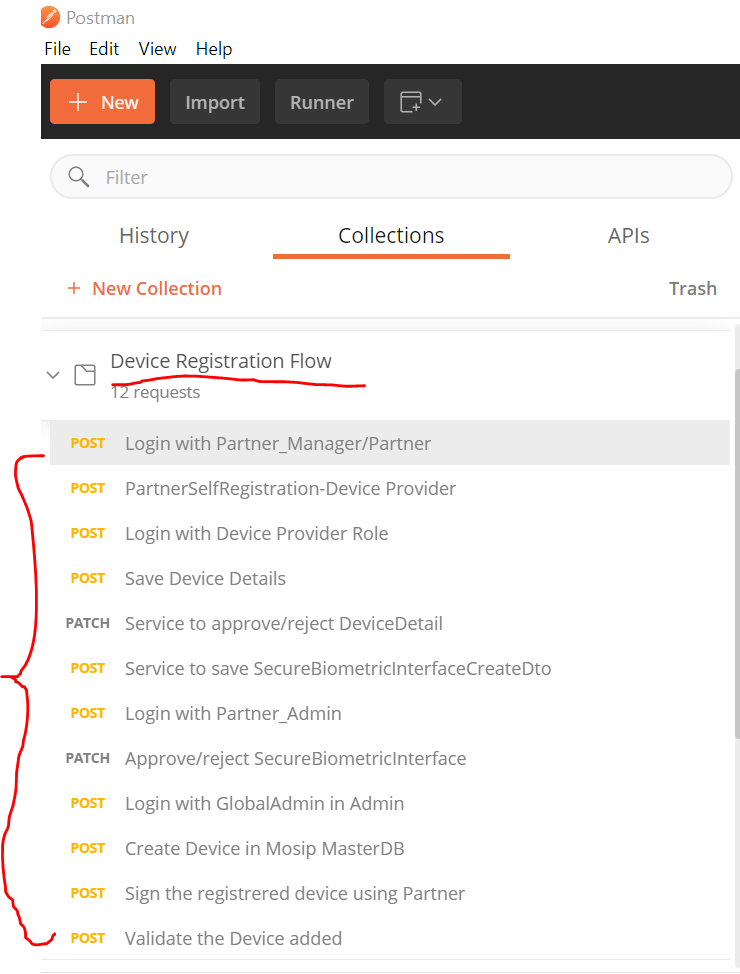
**Device Registration API Flow With Postman**

Prerequisites Note:

\*\* Configure the ICTA Vpn setup for your local machine and turn it ON.

*Refer email sent by Ravindu Jayadeera <*[*ravinduj@icta.lk*](mailto:ravinduj@icta.lk)*> for this*

1. Download this Postman Pack:   
   <https://github.com/ICTASL/UDI-poc/blob/master/documentations/postman_collection/Device%20Registration%20Flow.postman_collection.json>
2. Import the above json as a new Collection to Postman. Refer figure below to see a successfully imported json..



1. Click on the 1st POST request *“Login with Partner\_Manager/Partner”,*  In the address bar change the request URL variable {{url}} to **https://aws.digitalid.lgcc.gov.lk**

*Sample request payload:*

*{*

*"id": "string",*

*"metadata": {},*

*"request": {*

*"appId": "partner",*

*"password": "Techno@123",*

*"userName": "110122"*

*},*

*"requesttime": "2021-03-10T04:48:15.445Z",*

*"version": "string"*

*}*

*Response for  Successful login:*

*{*

*"id": "string",*

*"version": "string",*

*"responsetime": "2021-03-10T07:59:06.579Z",*

*"metadata":* ***null****,*

*"response": {*

*"status": "success",*

*"message": "Username and password combination had been validated successfully"*

*},*

*"errors":* ***null***

*}*

*Response for  Failed login:*

*{*

*"id": "string",*

*"version": "string",*

*"responsetime": "2021-03-10T08:01:44.171Z",*

*"metadata":* ***null****,*

*"response":* ***null****,*

*"errors": [*

*{*

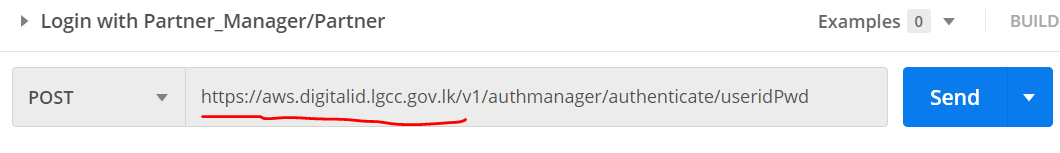
*"errorCode": "KER-ATH-023",*

*"message": "Invalid Credentials"*

*}*

*]*

*}*



1. Apply the change in **Step 3** to the other requests as required before Sending...

API Requests :

