Sewerage Service - Technical Document

Sewerage Services

This is one of the major application of the egov stack which helps municipal and citizens to handle sewerage services like creating sewerage connection , searching sewerage connection, sewerage connection , also creating property if it doesnt exist and creating and updating workflow where depending on different roles of the logged-in user he/she can perform various actions like editing or perform document verification and many more specified in workflow documentation and SMS & email notification sent to owner for various actions performed in workflow.
The setup of the Application is as follows

Requirements

- Prior Knowledge of Java/J2EE (Java 8 version preferrable).
- Prior Knowledge of Spring Boot.
- Prior Knowledge of REST APIs and related concepts like path parameters, headers, JSON etc.
- Prior knowledge of Git.
- Prior knowledge of eGov-mdms service, eGov-persister, eGov-user, eGov-location, eGov-localization, eGov-idgen, eGov-workflow-service will be helpful.

FUNCTIONALITY

The sewerage service provides multiple functionality starting from serving as a central repository where one can create sewerage connection, update sewerage connection, search for a particular sewerage connection based on different criterias like mobile number, sewerage connection nos etc. and also creating property if it doesnt exist ,creating and updating workflow where depending on different roles of the logged-in user he/she can perform various actions like editing or perform document verification and many more specified in workflow documentation and SMS & email notification sent to owner for various actions performed in workflow.The different services provided by the sewerage services are

- Creating sewerage connection
- Updating sewerage connection
- Apply for Property creation.
- Searching sewerage connection based on few criterias mentioned in swagger documentation later
- Creating property if it doesnt exist
- Creating and updating workflow.

SETUP AND USAGE

The Application is present among the municipal group of applications available in the eGov-services git repository. The spring boot application needs **lombok*** extension added in your IDE to load it. Once the application is up and running API requests can be posted to the url and ids can be generated.
in case of intellij the plugin can be installed directly, for eclipse the lombok jar location has to be added in eclipse.ini file in this format javaagent:lombok.jar

API Information

Please refer Swagger API for YAML file details. Link -https://github.com/egovernments/municipal-services/blob/RAIN-2317/docs/water-sewerage-services.yaml

The variables in the Application.properties to be updated are as follows

kafka topics persister configs for eGov persister

egov.sewarageservice.createconnection=save-sw-connection
egov.sewarageservice.updateconnection=update-sw-connection
egov.seweragesservice.updateseweragesconnection.workflow.topic=update-sw-workflow

URLs for the external API references

eGvo mdms :-> egov.mdms.host = https://egov-micro-dev.egovernments.org
eGov -idGen :-> egov.idgen.host = https://egov-micro-dev.egovernments.org/
localization service :-> egov.localization.host = https://egov-micro-dev.egovernments.org
workflow-service:-> egov.workflow.host = https://egov-micro-dev.egovernments.org/
Url-Shortner
egov.url.shortner.host=http://egov-url-shortening.egov:8080
idGen Id formats :->
egov.idgen.scid.name=sewerageservice.connection.id
egov.idgen.scid.format=SW/[CITY.CODE]/[fy:yyyy-yy]/[SEQ_EGOV_COMMON]
egov.idgen.scapid.name=sewerageservice.application.id
egov.idgen.scapid.format=SW_AP/[CITY.CODE]/[fy:yyyy-yy]/[SEQ_EGOV_COMMON]

Configuration

Mdms configuration

https://github.com/egovernments/egov-mdms-data/tree/master/data - Connect to preview

```
sw-services-calculation
PropertyTax
```

master-config.json for sewerage service

```
"sw-services-calculation": {
  "Interest": {
    "masterName": "Interest",
    "isStateLevel": true,
    "uniqueKeys": [
      "$.fromFY"
    ]
  },
  "Rebate": {
    "masterName": "Rebate",
    "isStateLevel": true,
    "uniqueKeys": [
      "$.fromFY"
    ]
  },
  "Penalty": {
    "masterName": "Penalty",
    "isStateLevel": true,
    "uniqueKeys": [
      "$.fromFY"
    ]
  },
  "SCBillingSlab": {
    "masterName": "SCBillingSlab",
    "isStateLevel": true,
    "uniqueKeys": []
```

```
  },
  "billingPeriod": {
    "masterName": "billingPeriod",
    "isStateLevel": true,
    "uniqueKeys": [
      "$.billingCycle"
    ]
  },
  "SW_CHARGE": {
    "masterName": "SW_CHARGE",
    "isStateLevel": true,
    "uniqueKeys": []
  },
  "SW_TIME_PENALTY": {
    "masterName": "SW_TIME_PENALTY",
    "isStateLevel": true,
    "uniqueKeys": []
  },
  "SW_Round_Off": {
    "masterName": "SW_Round_Off",
    "isStateLevel": true,
    "uniqueKeys": []
  },
  "PlotSizeSlab": {
    "masterName": "PlotSizeSlab",
    "isStateLevel": true,
    "uniqueKeys": []
  },
  "PropertyUsageType": {
    "masterName": "PropertyUsageType",
    "isStateLevel": true,
    "uniqueKeys": []
  },
  "FeeSlab": {
    "masterName": "FeeSlab",
    "isStateLevel": true,
    "uniqueKeys": []
  },
  "RoadType": {
    "masterName": "RoadType",
    "isStateLevel": true,
    "uniqueKeys": []
  },
  "CalculationAttribute": {
    "masterName": "CalculationAttribute",
    "isStateLevel": true,
    "uniqueKeys": []
  }
}
```

Property creation through WNS module
https://github.com/egovernments/egov-mdms-data/blob/DEV/data/pb/PropertyTax/PTWorkflow.json

***Persister configuration*:**
https://github.com/egovernments/configs/blob/qa/egov-persister/sewerage-persist.yml

***Postman link*:**
https://www.getpostman.com/collections/e4c39fd6c0ed0a7dfacd

***Workflow business service config*:**

```
{
 "BusinessServices": [
  {
   "tenantId": "pb",
   "businessService": "NewSW1",
   "business": "sw-services",
   "businessServiceSla": 259200000,
   "states": [
    {
      "sla": null,
      "state": null,
      "applicationStatus": null,
      "docUploadRequired": false,
      "isStartState": true,
      "isTerminateState": false,
      "isStateUpdatable": false,
      "actions": [
       {
         "action": "INITIATE",
         "nextState": "INITIATED",
         "roles": [
           "CITIZEN",
           "SW_CEMP"
         ]
       }
      ]
    },
    {
      "sla": null,
      "state": "INITIATED",
      "applicationStatus": "INITIATED",
      "docUploadRequired": false,
      "isStartState": false,
      "isTerminateState": false,
      "isStateUpdatable": true,
      "actions": [
       {
         "action": "SUBMIT_APPLICATION",
         "nextState": "PENDING_FOR_DOCUMENT_VERIFICATION",
         "roles": [
           "CITIZEN",
           "SW_CEMP"
         ]
       }
      ]
    },
```

```
    {
      "sla": null,
      "state": "PENDING_FOR_CITIZEN_ACTION",
      "applicationStatus": "PENDING_FOR_CITIZEN_ACTION",
      "docUploadRequired": false,
      "isStartState": false,
      "isTerminateState": false,
      "isStateUpdatable": true,
      "actions": [
        {
          "action": "RESUBMIT_APPLICATION",
          "nextState": "PENDING_FOR_DOCUMENT_VERIFICATION",
          "roles": [
            "CITIZEN",
            "SW_CEMP"
          ]
        }
      ]
    },
    {
      "sla": null,
      "state": "PENDING_FOR_DOCUMENT_VERIFICATION",
      "applicationStatus": "PENDING_FOR_DOCUMENT_VERIFICATION",
      "docUploadRequired": false,
      "isStartState": false,
      "isTerminateState": false,
      "isStateUpdatable": true,
      "actions": [
        {
          "action": "VERIFY_AND_FORWARD",
          "nextState": "PENDING_FOR_FIELD_INSPECTION",
          "roles": [
            "SW_DOC_VERIFIER"
          ]
        },
        {
          "action": "REJECT",
          "nextState": "REJECTED",
          "roles": [
            "SW_DOC_VERIFIER"
          ]
        },
        {
          "action": "SEND_BACK_TO_CITIZEN",
          "nextState": "PENDING_FOR_CITIZEN_ACTION",
          "roles": [
            "SW_DOC_VERIFIER"
          ]
        }
      ]
    },
    {
      "sla": null,
```

```
      "state": "REJECTED",
      "applicationStatus": "REJECTED",
      "isStateUpdatable": false,
      "docUploadRequired": false,
      "isStartState": false,
      "isTerminateState": true
    },
    {
      "sla": 86400000,
      "state": "PENDING_FOR_FIELD_INSPECTION",
      "applicationStatus": "PENDING_FOR_FIELD_INSPECTION",
      "docUploadRequired": false,
      "isStartState": false,
      "isStateUpdatable": true,
      "isTerminateState": false,
      "actions": [
        {
          "action": "VERIFY_AND_FORWARD",
          "nextState": "PENDING_APPROVAL_FOR_CONNECTION",
          "roles": [
            "SW_FIELD_INSPECTOR"
          ]
        },
        {
          "action": "REJECT",
          "nextState": "REJECTED",
          "roles": [
            "SW_FIELD_INSPECTOR"
          ]
        },
        {
          "action": "SEND_BACK_FOR_DOCUMENT_VERIFICATION",
          "nextState": "PENDING_FOR_DOCUMENT_VERIFICATION",
          "roles": [
            "SW_FIELD_INSPECTOR"
          ]
        }
      ]
    },
    {
      "sla": 43200000,
      "state": "PENDING_APPROVAL_FOR_CONNECTION",
      "applicationStatus": "PENDING_APPROVAL_FOR_CONNECTION",
      "docUploadRequired": false,
      "isStartState": false,
      "isStateUpdatable": true,
      "isTerminateState": false,
      "actions": [
        {
          "action": "APPROVE_FOR_CONNECTION",
          "nextState": "PENDING_FOR_PAYMENT",
          "roles": [
            "SW_APPROVER"
```

```
          ]
        },
        {
          "action": "REJECT",
          "nextState": "REJECTED",
          "roles": [
            "SW_APPROVER"
          ]
        },
        {
          "action": "SEND_BACK_FOR_FIELD_INSPECTION",
          "nextState": "PENDING_FOR_FIELD_INSPECTION",
          "roles": [
            "SW_APPROVER"
          ]
        }
      ]
    },
    {
      "sla": 43200000,
      "state": "PENDING_FOR_PAYMENT",
      "applicationStatus": "PENDING_FOR_PAYMENT",
      "docUploadRequired": false,
      "isStartState": false,
      "isTerminateState": false,
      "isStateUpdatable": false,
      "actions": [
        {
          "action": "PAY",
          "nextState": "PENDING_FOR_CONNECTION_ACTIVATION",
          "roles": [
            "CITIZEN",
            "SW_CEMP"
          ]
        }
      ]
    },
    {
      "sla": null,
      "state": "PENDING_FOR_CONNECTION_ACTIVATION",
      "applicationStatus": "PENDING_FOR_CONNECTION_ACTIVATION",
      "isStateUpdatable": true,
      "docUploadRequired": false,
      "isStartState": false,
      "isTerminateState": false,
      "actions": [
        {
          "action": "ACTIVATE_CONNECTION",
          "nextState": "CONNECTION_ACTIVATED",
          "roles": [
            "SW_CLERK"
          ]
        }
```

```
        ]
      },
      {
        "sla": null,
        "state": "CONNECTION_ACTIVATED",
        "applicationStatus": "CONNECTION_ACTIVATED",
        "isStateUpdatable": false,
        "docUploadRequired": false,
        "isStartState": false,
        "isTerminateState": true
      }
    ]
  }
 ]
}
```

Workflow for property creation through Water and Sewerage Module

```
{
  "BusinessServices": [
   {
    "tenantId": "pb",
    "businessService": "NewWS1",
    "business": "ws-services",
    "businessServiceSla": 259200000,
    "states": [
     {
        "sla": null,
        "state": null,
        "applicationStatus": "INWORKFLOW",
        "docUploadRequired": false,
        "isStartState": true,
        "isTerminateState": false,
        "isStateUpdatable": false,
        "actions": [
         {
          "action": "OPEN",
          "nextState": "INITIATED",
          "roles": [
           "CITIZEN",
           "WS_CEMP"
          ]
         }
        ]
      },
      {
        "sla": null,
        "state": "INITIATED",
        "applicationStatus": "INWORKFLOW",
        "docUploadRequired": false,
        "isStartState": true,
        "isTerminateState": false,
        "isStateUpdatable": true,
        "actions": [
```

```
        {
          "action": "SUBMIT",
          "nextState": "APPROVED",
          "roles": [
            "CITIZEN",
            "WS_CEMP"
          ]
        }
      ]
    },
    {
      "sla": null,
      "state": "APPROVED",
      "applicationStatus": "ACTIVE",
      "docUploadRequired": false,
      "isStartState": false,
      "isTerminateState": true,
      "isStateUpdatable": false,
      "actions": null
    }
  ]
 }
]
}
```

Indexer config for sewerage-service

The indexer provides the facility for indexing the data to elastic search.

