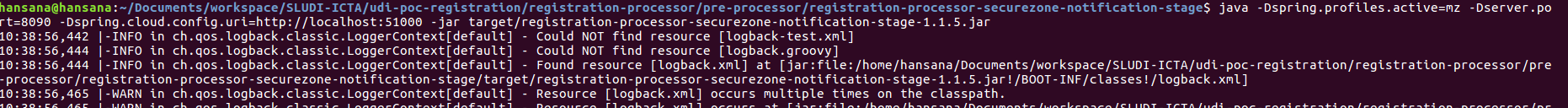
*Run securezone notification stage, camel bridge and the required stage in different ports in local machine*

**Running Camel bridge in local env**java -Dspring.profiles.active=mz -Dserver.port=8022 -Dspring.cloud.config.uri=http://localhost:51000 -jar target/registration-processor-common-camel-bridge-1.1.5.jar

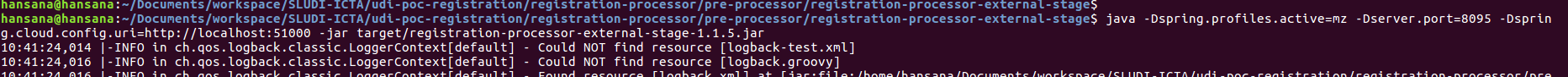
Graphical user interface

Description automatically generated with low confidence

**Runing securezone notification stage in local env** java -Dspring.profiles.active=mz -Dserver.port=8090 -Dspring.cloud.config.uri=http://localhost:51000 -jar target/registration-processor-securezone-notification-stage-1.1.5.jar



**Running external stage in local env**java -Dspring.profiles.active=mz -Dserver.port=8095 -Dspring.cloud.config.uri=http://localhost:51000 -jar target/registration-processor-external-stage-1.1.5.jar



STG DB IP: 192.168.204.11

User: postgres

Pass: mosip123

You can change these properties in registration-processor-mz.properties(Config Repository)

**Step 4:***The packet that is used for testing should be already available in object store, so first run the packet once with the cloud env, where all the services are running. It should atleast pass the packet uploader stage to make this work.*We already have done this, and a suitable RID is available in our DB.

**Step 5:**

*Open the below swagger URL in browser to authenticate and get a Authorization token*

URL : https://<env host name>/v1/authmanager/swagger-ui.html#/authmanager/clientIdSecretKeyUsingPOST

Click "Try it out" and update the below body

Body: {

"id": "string",

"metadata": {},

"request": {

"appId": "regproc",

"clientId": "mosip-regproc-client",

"secretKey": "<password>"

},

"requesttime": "2018-12-10T06:12:52.994Z",

"version": "string"

}

So used the below request body which worked,Fetch auth token

https://digitalid.lgcc.gov.lk/v1/authmanager/authenticate/clientidsecretkey POST Api

Body =>

{

"id": "string",

"metadata": {},

"request": {

"appId": "regproc",

"clientId": "mosip-regproc-client",

"secretKey": "abc123"

},

"requesttime": "2021-06-11T06:12:52.994Z",

"version": "1.0.0"

}

Text, table

Description automatically generated with medium confidence

**Step 6:**

*Use a rest client application like postman and make a request with the packet Id to the local securezone notfication stage using the below request template.*

*The step-6 will trigger the event to the local securezone notification stage, this stage will send the event after validation with DB (running in the cloud env) to the camel bridge running in the local evn. The camel bridge will route the event to the required stage since the local XML points the same. Now since the required stage is connected to the other required services from the same env, it should be able to process the packet.*

Response Auth token should be added to domain cookie.  
Graphical user interface, text, application, email, website

Description automatically generated

Securezone Notification Api

http://localhost:8090/registrationprocessor/v1/securezone/notification POST

Body =>

{

"reg\_type": "NEW",

"rid": "10002101020000320210604061235", //Use the RID that was already uploaded to the cloud env as per step 4

"isValid": true,

"internalError": false,

"messageBusAddress": "",

"retryCount": 5

}

DRP Stage Api (Trigger below request to debug the External stage)

http://localhost:8095/registrationprocessor/v1/external/drpstage POST

Body =>

{

"apiName": "LIST",

"reg\_type": "NEW",

"rid" : "",

"isValid": true,

"inteArnalError": false,

"messAageBusAddress": "",

"retrAyCount": 5,

"request": {

"operatorId": "OPERATOR1",

"centerId": "CENTER1"

}

}

Put a Debug point within the Reg-proc code (ExternalStage.java) to continue with the debugging.