**1.Login with Partner\_Manager/Partner [POST]**

<https://aws.digitalid.lgcc.gov.lk/v1/authmanager/authenticate/useridPwd>

Request payload:  
{

  "id": "string",

  "metadata": {},

  "request": {

    "appId": "partner",

    "password": "Techno@123",

    "userName": "110122"

  },

  "requesttime": "2021-03-10T04:48:15.445Z",

  "version": "string"

}

Response:

{

    "id": "string",

    "version": "string",

    "responsetime": "2021-03-10T11:08:16.818Z",

    "metadata": **null**,

    "response": {

        "status": "success",

        "message": "Username and password combination had been validated successfully"

    },

    "errors": **null**

}

**2. PartnerSelfRegistration-Device Provider [POST]**

<https://aws.digitalid.lgcc.gov.lk/partnermanagement/v1/partners/partners>

Request payload:

{

  "id": "string",

  "metadata": {},

  "request": {

    "address": "99x",

    "contactNumber": "01123456789",

    "emailId": "test@123.com",

    "organizationName": "99X Technology",

    "partnerId": "99XIO",

    "partnerType": "Device\_Provider",

    "policyGroup": "Device Provider"

  },

  "requesttime": "2021-12-25T04:55:50.853Z",

  "version": "string"

}

Response:

{

    "id": "string",

    "version": "string",

    "responsetime": "2021-03-10T11:03:46.685Z",

    "metadata": **null**,

    "response": {

        "partnerId": "99XIO",

        "status": "Active"

    },

    "errors": **null**

}

policyGroup here refers to the defined policy group **name** column in **mosip\_pms.policy\_group** table

To-clarify: **policy\_group** attribute here is not validated here. The partner is registered successfully even though the policy\_group is not in the DB. figure below..

Example request below,  
{

  "id": "string",

  "metadata": {},

  "request": {

    "address": "99x",

    "contactNumber": "01123456789",

    "emailId": "test99x33@123.com",

    "organizationName": "99X Technology33",

    "partnerId": "99XPartner33",

    "partnerType": "Device\_Provider",

    "policyGroup": "some dummy policy group!!!!!"

  },

  "requesttime": "2021-12-25T04:55:50.853Z",

  "version": "string"

}

Response:

{

    "id": "string",

    "version": "string",

    "responsetime": "2021-03-11T03:23:33.325Z",

    "metadata": **null**,

    "response": {

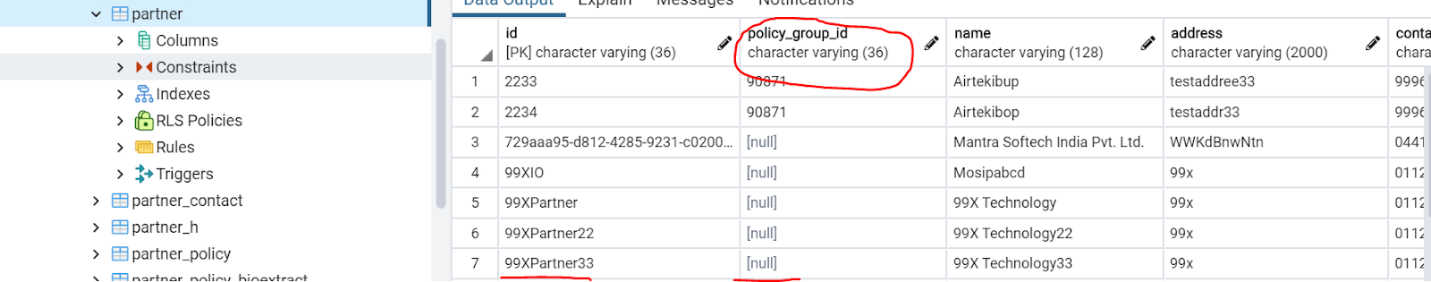
        "partnerId": "99XPartner33",

        "status": "Active"