Setup

Write the configuration for sewerage service. The structure of the config file is explained later in the same doc.

Provide the absolute path of the checked-in file to DevOps, to add it to the file-read path of egov-indexer. The file will be added to the egov-indexer's environment manifest file for it to be read at the start-up of the application.

Put indexer config file to the config repo under egov-indexer folder.(egovernments/configs)

Run the egov-indexer app, Since it is a consumer, it starts listening to the configured topics and indexes the data.

config Keys

The indexer uses a config file per module to store all the configurations pertaining to that module. Indexer reads multiple such files at start-up to support indexing for all the configured modules. The water service file contains the following keys:

a. **serviceName**: Name of the module to which this configuration belongs.

b. **summary**: Summary of the module.

c. **version**: The version of the configuration.

d. **mappings**: List of definitions within the module. Every definition corresponds to one index requirement. Which means, every object received onto the Kafka queue can be used to create multiple indexes, each of these indexes will need configuration, all such configurations belonging to one topic forms one entry in the mappings list. The keys listed henceforth together form one definition and multiple such definitions are part of this mappings key.

i. **topic**: Topic on which the data is to be received to activate this particular configuration.

ii. **configKey**: Key to identify to what type of job is this config for. values: INDEX, REINDEX, LEGACYINDEX. INDEX: LiveIndex, REINDEX: Reindex, LEGACYINDEX: LegacyIndex.
iii. **indexes**: Key to configure multiple index configurations for the data received on the particular topic. Multiple indexes based on different requirement can be created using the same object. This list of such configurations is a part of this key. uses the following keys:
name: Index name on the elasticsearch. (Index will be created if it doesn't exist with this name.)
type: Document type within that index to which the index json has to go. (Elasticsearch uses the structure of index/type/docId to locate any file within index/type with id = docId)
id: Takes comma separated JsonPaths. The JSONPath is applied on the record received on the queue, the values hence obtained are appended and used as id for the record.
jsonPath: Key to be used in case of indexing a part of the input JSON and in case of indexing a custom json where the values for custom json are to be fetched from this part of the input.
timeStampField: JSONPath of the field in the input which can be used to obtain the timestamp of the input.
i) **indexMapping**: A skeleton/mapping of the JSON that is to be indexed. Note that, this JSON must always contain a key called "Data" at the top-level and the custom mapping begins within this key. This is only a convention to smoothen dashboarding on Kibana when data from multiple indexes have to be fetched for a single dashboard.
ii) **fieldMapping**: Contains a list of configurations. Each configuration contains keys to identify the field of the input JSON that has to be mapped to the fields of the index json which is mentioned in the key 'indexMapping' in the config. Has the following keys:
inJsonPath: JSONPath of the field from the input.
outJsonPath: JSONPath of the field of the index json.
iii) **externalUriMapping**: Contains a list of configurations. Each configuration contains keys to identify the field of the input JSON that are to be enriched using APIs from the external services. The configuration for those APIs also is a part of this. Uses the following keys:
path: URI of the API to be used. (it should be POST/_search API.)
queryParam: Configruation of the query params to be used for the API call. It is a comma seperated key-value pair, where key is the parameter name as per the API contract and value is the JSONPath of the field to be equated against this paramter.
apiRequest: Request Body of the API. (Since we only use _search APIs, it should be only RequestInfo.)
uriResponseMapping: Contains a list of configuration. Each configuration contains two keys: One is a JSONPath to identify the field from response, Second is also a JSONPath to map the response field to a field of the index json mentioned in the key 'indexMapping'.
i) **inJsonPath**: JSONPath to identify the field from response
ii) **outJsonPath**: JSONPath to map the response field to a field of the index json

***sewerage-service* indexer config**

```
ServiceMaps:
 serviceName: Sewerage Service - rainmaker
 version: 1.0.0
 mappings:
  - topic: save-sw-connection
    configKey: INDEX
    indexes:
      - name: sewerage-services
        type: general
        id: $.id,$.tenantId
        timeStampField: $.auditDetails.createdTime
```

```
      jsonPath: $.SewerageConnection
      customJsonMapping:
        indexMapping: {"Data":{"workflow": {"state": {}, "action": "", "assignes":
[]},"applicationNo":"","applicationStatus":"","status":"","connectionNo":"","oldConnectionNo":"","plumberInfo":
[],"roadCuttingInfo":[],"connectionHolders":[],"roadType":"","roadCuttingArea":"","connectionExecutionDate":"
","connectionCategory":"","connectionType":"","additionalDetails":{},"id":"","propertyId":"","tenantId":"","propos
edWaterClosets":"","proposedToilets":"","noOfWaterClosets":"","noOfToilets":"","applicationType":"","dateEff
ectiveFrom":"","history":{}}}
        fieldMapping:
        - inJsonPath: $.applicationStatus.state
          outJsonPath: $.Data.workflow.state
        - inJsonPath: $.processInstance.action
          outJsonPath: $.Data.workflow.action
        - inJsonPath: $.processInstance.assignes.*.uuid
          outJsonPath: $.Data.workflow.assignes
        - inJsonPath: $.applicationNo
          outJsonPath: $.Data.applicationNo
        - inJsonPath: $.applicationStatus
          outJsonPath: $.Data.applicationStatus
        - inJsonPath: $.status
          outJsonPath: $.Data.status
        - inJsonPath: $.connectionNo
          outJsonPath: $.Data.connectionNo
        - inJsonPath: $.oldConnectionNo
          outJsonPath: $.Data.oldConnectionNo
        - inJsonPath: $.plumberInfo
          outJsonPath: $.Data.plumberInfo
        - inJsonPath: $.roadCuttingInfo
          outJsonPath: $.Data.roadCuttingInfo
        - inJsonPath: $.connectionHolders
          outJsonPath: $.Data.connectionHolders
        - inJsonPath: $.roadType
          outJsonPath: $.Data.roadType
        - inJsonPath: $.roadCuttingArea
          outJsonPath: $.Data.roadCuttingArea
        - inJsonPath: $.connectionExecutionDate
          outJsonPath: $.Data.connectionExecutionDate
        - inJsonPath: $.connectionCategory
          outJsonPath: $.Data.connectionCategory
        - inJsonPath: $.connectionType
          outJsonPath: $.Data.connectionType
        - inJsonPath: $.additionalDetails
          outJsonPath: $.Data.additionalDetails
        - inJsonPath: $.id
          outJsonPath: $.Data.id
        - inJsonPath: $.propertyId
          outJsonPath: $.Data.propertyId
        - inJsonPath: $.tenantId
          outJsonPath: $.Data.tenantId
        - inJsonPath: $.proposedWaterClosets
          outJsonPath: $.Data.proposedWaterClosets
        - inJsonPath: $.proposedToilets
          outJsonPath: $.Data.proposedToilets
```

```
      - inJsonPath: $.noOfWaterClosets
        outJsonPath: $.Data.noOfWaterClosets
      - inJsonPath: $.noOfToilets
        outJsonPath: $.Data.noOfToilets
      - inJsonPath: $.applicationType
        outJsonPath: $.Data.applicationType
      - inJsonPath: $.dateEffectiveFrom
        outJsonPath: $.Data.dateEffectiveFrom
      externalUriMapping:
      - path: http://egov-workflow-v2.egov:8080/egov-workflow-v2/egov-wf/process/_search
        queryParam: businessIds=$.applicationNo,history=true,tenantId=$.tenantId
        apiRequest:
{"RequestInfo":{"apiId":"org.egov.pt","ver":"1.0","ts":1502890899493,"action":"asd","did":"4354648646","key":"xyz","msgId":"654654","requesterId":"61","authToken":"d9994555-7656-4a67-ab3a-a952a0d4dfc8","userInfo":{"id":1,"uuid":"1fec8102-0e02-4d0a-b283-cd80d5dab067","type":"EMPLOYEE","tenantId":"pb.amritsar","roles":[{"name":"Employee","code":"EMPLOYEE","tenantId":"pb.amritsar"}]}}}
        uriResponseMapping:
        - inJsonPath: $.ProcessInstances
          outJsonPath: $.Data.history

- topic: update-sw-connection
  configKey: INDEX
  indexes:
    - name: sewerage-services
      type: general
      id: $.id,$.tenantId
      timeStampField: $.auditDetails.lastModifiedTime
      jsonPath: $.SewerageConnection
      customJsonMapping:
        indexMapping: {"Data":{"workflow": {"state": {}, "action": "", "assignes": []},"applicationNo":"","applicationStatus":"","status":"","connectionNo":"","oldConnectionNo":"","plumberInfo":[],"roadCuttingInfo":[],"connectionHolders":[],"roadType":"","roadCuttingArea":"","connectionExecutionDate":"","connectionCategory":"","connectionType":"","additionalDetails":{},"id":"","propertyId":"","tenantId":"","proposedWaterClosets":"","proposedToilets":"","noOfWaterClosets":"","noOfToilets":"","applicationType":"","dateEffectiveFrom":"","history":{}}}
        fieldMapping:
        - inJsonPath: $.applicationStatus.state
          outJsonPath: $.Data.workflow.state
        - inJsonPath: $.processInstance.action
          outJsonPath: $.Data.workflow.action
        - inJsonPath: $.processInstance.assignes.*.uuid
          outJsonPath: $.Data.workflow.assignes
        - inJsonPath: $.applicationNo
          outJsonPath: $.Data.applicationNo
        - inJsonPath: $.applicationStatus
          outJsonPath: $.Data.applicationStatus
        - inJsonPath: $.status
          outJsonPath: $.Data.status
        - inJsonPath: $.connectionNo
          outJsonPath: $.Data.connectionNo
        - inJsonPath: $.oldConnectionNo
          outJsonPath: $.Data.oldConnectionNo
```

```
      - inJsonPath: $.plumberInfo
        outJsonPath: $.Data.plumberInfo
      - inJsonPath: $.roadCuttingInfo
        outJsonPath: $.Data.roadCuttingInfo
      - inJsonPath: $.connectionHolders
        outJsonPath: $.Data.connectionHolders
      - inJsonPath: $.roadType
        outJsonPath: $.Data.roadType
      - inJsonPath: $.roadCuttingArea
        outJsonPath: $.Data.roadCuttingArea
      - inJsonPath: $.connectionExecutionDate
        outJsonPath: $.Data.connectionExecutionDate
      - inJsonPath: $.connectionCategory
        outJsonPath: $.Data.connectionCategory
      - inJsonPath: $.connectionType
        outJsonPath: $.Data.connectionType
      - inJsonPath: $.additionalDetails
        outJsonPath: $.Data.additionalDetails
      - inJsonPath: $.id
        outJsonPath: $.Data.id
      - inJsonPath: $.propertyId
        outJsonPath: $.Data.propertyId
      - inJsonPath: $.tenantId
        outJsonPath: $.Data.tenantId
      - inJsonPath: $.proposedWaterClosets
        outJsonPath: $.Data.proposedWaterClosets
      - inJsonPath: $.proposedToilets
        outJsonPath: $.Data.proposedToilets
      - inJsonPath: $.noOfWaterClosets
        outJsonPath: $.Data.noOfWaterClosets
      - inJsonPath: $.noOfToilets
        outJsonPath: $.Data.noOfToilets
      - inJsonPath: $.applicationType
        outJsonPath: $.Data.applicationType
      - inJsonPath: $.dateEffectiveFrom
        outJsonPath: $.Data.dateEffectiveFrom
      externalUriMapping:
      - path: http://egov-workflow-v2.egov:8080/egov-workflow-v2/egov-wf/process/_search
        queryParam: businessIds=$.applicationNo,history=true,tenantId=$.tenantId
        apiRequest:
{"RequestInfo":{"apiId":"org.egov.pt","ver":"1.0","ts":1502890899493,"action":"asd","did":"4354648646","key":
"xyz","msgId":"654654","requesterId":"61","authToken":"d9994555-7656-4a67-ab3a-
a952a0d4dfc8","userInfo":{"id":1,"uuid":"1fec8102-0e02-4d0a-b283-
cd80d5dab067","type":"EMPLOYEE","tenantId":"pb.amritsar","roles":[{"name":"Employee","code":"EMPLOYE
E","tenantId":"pb.amritsar"}]}}}
        uriResponseMapping:
        - inJsonPath: $.ProcessInstances
          outJsonPath: $.Data.history


- topic: update-sw-workflow
  configKey: INDEX
  indexes:
```

```
  - name: sewerage-services
    type: general
    id: $.id,$.tenantId
    timeStampField: $.auditDetails.lastModifiedTime
    jsonPath: $.SewerageConnection
    customJsonMapping:
      indexMapping: {"Data":{"workflow": {"state": {}, "action": "", "assignes":
[]},"applicationNo":"","applicationStatus":"","status":"","connectionNo":"","oldConnectionNo":"","plumberInfo":
[],"roadCuttingInfo":[],"connectionHolders":[],"roadType":"","roadCuttingArea":"","connectionExecutionDate":"
","connectionCategory":"","connectionType":"","additionalDetails":{},"id":"","propertyId":"","tenantId":"","propos
edWaterClosets":"","proposedToilets":"","noOfWaterClosets":"","noOfToilets":"","applicationType":"","dateEff
ectiveFrom":"","history":{}}}
      fieldMapping:
      - inJsonPath: $.applicationStatus.state
        outJsonPath: $.Data.workflow.state
      - inJsonPath: $.processInstance.action
        outJsonPath: $.Data.workflow.action
      - inJsonPath: $.processInstance.assignes.*.uuid
        outJsonPath: $.Data.workflow.assignes
      - inJsonPath: $.applicationNo
        outJsonPath: $.Data.applicationNo
      - inJsonPath: $.applicationStatus
        outJsonPath: $.Data.applicationStatus
      - inJsonPath: $.status
        outJsonPath: $.Data.status
      - inJsonPath: $.connectionNo
        outJsonPath: $.Data.connectionNo
      - inJsonPath: $.oldConnectionNo
        outJsonPath: $.Data.oldConnectionNo
      - inJsonPath: $.plumberInfo
        outJsonPath: $.Data.plumberInfo
      - inJsonPath: $.roadCuttingInfo
        outJsonPath: $.Data.roadCuttingInfo
      - inJsonPath: $.connectionHolders
        outJsonPath: $.Data.connectionHolders
      - inJsonPath: $.roadType
        outJsonPath: $.Data.roadType
      - inJsonPath: $.roadCuttingArea
        outJsonPath: $.Data.roadCuttingArea
      - inJsonPath: $.connectionExecutionDate
        outJsonPath: $.Data.connectionExecutionDate
      - inJsonPath: $.connectionCategory
        outJsonPath: $.Data.connectionCategory
      - inJsonPath: $.connectionType
        outJsonPath: $.Data.connectionType
      - inJsonPath: $.additionalDetails
        outJsonPath: $.Data.additionalDetails
      - inJsonPath: $.id
        outJsonPath: $.Data.id
      - inJsonPath: $.propertyId
        outJsonPath: $.Data.propertyId
      - inJsonPath: $.tenantId
        outJsonPath: $.Data.tenantId
```

```
      - inJsonPath: $.proposedWaterClosets
        outJsonPath: $.Data.proposedWaterClosets
      - inJsonPath: $.proposedToilets
        outJsonPath: $.Data.proposedToilets
      - inJsonPath: $.noOfWaterClosets
        outJsonPath: $.Data.noOfWaterClosets
      - inJsonPath: $.noOfToilets
        outJsonPath: $.Data.noOfToilets
      - inJsonPath: $.applicationType
        outJsonPath: $.Data.applicationType
      - inJsonPath: $.dateEffectiveFrom
        outJsonPath: $.Data.dateEffectiveFrom
      externalUriMapping:
      - path: http://egov-workflow-v2.egov:8080/egov-workflow-v2/egov-wf/process/_search
        queryParam: businessIds=$.applicationNo,history=true,tenantId=$.tenantId
        apiRequest:
{"RequestInfo":{"apiId":"org.egov.pt","ver":"1.0","ts":1502890899493,"action":"asd","did":"4354648646","key":"xyz","msgId":"654654","requesterId":"61","authToken":"d9994555-7656-4a67-ab3a-a952a0d4dfc8","userInfo":{"id":1,"uuid":"1fec8102-0e02-4d0a-b283-cd80d5dab067","type":"EMPLOYEE","tenantId":"pb.amritsar","roles":[{"name":"Employee","code":"EMPLOYEE","tenantId":"pb.amritsar"}]}}}
        uriResponseMapping:
        - inJsonPath: $.ProcessInstances
          outJsonPath: $.Data.history
```

Notifications

To enable or disable notifcation

notification.sms.enabled=true

egov.user.event.notification.enabled=true

Notification config:

notification.url = https://egov-micro-dev.egovernments.org/

kafka.topics.notification.sms=egov.core.notification.sms

notification.sms.link=citizen/egov-common/pay?consumerCode=$consumerCode&tenantId=$tenantId&businessService=SW

sw.mseva.app.link=https://play.google.com/store/apps/details?id=org.egovernment.mseva.citizen

sw.view.history.link=citizen/wns/search-preview?applicationNumber=$applicationNumber&history=true&tenantId=$tenantId&service=SEWERAGE

sw.connectiondetails.link=citizen/wns/connection-details?connectionNumber=$connectionNumber&tenantId=$tenantId&service=SEWERAGE

sw.application.pay.link=citizen/egov-common/pay?consumerCode=$consumerCode&tenantId=$tenantId&businessService=SW.ONE_TIME_FEE

The Current localization messages for notification

```
[
    {
        "code": "SW_INITIATE_INITIATED_SMS_MESSAGE",
        "message": "Dear <Owner Name>, You have successfully submitted your application for a New <Service> Connection. Your Application No. is <Application number>. Click here to download your
```

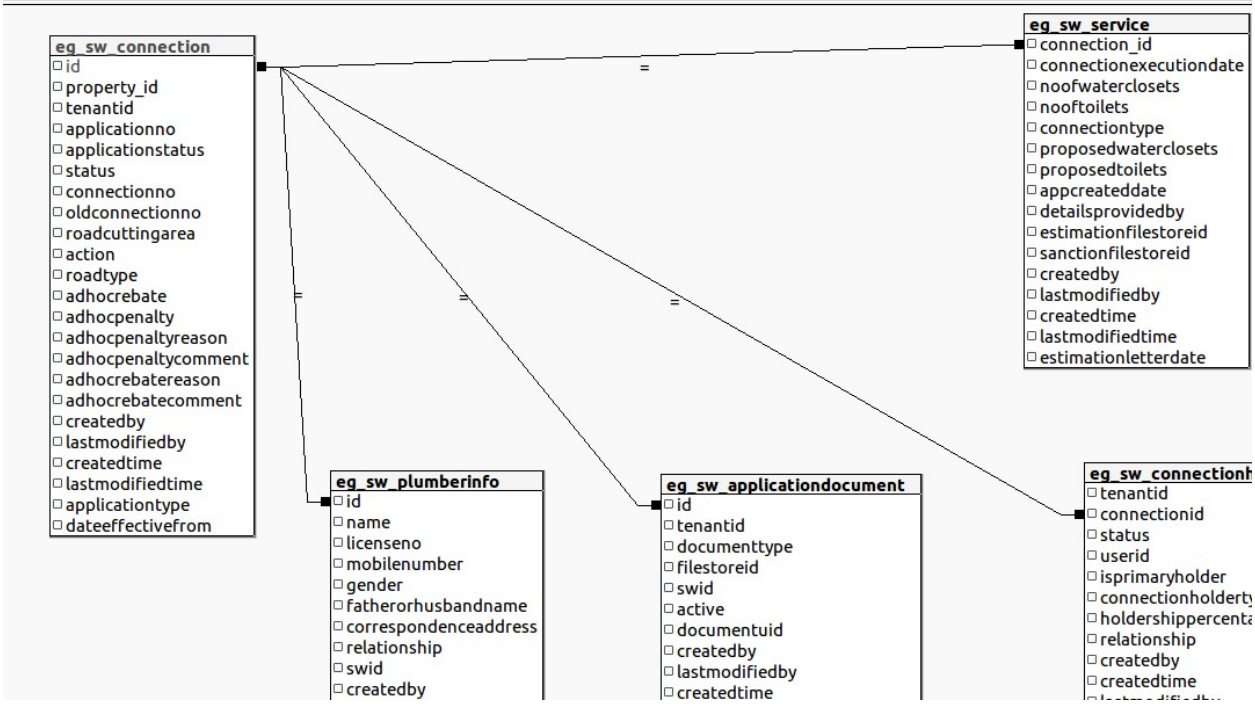
```
application <Application download link>. For more information, please log in to <mseva URL> or download <mseva app link>.",
        "module": "rainmaker-ws",
        "locale": "en_IN"
    },
    {
        "code": "SW_INITIATE_INITIATED_APP_MESSAGE",
        "message": "Dear <Owner Name>, You have successfully submitted your application for a New <Service> Connection. Your Application No. is <Application number> . <Action Button>Download Application</Action Button>",
        "module": "rainmaker-ws",
        "locale": "en_IN"
    },
    {
        "code": "SW_REJECT_REJECTED_SMS_MESSAGE",
        "message": "Dear <Owner Name>, Your Application <Application number>  for a New <Service> Connection has been rejected. For more details, please log in to <mseva URL> or download <mseva app link>.",
        "module": "rainmaker-ws",
        "locale": "en_IN"
    },
    {
        "code": "SW_REJECT_REJECTED_APP_MESSAGE",
        "message": "Dear <Owner Name>, Your Application <Application number>  for a New <Service> Connection has been rejected. Click here for more details <View History Link>",
        "module": "rainmaker-ws",
        "locale": "en_IN"
    },
    {
        "code": "SW_EDIT_SMS_MESSAGE",
        "message": "Dear <Owner Name>, Your Application <Application number>  for a New <Service> Connection has been edited. For more details, please log in to <mseva URL> or download <mseva app link>.",
        "module": "rainmaker-ws",
        "locale": "en_IN"
    },
    {
        "code": "SW_EDIT_IN_APP_MESSAGE",
        "message": "Dear <Owner Name>, Your Application <Application number>  for a New <Service> Connection has been edited. Click here for more details <View History Link>.",
        "module": "rainmaker-ws",
        "locale": "en_IN"
    },
    {
        "code": "SW_VERIFY_AND_FORWARD_PENDING_FOR_FIELD_INSPECTION_SMS_MESSAGE",
        "message": "Dear <Owner Name>, Status of your application <Application number> for a New <Service> Connection has changed to PENDING FOR FIELD INSPECTION from PENDING FOR DOCUMENT VERIFICATION. For more details, please log in to <mseva URL> or download <mseva app link>.",
        "module": "rainmaker-ws",
        "locale": "en_IN"
    },
    {
        "code": "SW_VERIFY_AND_FORWARD_PENDING_FOR_FIELD_INSPECTION_APP_MESSAGE",
```

```
46          "message": "Dear <Owner Name>, Status of your application <Application number> for a New <Service> Connection has changed to PENDING FOR FIELD INSPECTION from PENDING FOR DOCUMENT VERIFICATION. To track your application, click on <View History Link>.",
47          "module": "rainmaker-ws",
48          "locale": "en_IN"
49      },
50      {
51          "code": "SW_SEND_BACK_FOR_FIELD_INSPECTION_PENDING_FOR_FIELD_INSPECTION_SMS_MESSAGE",
52          "message": "Dear <Owner Name>, Your Application <Application number>  for a New <Service> Connection has been sent back. For more details, please log in to <mseva URL> or download <mseva app link>.",
53          "module": "rainmaker-ws",
54          "locale": "en_IN"
55      },
56      {
57          "code": "SW_SEND_BACK_FOR_FIELD_INSPECTION_PENDING_FOR_FIELD_INSPECTION_APP_MESSAGE",
58          "message": "Dear <Owner Name>, Your Application <Application number>  for a New <Service> Connection has been sent back. Click here for more details <View History Link>",
59          "module": "rainmaker-ws",
60          "locale": "en_IN"
61      },
62      {
63          "code": "SW_SEND_BACK_FOR_DOCUMENT_VERIFICATION_PENDING_FOR_DOCUMENT_VERIFICATION_SMS_MESSAG",
64          "message": "Dear <Owner Name>, Your Application <Application number>  for a New <Service> Connection has been sent back. For more details, please log in to <mseva URL> or download <mseva app link>.",
65          "module": "rainmaker-ws",
66          "locale": "en_IN"
67      },
68      {
69          "code": "SW_SEND_BACK_FOR_DOCUMENT_VERIFICATION_PENDING_FOR_DOCUMENT_VERIFICATION_APP_MESSAGE",
70          "message": "Dear <Owner Name>, Your Application <Application number>  for a New <Service> Connection has been sent back. Click here for more details <View History Link>",
71          "module": "rainmaker-ws",
72          "locale": "en_IN"
73      },
74      {
75          "code": "SEWERAGE_CONNECTION_BILL_GENERATION_MESSAGE",
76          "message": "Dear <Owner Name>,Your<Service>  Bill for <Billing Period> has been generated. Please pay the amount due: <bill amount> by due date <Due Date>. Following is the link to your bill: <Link to Bill>",
77          "module": "rainmaker-common",
78          "locale": "en_IN"
79      },
80      {
81          "code": "SEWERAGE_CONNECTION_BILL_GENERATION_SMS_MESSAGE",
```

```
        "message": "Dear <Owner Name>, Your <Service>  Bill has been generated. Please pay the amount due: <bill amount> by due date <Due Date>. Following is the link to your bill: <Link to Bill>",
        "module": "rainmaker-ws",
        "locale": "en_IN"
    },
    {
        "code": "SEWERAGE_CONNECTION_BILL_GENERATION_APP_MESSAGE",
        "message": "Dear <Owner Name>, Your <Service>  Bill has been generated. Please pay the amount due: <bill amount> by due date <Due Date>.",
        "module": "rainmaker-ws",
        "locale": "en_IN"
    },
     {
        "code": "SW_APPROVE_FOR_CONNECTION_PENDING_FOR_PAYMENT_SMS_MESSAGE",
        "message": "Dear <Owner Name>, Your New <Service> connection against the application <Application number> has been approved. To make payment against your application, please click on <payment link> . Log in to <mseva URL> or download <mseva app link> for more details.",
        "module": "rainmaker-ws",
        "locale": "en_IN"
    },
    {
        "code": "SW_APPROVE_FOR_CONNECTION_PENDING_FOR_PAYMENT_APP_MESSAGE",
         "message": "Dear <Owner Name>, Your New <Service> connection against the application <Application number> has been approved. To make payment against your application, please click on PAY NOW .<Action Button>Download Application</Action Button>",
         "module": "rainmaker-ws",
         "locale": "en_IN"
     },
     {
        "code": "SW_PAY_PENDING_FOR_CONNECTION_ACTIVATION_SMS_MESSAGE",
        "message": "Dear <Owner Name> , Your payment for New <Service> connection against the application <Application number> has been been succesfully recorded. You can download your receipt using this link <receipt download link>. For more details, please log in to <mseva URL> or download <mseva app link>.",
        "module": "rainmaker-ws",
        "locale": "en_IN"
    },
    {
        "code": "SW_PAY_PENDING_FOR_CONNECTION_ACTIVATION_APP_MESSAGE",
        "message": "Dear <Owner Name> , Your payment for New <Service> connection against the application <Application number> has been been succesfully recorded. You can download the receipt by clicking DOWNLOAD RECEIPT.<Action Button>Download Application</Action Button>",
        "module": "rainmaker-ws",
        "locale": "en_IN"
    },
    {
        "code": "SW_ACTIVATE_CONNECTION_CONNECTION_ACTIVATED_SMS_MESSAGE",
        "message": "Dear <Owner Name>, Your New <Service> connection against the application <Application number> has been activated. To check your connection details, please log in to <mseva URL> or download <mseva app link>",
        "module": "rainmaker-ws",
        "locale": "en_IN"
    },
```

```
122         {
123           "code": "SW_ACTIVATE_CONNECTION_CONNECTION_ACTIVATED_APP_MESSAGE",
124           "message": "Dear <Owner Name>, Your New <Service> connection against the application
<Application number> has been activated. To check your connection details, click here <connection
details page>.",
125           "module": "rainmaker-ws",
126           "locale": "en_IN"
127       }
128   ]
```