    },

    "errors": **null**

}



**3.Login with Device Provider Role**  
Same as Login with Partner\_Manager/Partner API request.

**4. Save Device Details [POST]**  
<https://aws.digitalid.lgcc.gov.lk/partnermanagement/v1/partners/devicedetail>

Request payload:

{

  "id": "string",

  "metadata": {},

  "request": {

    "deviceProviderId": "99XIO",

    "deviceSubTypeCode": "Double",

    "deviceTypeCode": "Iris",

    "id": "242425",

    "isItForRegistrationDevice": **true**,

    "make": "EYECOOL",

    "model": "TCI322",

    "partnerOrganizationName": "99X Technology"

  },

  "requesttime": "2021-02-25T09:40:35.029Z",

  "version": "string"

}

Response:

{

    "id": **null**,

    "version": **null**,

    "responsetime": "2021-03-11T04:04:48.580Z",

    "metadata": **null**,

    "response": {

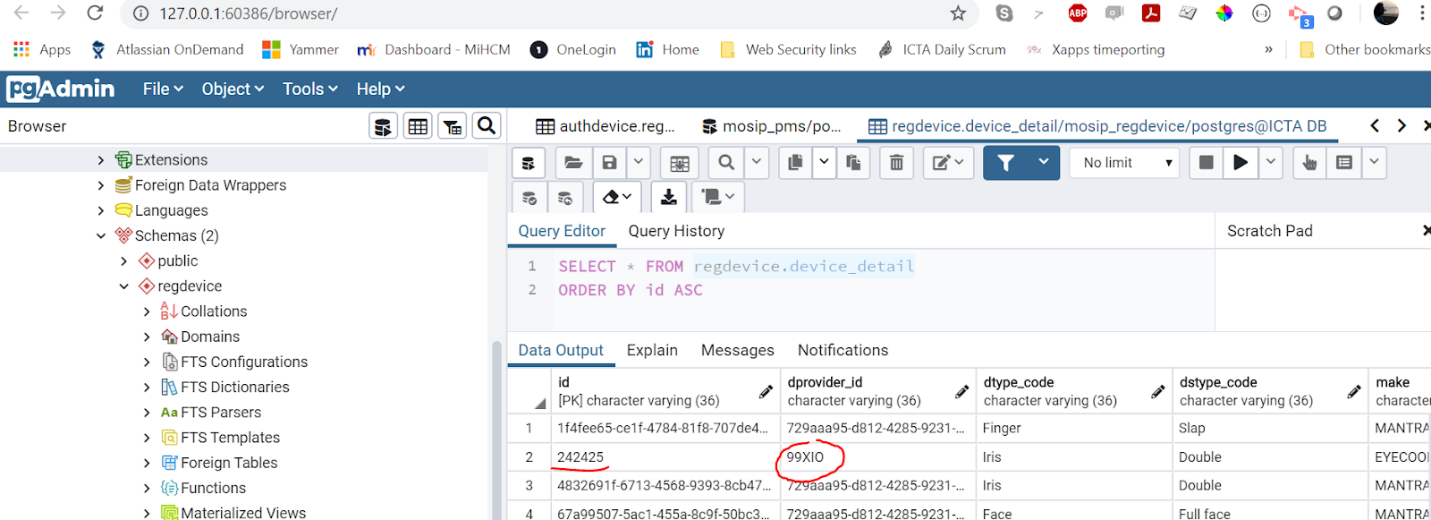
        "id": "242425"