Table UML Diagram



Modify connection

After connection activation or legacy connection, we can edit the connection. This process based on defined workflow. Any action is based on defined roles on the action level. For edit connection, we need to upload some supporting documents and mandatory info.

Workflow config for edit connection

```
1{
2  "BusinessServices": [
3    {
4      "tenantId": "pb",
5      "businessService": "ModifySWConnection",
6      "business": "sw-services",
7      "businessServiceSla": 259200000,
8      "states": [
9        {
10           "sla": null,
11           "state": null,
12           "applicationStatus": null,
13           "docUploadRequired": false,
```

```
      "isStartState": true,
      "isTerminateState": false,
      "isStateUpdatable": false,
      "actions": [
       {
        "action": "INITIATE",
        "nextState": "INITIATED",
        "roles": [
         "SW_CEMP"
        ]
       }
      ]
     },
     {
      "sla": null,
      "state": "INITIATED",
      "applicationStatus": "INITIATED",
      "docUploadRequired": false,
      "isStartState": false,
      "isTerminateState": false,
      "isStateUpdatable": true,
      "actions": [
       {
        "action": "SUBMIT_APPLICATION",
        "nextState": "PENDING_FOR_APPROVAL",
        "roles": [
         "SW_CEMP"
        ]
       }
      ]
     },
     {
      "sla": 86400000,
      "state": "PENDING_FOR_APPROVAL",
      "applicationStatus": "PENDING_FOR_APPROVAL",
      "docUploadRequired": false,
      "isStartState": false,
      "isStateUpdatable": true,
      "isTerminateState": false,
      "actions": [
       {
        "action": "APPROVE_CONNECTION",
        "nextState": "APPROVED",
        "roles": [
         "SW_APPROVER"
        ]
       },
       {
        "action": "REJECT",
        "nextState": "REJECTED",
        "roles": [
         "SW_APPROVER"
        ]
```

```
        }
       ]
     },
     {
       "sla": null,
       "state": "REJECTED",
       "applicationStatus": "REJECTED",
       "isStateUpdatable": false,
       "docUploadRequired": false,
       "isStartState": false,
       "isTerminateState": true
     },
     {
       "sla": null,
       "state": "APPROVED",
       "applicationStatus": "APPROVED",
       "isStateUpdatable": false,
       "docUploadRequired": false,
       "isStartState": false,
       "isTerminateState": true
     }
   ]
  }
 ]
}
```

Notification

Notification will be sent to the property owners and connection holders based on different application states.

Capturing connection holders

We can add connection holders to the sewerage connection which will be the owner of the connection. We can fill the connection holders' details or we can just make the property owner to the connection holder.
The connection holder will get notification based on a different state of the application. We are pushing the data of the connection holders in the user service too.

Multiple Road Type Support

We can add road cutting details of multiple roads to the sewerage connection. For each road which goes under cutting process we have to fill their road type details and road cutting area.
Based on this information, application one time fee estimate is calculated.

Sewerage Calculator Service

This is one of the major business logic services which is used for calculation of sewerage charge, generating demand, update existing demand , SMS & email notification to the ULB officials on demand generation and also triggering demands(job scheduler) at some intervals and estimation of water charge(one-time cost) which involves cost like road-cutting charge , form fee , scrutiny fee etc.

Requirements:

- Knowledge of Java/J2EE(preferably Java 8 version)
- Knowledge of Spring Boot and spring-boot microservices.

- Knowledge of Git or any version control system.
- Knowledge of RESTful Web services.
- Knowledge of the Lombok library will helpful.
- knowledge of eGov-mdms service, eGov-persister, eGov-idgen, eGov-sms, eGov-email,eGov-user, eGov-localization, eGov-workflow-service will be helpful.

Functionality

Sewerage calculator services present in municipal services provides multiple functionalities like calculating sewerage charges, generating demands for a particular sewerage connection , updating demands, SMS & email notification to the ULB officials on demand generation and also triggering demands(job scheduler) at some intervals and estimation of water charge(one-time cost) which involves cost like road-cutting charge , form fee , scrutiny fee etc. The different functionalities provided by sewerage calculator services are:

Sewerage charge calculation

Demand generation (here as it is always non-metered demand will be generated based on time period)
Sewerage charge estimation (one-time cost which involves cost like road-cutting charge , form fee , scrutiny fee etc.)

Setup and usage

The **Application** is present among the ***municipal services*** group of applications available in the eGov-services git repository. The spring boot application needs the **Lombok*** extension added in your IDE to load it. Once the application is up and running API requests can be posted to the URL and ids can be generated.

- in case of IntelliJ, the plugin can be installed directly, for eclipse the Lombok jar location has to be added in eclipse.ini file in this format javaagent:lombok.jar

API Information

- Please refer Swagger API for YAML file details. Link -https://app.swaggerhub.com/apis/egov-foundation/Water-Sewerage-1.0/1.0.0.

Application.properties File Information

kafka topics persister configs for eGov persister(for notification)

- kafka.topics.notification.sms=egov.core.notification.sms
- notification.sms.enabled=true
- kafka.topics.notification.mail=notification.mail
- notification.mail.enabled=true
- kafka.topics.notification.mail.name=egov.core.notification.email
- egov.seweragecalculatorservice.createdemand=sw-generate-demand
- notification.sms.link=citizen/egov-common/pay?consumerCode=$consumerCode&tenantId=$tenantId&businessService=SW

URLs for the external API references

- eGvo mdms :-> egov.mdms.host = https://egov-micro-dev.egovernments.org
- eGov -idGen :-> egov.idgen.host = https://egov-micro-dev.egovernments.org/
- localization service :-> egov.localization.host = https://egov-micro-dev.egovernments.org/
- user-service :-> egov.user.host = https://egov-micro-dev.egovernments.org/

- Url-Shortner
  egov.url.shortner.host=http://egov-url-shortening.egov:8080

Billing Slabs

Criteria

- connection type
- building type
- calculation attribute
- property usage type

The combination of the above can be used to define the billing slab. Billing Slab is defined in mdms under sw-services-calculation folder with the SCBillingSlab. The following is the sample slab.

```
{
 "tenantId": "pb",
 "moduleName": "sw-services-calculation",
 "SCBillingSlab": [
  {
   "id": "1",
   "buildingType": "RESIDENTIAL",
   "calculationAttribute": "No. of water closets",
   "connectionType": "Non Metered",
    "minimumCharge": 0,
    "slabs": [
     {
      "from": 0,
      "to": 1000000000,
      "charge": 15
     }
    ]
  },
  {
   "id": "2",
   "buildingType": "RESIDENTIAL",
   "calculationAttribute": "No. of toilets",
   "connectionType": "Non Metered",
   "minimumCharge": 0,
   "slabs": [
    {
     "from": 0,
     "to": 1000000000,
     "charge": 15
    }
   ]
  },
  {
   "id": "3",
   "buildingType": "NONRESIDENTIAL",
   "calculationAttribute": "No. of water closets",
   "connectionType": "Non Metered",
   "minimumCharge": 0,
   "slabs": [
```

```
      {
        "from": 0,
        "to": 1000000000,
        "charge": 30
      }
    ]
  },
  {
    "id": "4",
    "buildingType": "NONRESIDENTIAL",
    "calculationAttribute": "No. of toilets",
    "connectionType": "Non Metered",
    "minimumCharge": 0,
    "slabs": [
      {
        "from": 0,
        "to": 1000000000,
        "charge": 30
      }
    ]
  },
  {
    "id": "5",
    "buildingType": "Commercial",
    "calculationAttribute": "No. of water closets",
    "connectionType": "Non Metered",
    "minimumCharge": 0,
    "slabs": [
      {
        "from": 0,
        "to": 1000000000,
        "charge": 30
      }
    ]
  },
  {
    "id": "6",
    "buildingType": "Commercial",
    "calculationAttribute": "No. of toilets",
    "connectionType": "Non Metered",
    "slabs": [
      {
        "from": 0,
        "to": 1000000000,
        "charge": 30
      }
    ]
  },
  {
    "id": "7",
    "buildingType": "Government",
    "calculationAttribute": "No. of water closets",
    "connectionType": "Non Metered",
```

```
"slabs": [
 {
  "from": 0,
  "to": 1000000000,
  "charge": 30
 }
]
},
{
 "id": "8",
 "buildingType": "Government",
 "calculationAttribute": "No. of toilets",
 "connectionType": "Non Metered",
 "slabs": [
  {
   "from": 0,
   "to": 1000000000,
   "charge": 30
  }
 ]
},
{
 "id": "9",
 "buildingType": "Partly Commercial",
 "calculationAttribute": "No. of water closets",
 "connectionType": "Non Metered",
 "slabs": [
  {
   "from": 0,
   "to": 1000000000,
   "charge": 25
  }
 ]
},
{
 "id": "10",
 "buildingType": "Partly Commercial",
 "calculationAttribute": "No. of toilets",
 "connectionType": "Non Metered",
 "slabs": [
  {
   "from": 0,
   "to": 1000000000,
   "charge": 25
  }
 ]
},
{
 "id": "11",
 "buildingType": "RESIDENTIAL",
 "calculationAttribute": "Flat",
 "connectionType": "Non Metered",
 "minimumCharge": 100,
```

```
    "slabs": []
  },
  {
    "id": "12",
    "buildingType": "NONRESIDENTIAL",
    "calculationAttribute": "Flat",
    "connectionType": "Non Metered",
    "minimumCharge": 250,
    "slabs": []
  },
  {
    "id": "13",
    "buildingType": "Commercial",
    "calculationAttribute": "Flat",
    "connectionType": "Non Metered",
    "minimumCharge": 250,
    "slabs": []
  },
  {
    "id": "14",
    "buildingType": "Government",
    "calculationAttribute": "Flat",
    "connectionType": "Non Metered",
    "minimumCharge": 350,
    "slabs": []
  },
  {
    "id": "15",
    "buildingType": "Partly commercial",
    "calculationAttribute": "Flat",
    "connectionType": "Non Metered",
    "minimumCharge": 200,
    "slabs": []
  }
 ]
}
```


If all criteria will match for that sewerage connection this slab will use for calculation.

Estimation

For application one-time fee, the estimation will return all the related tax head based on criteria. For estimation, all configuration is present in ws-services-calculation.

1. FeeSlab.json
2. PlotSizeSlab.json
3. RoadType.json

All the above master configuration is used for estimation.

Following are the exemptions and taxes that are calculated:

- Form fee
- Scrutiny fee
- Other charges

- Road cutting charges
- One time fee
- Security charges
- Tax and cess

Sewerage Charge and Tax
Sewerage charge is based on billing slab, for sewerage application charge will be based on slab and tax based on master configuration.

Interest
Below is a sample of master data JSON for interest:

```
{
 "tenantId": "pb",
 "moduleName": "sw-services-calculation",
 "Interest": [
  {
    "rate": 5,
    "minAmount": null,
    "applicableAfterDays":0,
    "flatAmount": null,
     "maxAmount": null,
     "fromFY": "2019-20",
     "startingDay": "1/01/2019"
  }
 ]
}
```

Penalty
Below is a sample of master data JSON for penalty:

```
{
 "tenantId": "pb",
 "moduleName": "sw-services-calculation",
 "Penalty": [
  {
    "rate": 10,
    "minAmount": null,
    "applicableAfterDays": 0,
    "flatAmount": null,
     "fromFY": "2019-20",
     "startingDay": "1/01/2019"
  }
 ]
}
```

Round Off
If the fraction is greater than equal to 0.5 the number is round up else it's round down. eg: 100.4 will be rounded to 100 while 100.6 will be rounded to 101.

Adding Adhoc penalty or rebate
The only employee can apply for a penalty or rebate for an existing connection. As an employee, I can update or add the penalty and rebate of a connection. This applied penalty or rebate will be added or

updated in existing demand as tax heads. For configuration, we have to add the tax head in TaxHeadMaster.json file.

```
{
  "category": "TAX",
  "service": "SW",
  "name": "Sewerage adhoc rebate",
  "code": "SW_TIME_ADHOC_REBATE",
  "isDebit": false,
  "isActualDemand": true,
  "order": "5",
  "isRequired": false
  },
  {
    "category": "TAX",
    "service": "SW",
    "name": "Sewerage adhoc penalty",
    "code": "SW_TIME_ADHOC_PENALTY",
    "isDebit": false,
    "isActualDemand": true,
    "order": "6",
    "isRequired": false
  },
```

Demand Generation

Once sewerage is sent to calculator it's tax estimates are calculated. Using this tax head estimates demand details are created. For every tax head, estimate demand generates function will create a corresponding demand detail.

Whenever _calculate API is called demand is first searched based on the connection no or application no and the demand from and to period. If demand already exists the same demand is updated else new demand is generated with consumer code as connection no or application no and demand from and to a period equal to financial year start and end period.

In case of the update if the tax head estimates change, the difference in amount for that tax head is added as new demand detail. For example, if the initial demand has one demand detail with SEWERAGE_CHARGE equal to 120

```
"demandDetails": [
        {
          "id": "77ba1e93-a535-409c-b9d1-a312c409bd45",
          "demandId": "687c3176-305b-461d-9cec-2fa26a30c88f",
          "taxHeadMasterCode": "SEWERAGE_CHARGE",
          "taxAmount": 120,
          "collectionAmount": 120,
          "additionalDetails": null,
          "auditDetails": {
              "createdBy": "04956309-87cd-4526-b4e6-48123abd4f3d",
              "lastModifiedBy": "04956309-87cd-4526-b4e6-48123abd4f3d",
              "createdTime": 1583675275873,
              "lastModifiedTime": 1583675298705
```

```
            },
            "tenantId": "pb.amritsar"
        }
    ],
```

After updating if the SEWERAGE_CHARGE increases to 150 we add one more demand detail to account for the increased amount. The demand detail will be updated to:

```
"demandDetails": [
        {
            "id": "77ba1e93-a535-409c-b9d1-a312c409bd45",
            "demandId": "687c3176-305b-461d-9cec-2fa26a30c88f",
            "taxHeadMasterCode": "SEWERAGE_CHARGE",
            "taxAmount": 120,
            "collectionAmount": 0,
            "additionalDetails": null,
            "auditDetails": {
                "createdBy": "04956309-87cd-4526-b4e6-48123abd4f3d",
                "lastModifiedBy": "04956309-87cd-4526-b4e6-48123abd4f3d",
                "createdTime": 1583675275873,
                "lastModifiedTime": 1583675298705
            },
            "tenantId": "pb.amritsar"
        },
        {
            "id": "0d83f4b0-6442-11ea-bc55-0242ac130003 ",
            "demandId": "687c3176-305b-461d-9cec-2fa26a30c88f",
            "taxHeadMasterCode": "SEWERAGE_CHARGE",
            "taxAmount": 30,
            "collectionAmount": 0,
            "additionalDetails": null,
            "auditDetails": {
                "createdBy": "04956309-87cd-4526-b4e6-48123abd4f3d",
                "lastModifiedBy": "04956309-87cd-4526-b4e6-48123abd4f3d",
                "createdTime": 1583675275873,
                "lastModifiedTime": 1583675298705
            },
            "tenantId": "pb.amritsar"
        }
    ],
```

RoundOff is bill based i.e every time bill is generated round off is adjusted so that payable amount is the whole number. Individual SW_ROUNDOFF in demand detail can be greater than 0.5 but the sum of all SW_ROUNDOFF will always be less than 0.5.

Frontend
Configurations
ws-services-masters MDMS folder:
https://github.com/egovernments/egov-mdms-data/tree/master/data/pb/ws-services-masters - Connect to preview

Documents.json

Used to display the order of the files in the Documents section which needs to be uploaded from the Citizen or Employee while creating the Water Service application.

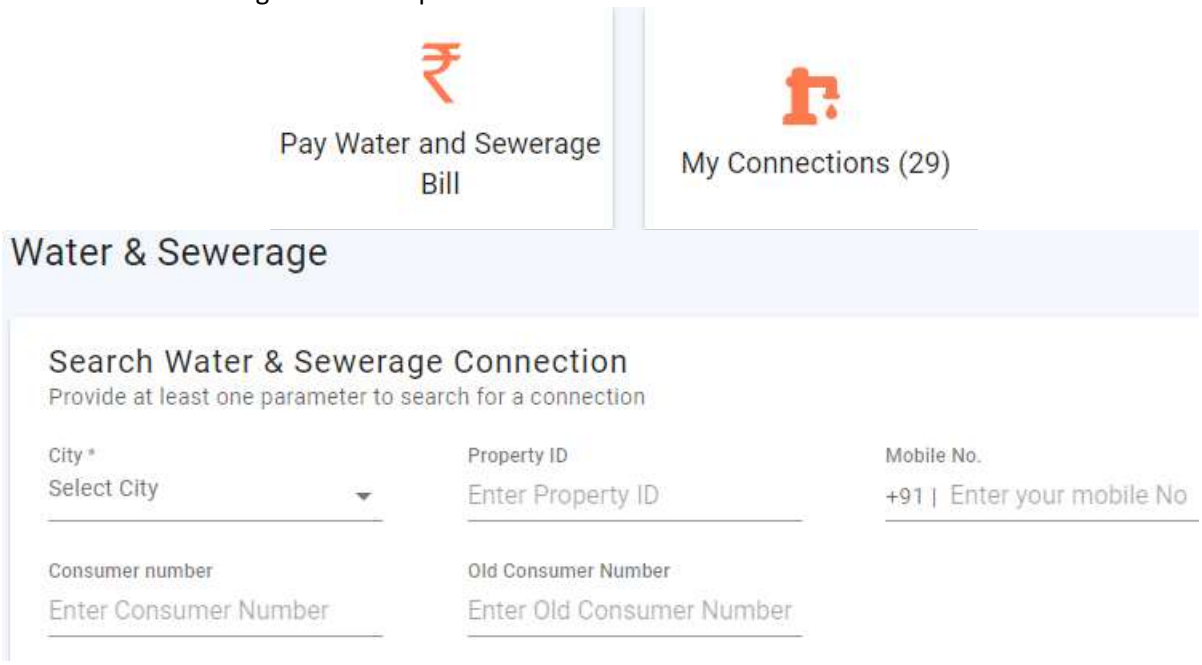
WaterSource.json - Provides details of the different types of water source and their sub types.
sw-services-calculation MDMS folder:
https://github.com/egovernments/egov-mdms-data/tree/master/data/pb/sw-services-calculation - Connect to preview

Pipesize.json - Provides the details of the pipe sizes.
Roadtype.json - Provides the details of the different types of the Roads and their cutting charges.

PDF Configurations
The UI and the PDFService retrieves the Data and Format configurations from the following path:
https://github.com/egovernments/configs/tree/master/pdf-service - Connect to preview

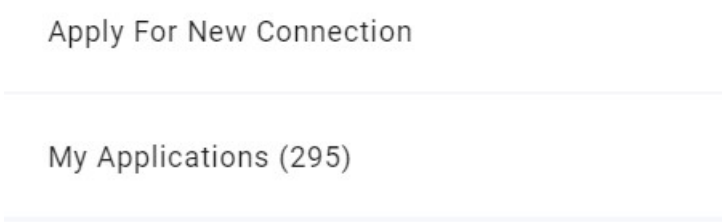
Citizen UI Guide
List of features available in the W&S service for Citizen role.

a)Search Bills & Pay:
Citizen, by using different search criteria to find the particular connection and also he/she can able to pay the water and sewerage bill for the particular connection.