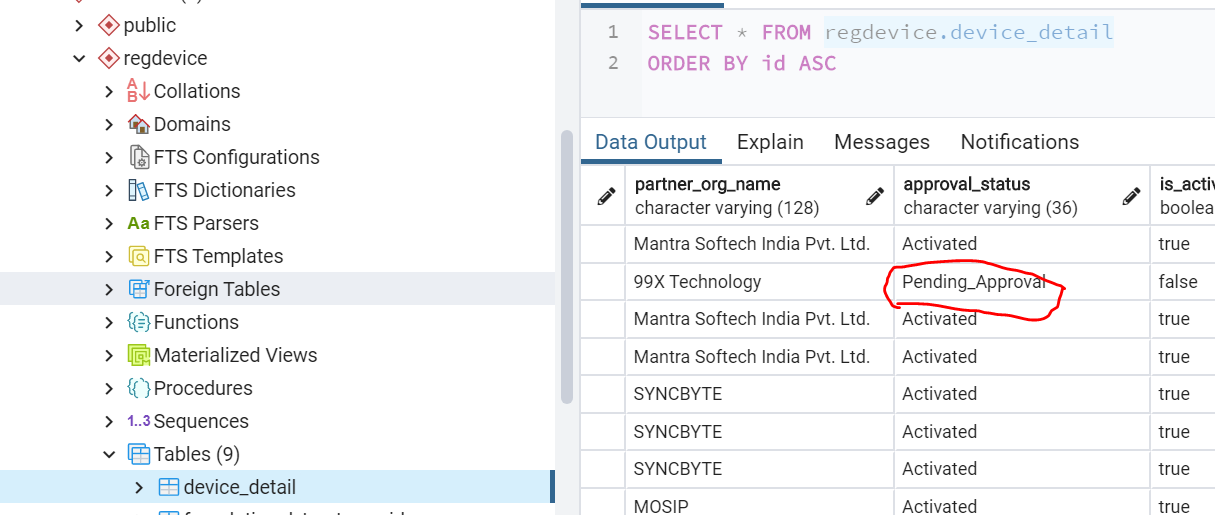
    },

    "errors": **null**

}

So the deviceProviderId here refers to the partnerId param (of PartnerSelfRegistration-Device Provider request above), and partnerOrganizationName refers to the organizationName param (of PartnerSelfRegistration-Device Provider request above)  
  
deviceSubTypeCode and deviceTypeCode are coming from  **mosip\_authdevice.reg\_device\_sub\_type** table.

As per the Mosip MDS specification there is something called the **deviceinfo API.** this APi is used to get the information of the plugged-in device for the above request payload param values. Name, model etc.. etc… So when request for the device details through this api it responds with an encrypted string, which should be decrypted back to get the plain text which contains the full device info we need to run the above request.  
  
After registering above device it is saved in the **mosip\_regdevice.device\_detail** table with approval\_status set to *Pending\_Approval* **..  
**

****

**5. Service to approve/reject DeviceDetail [PATCH]**

Done by the partner manager, will come and approve the device by below request.  
<https://aws.digitalid.lgcc.gov.lk/partnermanagement/v1/partners/devicedetail>

Request payload:

{

  "id": "string",

  "metadata": {},

  "request": {

    "approvalStatus": "Activate",

    "id": "242425",

    "isItForRegistrationDevice": **true**

  },

  "requesttime": "2021-02-25T06:51:38.294Z",

  "version": "string"

}

Response:

{

    "id": **null**,

    "version": **null**,

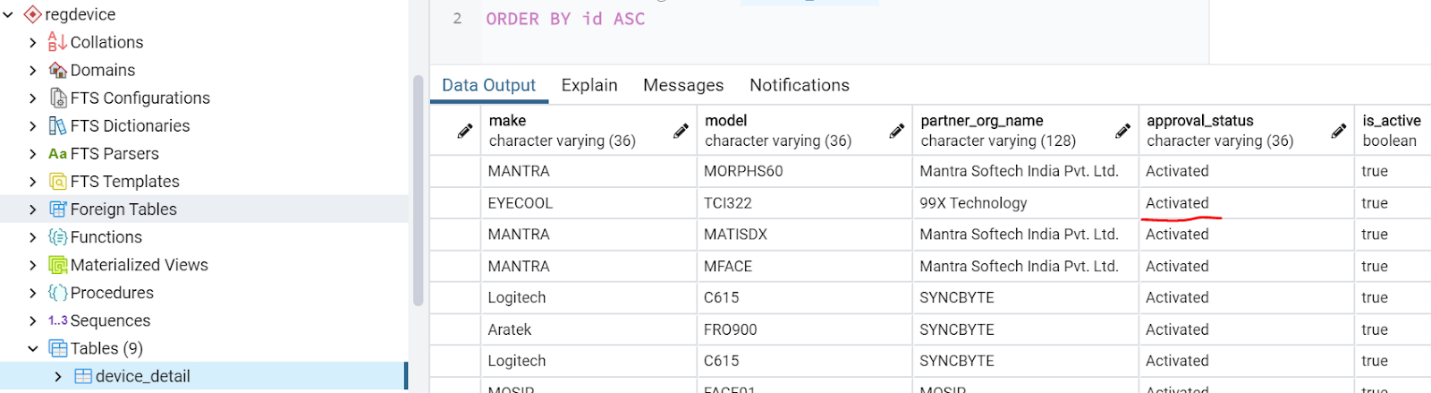
    "responsetime": "2021-03-11T04:15:28.710Z",