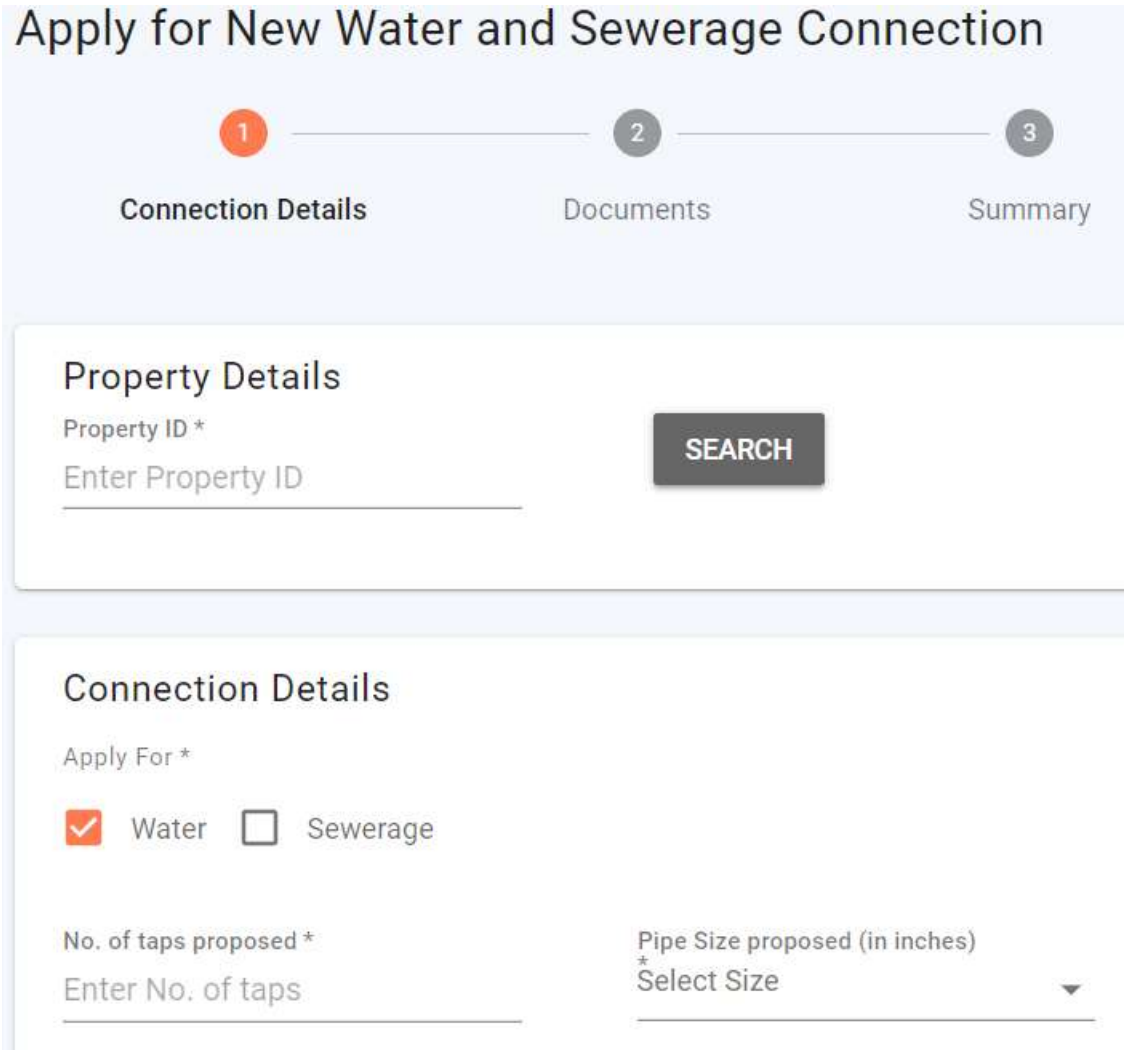


My Connections & Connection Details:
All the consumer numbers are clickable in connections list. citizen can see the all connection details and also able to download.

b) Create new application

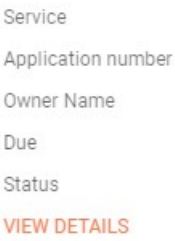


citizen can able to create new application on click of “Apply for new connection”



In this page citizen need to fill all relevant details for creating the application this is the first page of application, second page is documents upload, Third page is the summary page which includes all the provided details.

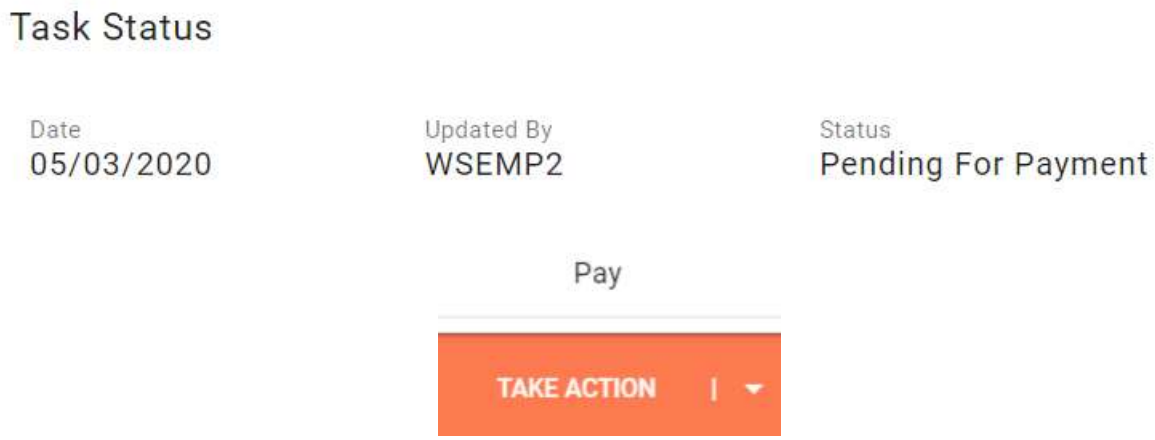
c) View Application



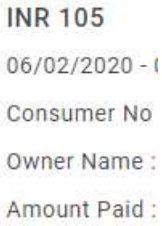
In My Applications, Citizen can see the list of applications he/she have. For every application Citizen can see the above fields. on click of ViewDetails button citizen can see the workflow page,There Citizen can perform the actions like(Edit and Resubmit)the application.

d) Pay

Citizen can also Pay the Due amount by using VIEW DETAILS link based on status (Pending for payment).



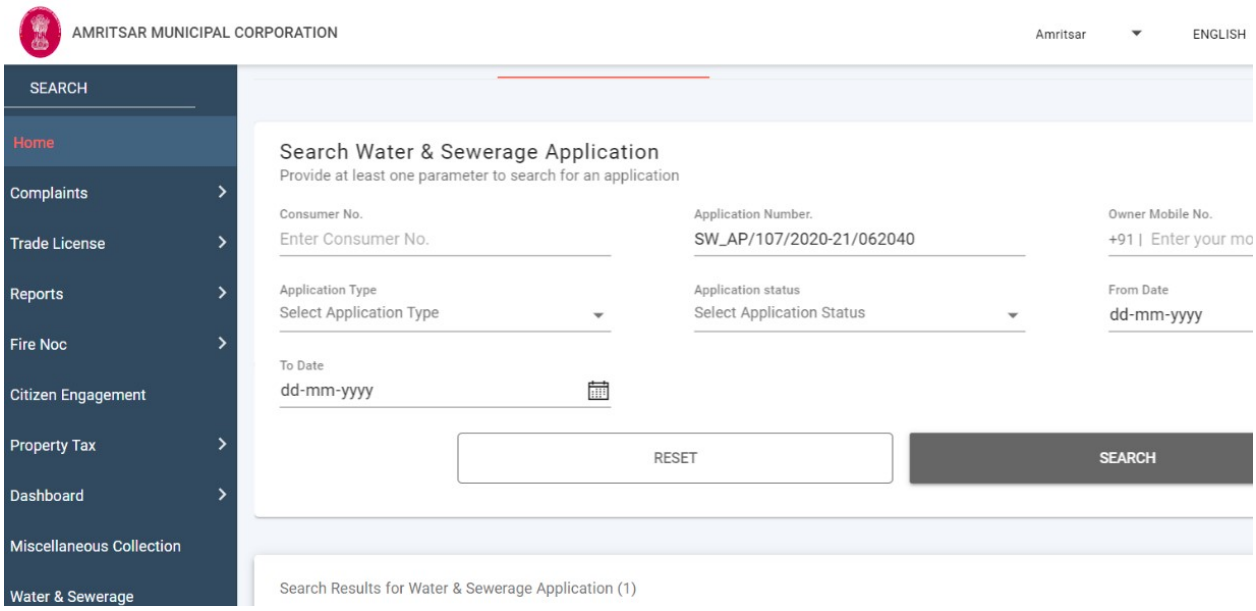
e)Past payments
Citizen can see his past payment records like which month he paid how much money and basic details are shown in this.



Employee UI Guide

SEARCH APPLICATION / CONNECTION

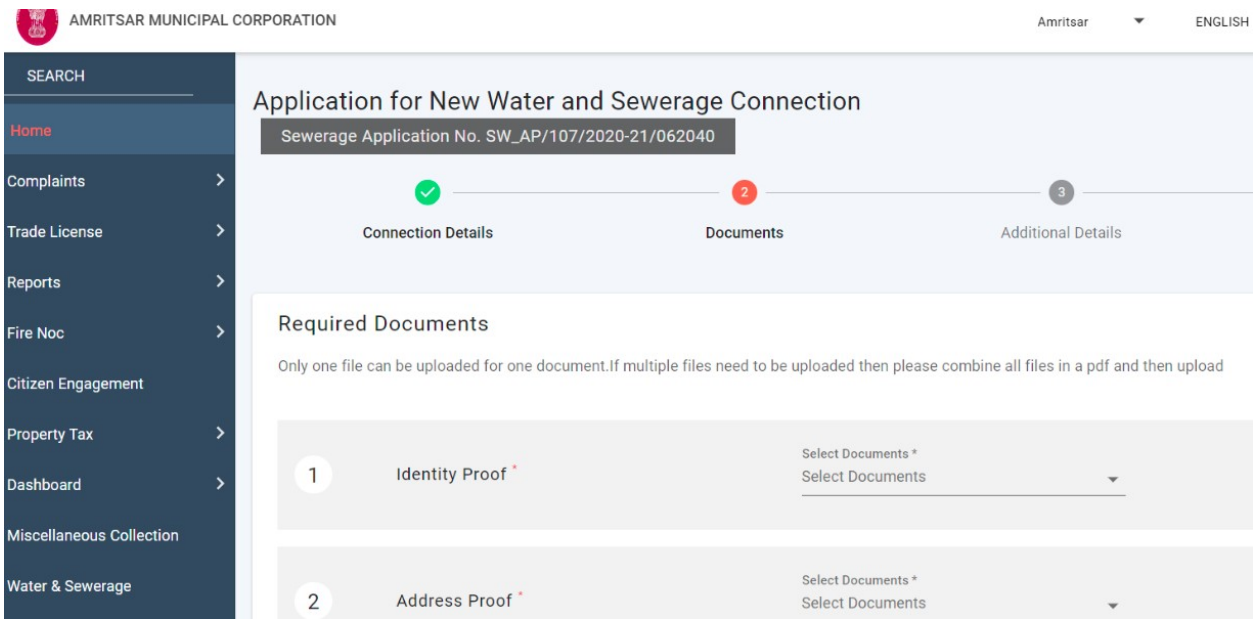
Once an Application is created (INITIATED state in Workflow), the application number can be used to search the application. There are several other criteria's that can be used to search the application.



APPLY FOR NEW WATER AND SEWERAGE CONNECTION

All the application once INITIATED can have can have until CONNECTION ACTIVATION can have multiple actions buttons. An with selected roles can forward the application with a specific action to the next stage or can make corrections to it using EDIT.

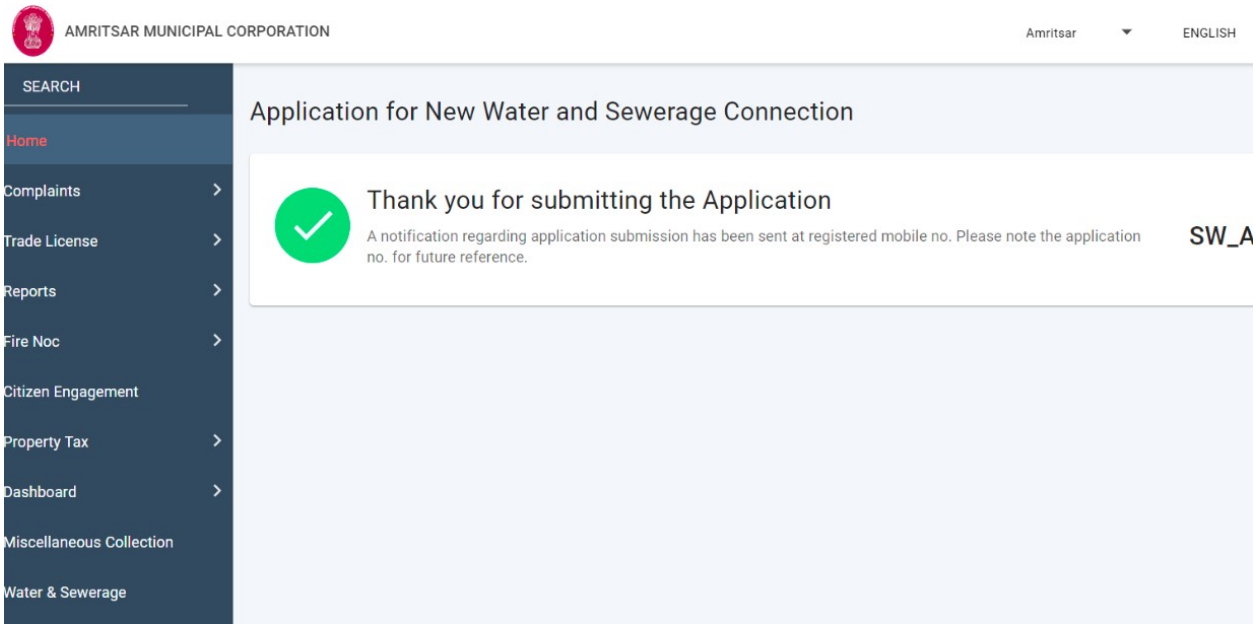
A water application has been INITIATED (The First Stage in Workflow)



All the applications that has been created once can be submitted (SUBMIT_APPLICATION state in Workflow), by either searching them in using their application number or employee can continue to the

next stages and add all the fields required and they will be able to submit the application. Once an application is submitted, they will be redirected to a screen depicted in the below image.

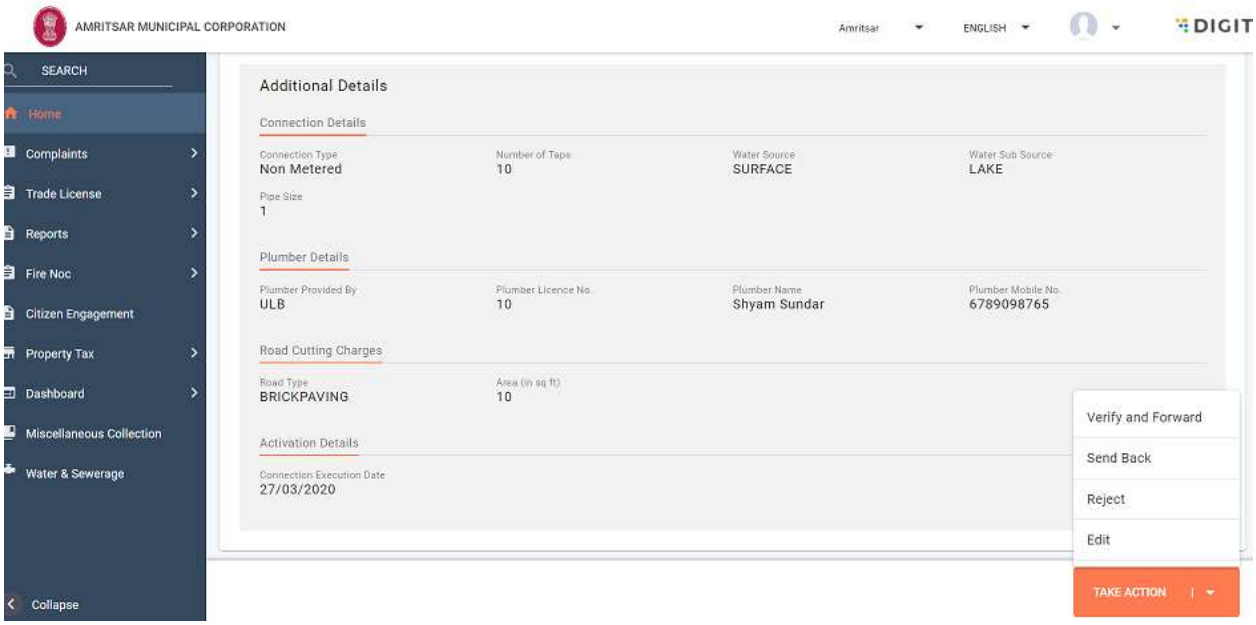
A water application has been SUBMIT_APPLICATION (The Submit Application in Workflow)



Here they will get an option to Download and Print the application, which they have just submitted. **An employee can create both Water and Sewerage application at once.**

VIEW APPLICATION & CONNECTION DETAILS

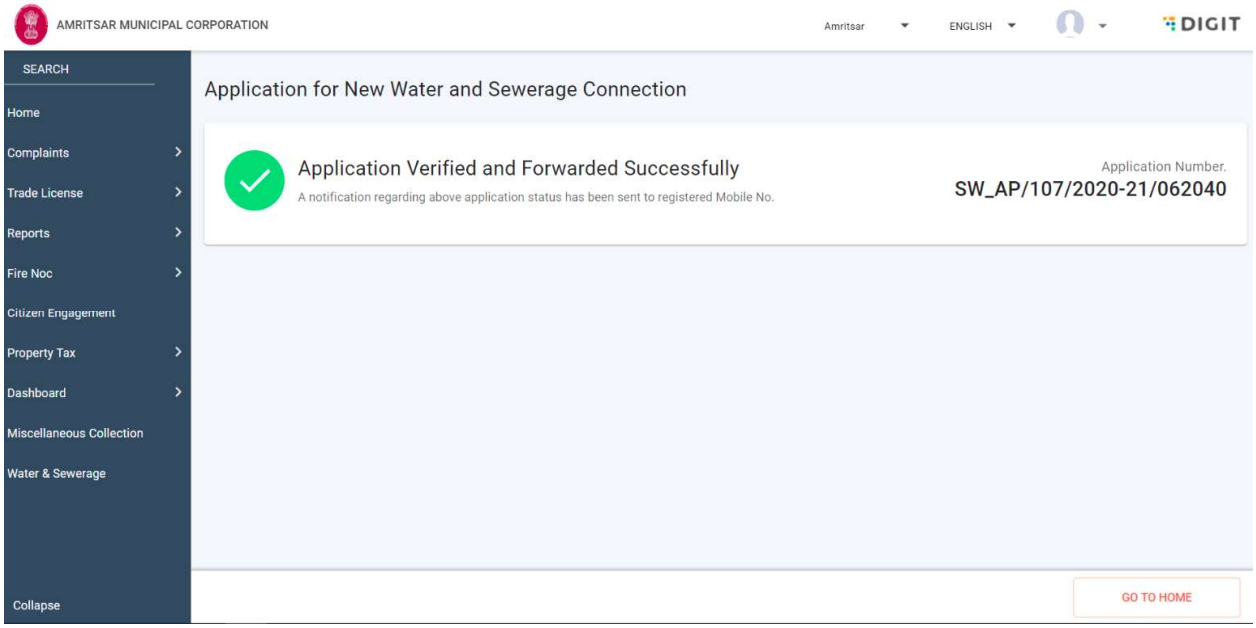
Here an Employee can edit the application, VERIFY AND FORWARD the application to the next stages, REJECT, SEND BACK TO CITIZEN who has applied for this connection. These actions that are seen in the below image, appear only for employees having a specific role which allows the employee to take the below actions.



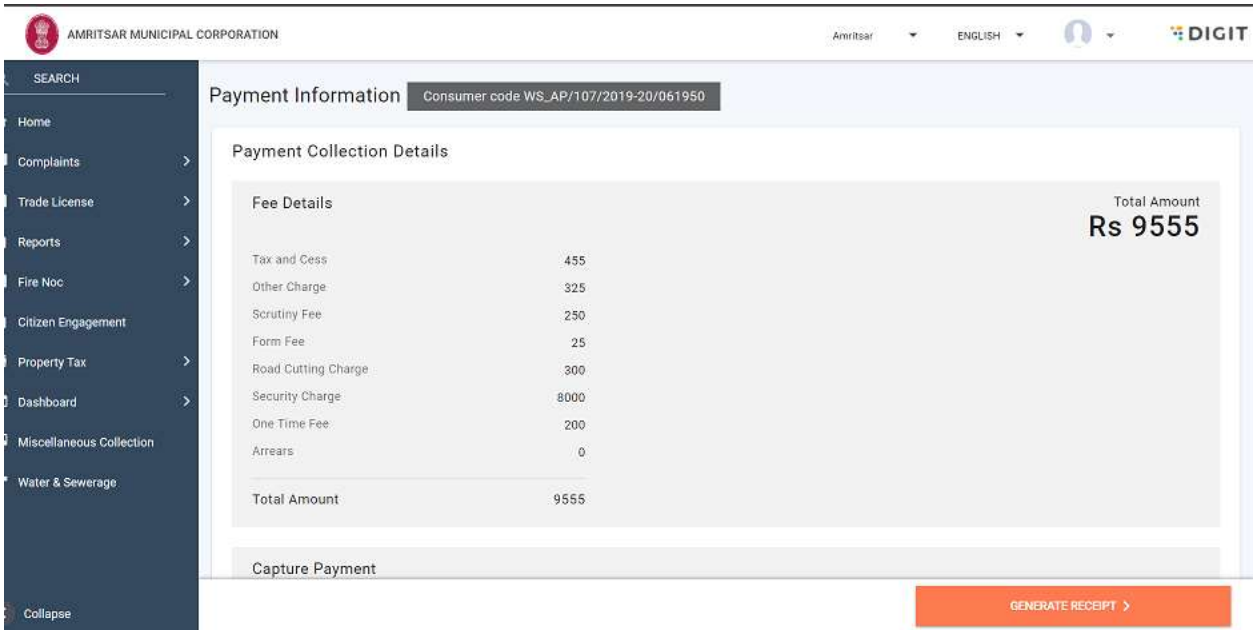
From the options provided in the above image, if the employee clicks on EDIT, it will be redirected to the apply screen where the employee will find the details of the application when it was last updated. The employee can click other option as well. It can also click on one of these options after editing the application and then it can do whatever that the employee deems right for the application.

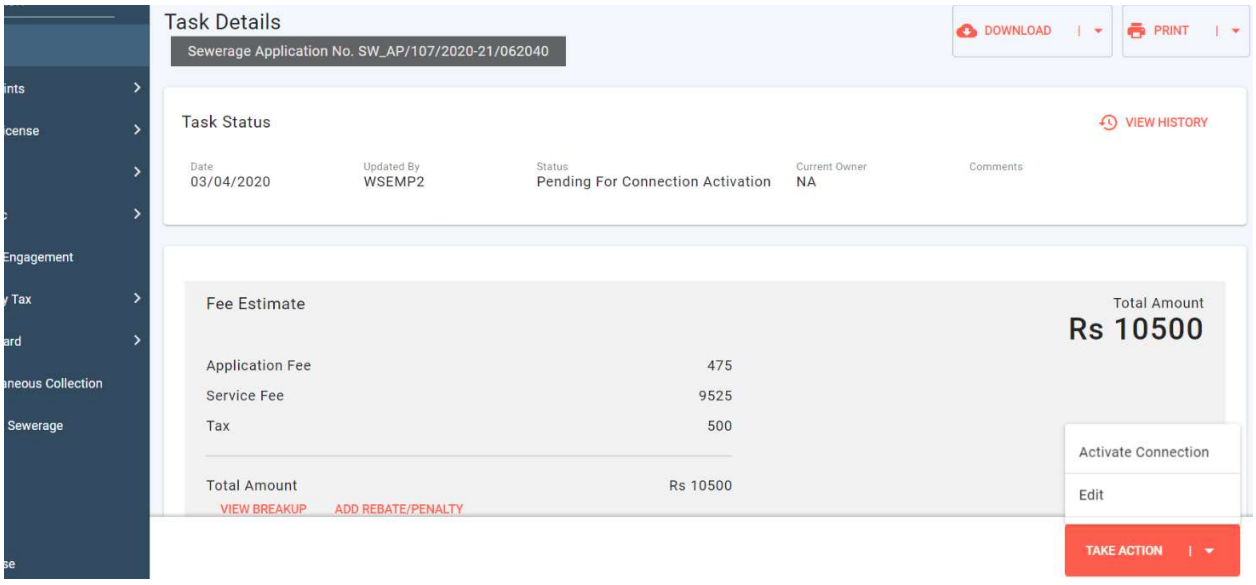
Note: The employee will only get these options if it is authorized to take any of such actions provided in the above image.

On successful completion of any of the above processes, the employee will be redirected to the below screen. The messages may change based on the actions clicked. Here the action that I have taken is VERIFY_AND_FORWARD. It can be any of the action provided in the above image.