    "metadata": **null**,

    "response": "Device details approved successfully.",

    "errors": **null**

}



**6. Service to save SecureBiometricInterfaceCreateDto [POST]**

<https://aws.digitalid.lgcc.gov.lk/partnermanagement/v1/partners/securebiometricinterface>

Give the deviceDetailId param similarly as the "id": "242425" from the previous request above in Step 5 . Once the request is made the device is secured through SBI. details are stored in **mosip\_regdevice.secure\_biometric\_interface** table with **approval\_status** set to *Pending\_Approval* **..**

Returns a id for the secure device from above step.

**Ie: d2479cf1-4c63-420e-9480-105e744caa99**

**7. Login with Partner\_Admin [POST]**

<https://aws.digitalid.lgcc.gov.lk/v1/authmanager/authenticate/useridPwd>

Logs in as a **Partner\_Admin**  to approve above secured device.

**8. Approve/reject SecureBiometricInterface [PATCH]**

<https://aws.digitalid.lgcc.gov.lk/partnermanagement/v1/partners/securebiometricinterface>

Logged in **Partner\_Admin**  approves above secured device. . **"id"** param here is the id value returned in step 6 request (Service to save SecureBiometricInterfaceCreateDto)

**9. Login with GlobalAdmin in Admin [POST]**[**https://aws.digitalid.lgcc.gov.lk//v1/authmanager/authenticate/useridPwd**](https://aws.digitalid.lgcc.gov.lk/v1/authmanager/authenticate/useridPwd)

Logs in as a **GlobalAdmin** to register the device in the Mosip.

**10. Create Device in Mosip MasterDB [POST]**[**https://aws.digitalid.lgcc.gov.lk/v1/masterdata/devices**](https://aws.digitalid.lgcc.gov.lk/v1/masterdata/devices)

Persists the device spec details to masterDB of Mosip. Kind of whitelisting our device...

deviceSpecId param value comes from the **mosip\_master.device\_spec** tables id column.  
  
**11. Sign the registered device using Partner [POST]**[**https://aws.digitalid.lgcc.gov.lk/partnermanagement/v1/partners/registereddevices**](https://aws.digitalid.lgcc.gov.lk/partnermanagement/v1/partners/registereddevices)

This is the step of officially registering our device into MOSIP. Here the **deviceData** param value is generated using 2-3 level encryption.. Below is the process of generating the value for the **deviceData** parameter. (this should be done under 5 mins - is configurable..)

1. Populate the following body with the correct details

{

“serialNo” : “ ”,

“deviceProvider” : “ ”,

“deviceProviderId” : “ ”,

“make” : “ ”,

“model” : “ ”,

“dateTime” : “ ”,

“type” : “ ”,

“deviceSubType” : “ ”,

}

1. Encode the above using base64 **(1st Encoding)** (any preferred online tool - <https://jwt.io/>)
2. Get the encoded details and create the body as follow :

{

“deviceSubId”:[“1”,”2”],

“Certification” : “L0”,

“digitalId: “<encoded output from the 1st encoding done above>”,

“Firmware”: ” ”,

“deviceExpiry”: ” ”,

“Timestamp” : “ “

}

1. Encode the above body again with base 64.**(2nd Encoding)**
2. Get the encoded value and create another body again as follow :

{

“deviceId” : “ ”,

“purpose” : “ ”,

“deviceInfo” : “ < 2nd encoded output>”

}

1. Encode the above body again with base 64 **(3rd Encoding)**
2. Get the encoded value and create the final body:

{

“Id” : “string”,

“metadata” : { },

“request”  :  {

“deviceData” : “ <3rd encoded detail>”

        },

“requesttime” : “ ”,

“version” : “ ”

}

**12. Validate the Device added [POST]**

[**https://aws.digitalid.lgcc.gov.lk/partnermanagement/v1/partners/deviceprovidermanagement/validate**](https://aws.digitalid.lgcc.gov.lk/partnermanagement/v1/partners/deviceprovidermanagement/validate)

Confirms if the above device was registered successfully.