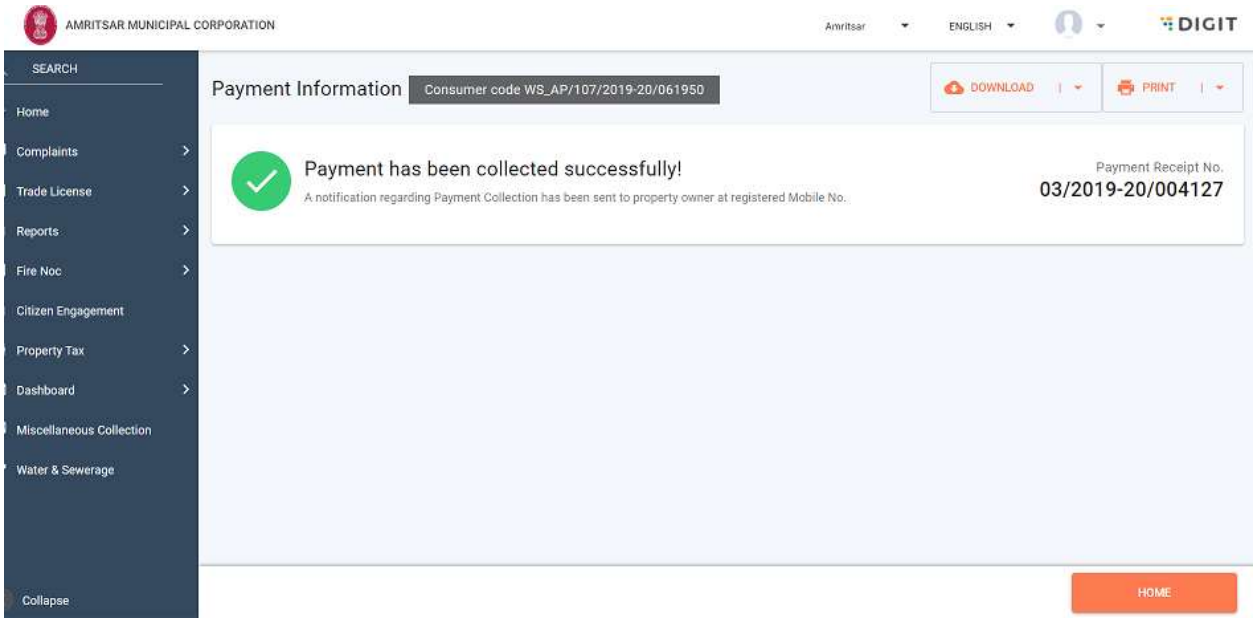


On successful completion of all the states triggered from the actions taken, the employee will reach to the action as PAY. On click of Pay, the employee will be redirected to the below screen. Here the employee can generate receipt of the amount collected from citizen. Employees will be able to see the PAY option, if they are authorized to Collect Payment from citizen.

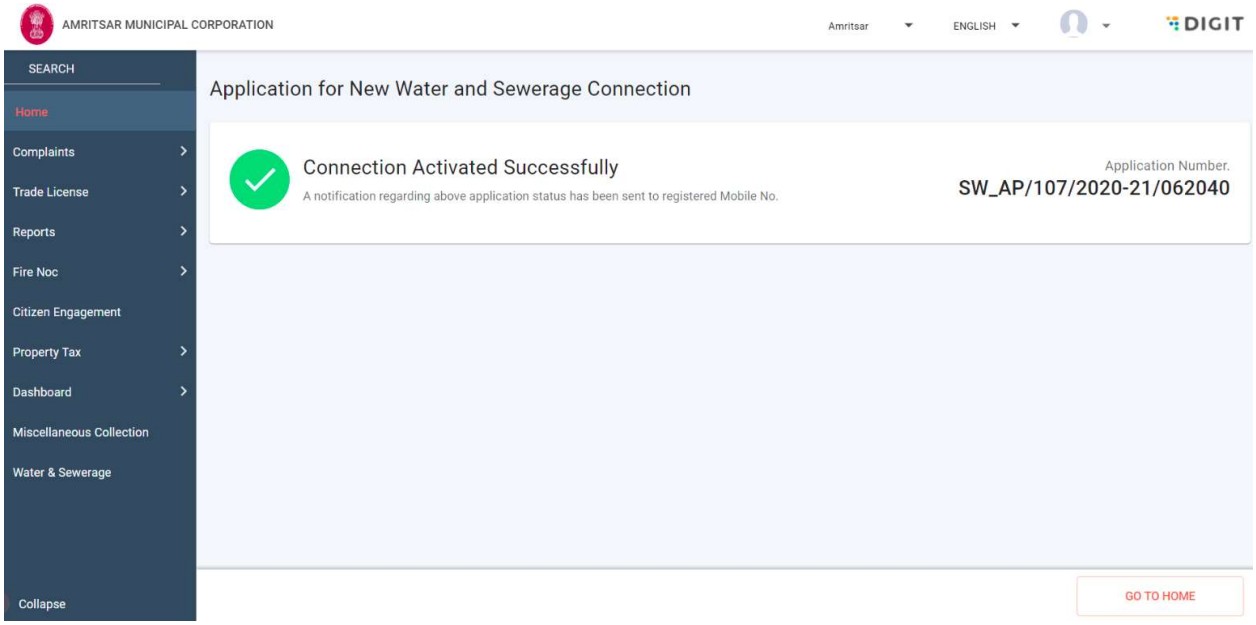




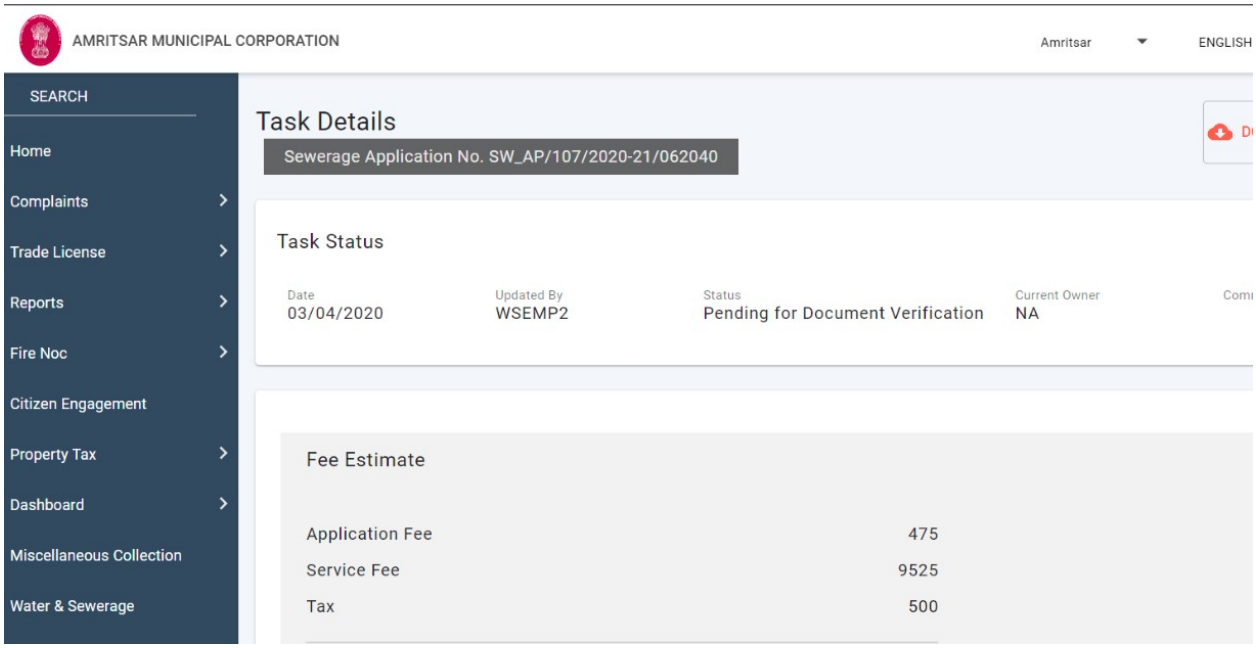
Once the payment is collected, the employee will be redirected to the below screen. Here employee can download and print receipt.



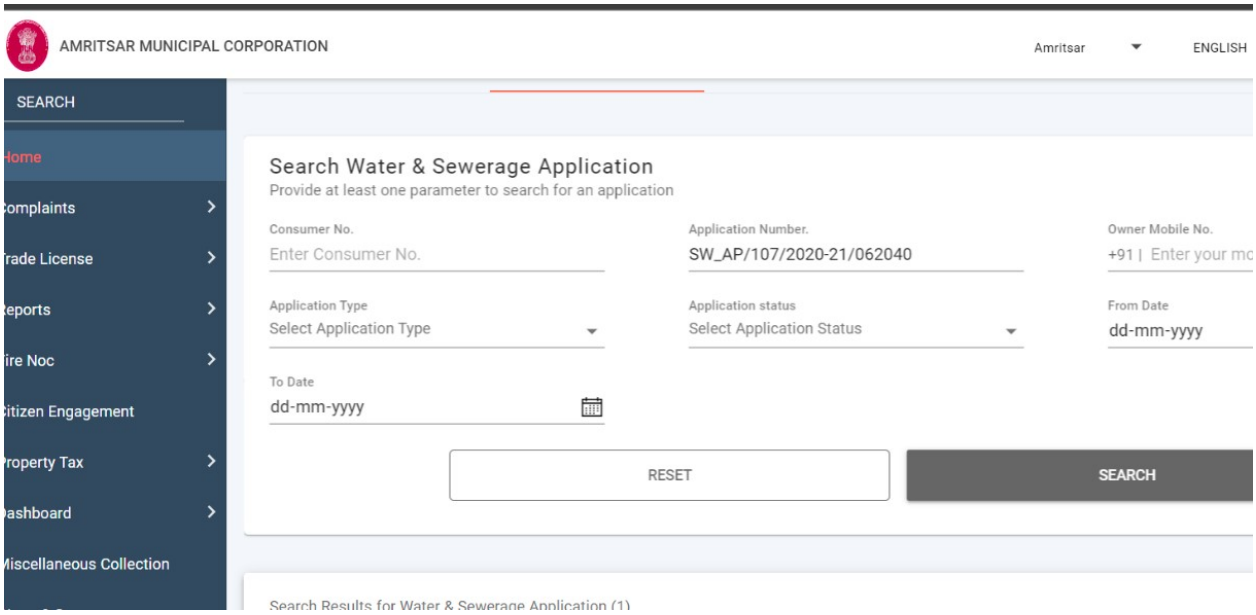
Once the connection is activated, the employee will be redirected to the below screen. Here employee can go back to home screen.



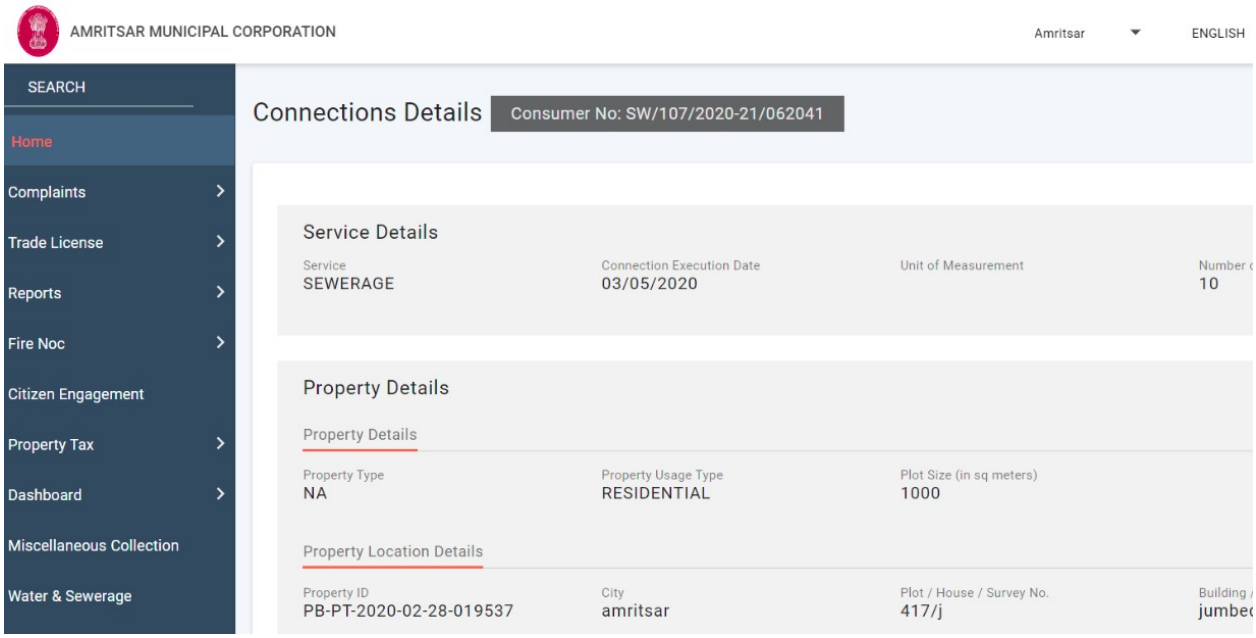
Once payment has been accepted and the receipt has been generated, the employee can go to search and search the application based on application number. The employee will find the application in Pending Connection Activation state, similar to what is show in the image below. If the employee is authorized to activate the connection, it can activate the connection.



After the connection has been activated, employee can go to the search and search the connection based on connection number, as shown in the image below.



After getting the search result as in the above image for the activate connection, the employee can see the details of the connection (Water or Sewerage) after clicking on the connection number in the Consumer No. column. The connection details looks like below, where the employee can find all the details related to the connection created.



Employee can also download and print the connection details if required.