

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 preferable).
- 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.sewageservice.createconnection=save-sw-connection

egov.sewageservice.updateconnection=update-sw-connection

egov.sewageservice.updatesewerageconnection.workflow.topic=update-sw-workflow

URLs for the external API references

eGov 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=sewageservice.connection.id](#)

egov.idgen.scid.format=SW/[CITY.CODE]/[fy:yyyy-yy]/[SEQ_EGOV_COMMON]

[egov.idgen.scapid.name=sewageservice.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

1sw-services-calculation

2PropertyTax

master-config.json for sewerage service

```
1"sw-services-calculation": {
2  "Interest": {
3    "masterName": "Interest",
4    "isStateLevel": true,
5    "uniqueKeys": [
6      "$.fromFY"
7    ]
8  },
9  "Rebate": {
10   "masterName": "Rebate",
11   "isStateLevel": true,
12   "uniqueKeys": [
13     "$.fromFY"
14   ]
15 },
16 "Penalty": {
17   "masterName": "Penalty",
18   "isStateLevel": true,
19   "uniqueKeys": [
20     "$.fromFY"
21   ]
22 },
23 "SCBillingSlab": {
24   "masterName": "SCBillingSlab",
25   "isStateLevel": true,
26   "uniqueKeys": []
```

```

27 },
28 "billingPeriod": {
29   "masterName": "billingPeriod",
30   "isStateLevel": true,
31   "uniqueKeys": [
32     "$.billingCycle"
33   ]
34 },
35 "SW_CHARGE": {
36   "masterName": "SW_CHARGE",
37   "isStateLevel": true,
38   "uniqueKeys": []
39 },
40 "SW_TIME_PENALTY": {
41   "masterName": "SW_TIME_PENALTY",
42   "isStateLevel": true,
43   "uniqueKeys": []
44 },
45 "SW_Round_Off": {
46   "masterName": "SW_Round_Off",
47   "isStateLevel": true,
48   "uniqueKeys": []
49 },
50 "PlotSizeSlab": {
51   "masterName": "PlotSizeSlab",
52   "isStateLevel": true,
53   "uniqueKeys": []
54 },
55 "PropertyUsageType": {
56   "masterName": "PropertyUsageType",
57   "isStateLevel": true,
58   "uniqueKeys": []
59 },
60 "FeeSlab": {
61   "masterName": "FeeSlab",
62   "isStateLevel": true,
63   "uniqueKeys": []
64 },
65 "RoadType": {
66   "masterName": "RoadType",
67   "isStateLevel": true,
68   "uniqueKeys": []
69 },
70 "CalculationAttribute": {
71   "masterName": "CalculationAttribute",
72   "isStateLevel": true,
73   "uniqueKeys": []
74 }
75 }

```

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:

```
1{
2  "BusinessServices": [
3    {
4      "tenantId": "pb",
5      "businessService": "NewSW1",
6      "business": "sw-services",
7      "businessServiceSla": 259200000,
8      "states": [
9        {
10         "sla": null,
11         "state": null,
12         "applicationStatus": null,
13         "docUploadRequired": false,
14         "isStartState": true,
15         "isTerminateState": false,
16         "isStateUpdatable": false,
17         "actions": [
18           {
19             "action": "INITIATE",
20             "nextState": "INITIATED",
21             "roles": [
22               "CITIZEN",
23               "SW_CEMP"
24             ]
25           }
26         ],
27       },
28       {
29         "sla": null,
30         "state": "INITIATED",
31         "applicationStatus": "INITIATED",
32         "docUploadRequired": false,
33         "isStartState": false,
34         "isTerminateState": false,
35         "isStateUpdatable": true,
36         "actions": [
37           {
38             "action": "SUBMIT_APPLICATION",
39             "nextState": "PENDING_FOR_DOCUMENT_VERIFICATION",
40             "roles": [
41               "CITIZEN",
42               "SW_CEMP"
43             ]
44           }
45         ]
46       },
47     ]
48   }
49 }
```

```

47 {
48   "sla": null,
49   "state": "PENDING_FOR_CITIZEN_ACTION",
50   "applicationStatus": "PENDING_FOR_CITIZEN_ACTION",
51   "docUploadRequired": false,
52   "isStartState": false,
53   "isTerminateState": false,
54   "isStateUpdatable": true,
55   "actions": [
56     {
57       "action": "RESUBMIT_APPLICATION",
58       "nextState": "PENDING_FOR_DOCUMENT_VERIFICATION",
59       "roles": [
60         "CITIZEN",
61         "SW_CEMP"
62       ]
63     }
64   ],
65 },
66 {
67   "sla": null,
68   "state": "PENDING_FOR_DOCUMENT_VERIFICATION",
69   "applicationStatus": "PENDING_FOR_DOCUMENT_VERIFICATION",
70   "docUploadRequired": false,
71   "isStartState": false,
72   "isTerminateState": false,
73   "isStateUpdatable": true,
74   "actions": [
75     {
76       "action": "VERIFY_AND_FORWARD",
77       "nextState": "PENDING_FOR_FIELD_INSPECTION",
78       "roles": [
79         "SW_DOC_VERIFIER"
80       ]
81     },
82     {
83       "action": "REJECT",
84       "nextState": "REJECTED",
85       "roles": [
86         "SW_DOC_VERIFIER"
87       ]
88     },
89     {
90       "action": "SEND_BACK_TO_CITIZEN",
91       "nextState": "PENDING_FOR_CITIZEN_ACTION",
92       "roles": [
93         "SW_DOC_VERIFIER"
94       ]
95     }
96   ],
97 },
98 {
99   "sla": null,

```

```
100     "state": "REJECTED",
101     "applicationStatus": "REJECTED",
102     "isStateUpdatable": false,
103     "docUploadRequired": false,
104     "isStartState": false,
105     "isTerminateState": true
106 },
107 {
108     "sla": 86400000,
109     "state": "PENDING_FOR_FIELD_INSPECTION",
110     "applicationStatus": "PENDING_FOR_FIELD_INSPECTION",
111     "docUploadRequired": false,
112     "isStartState": false,
113     "isStateUpdatable": true,
114     "isTerminateState": false,
115     "actions": [
116         {
117             "action": "VERIFY_AND_FORWARD",
118             "nextState": "PENDING_APPROVAL_FOR_CONNECTION",
119             "roles": [
120                 "SW_FIELD_INSPECTOR"
121             ]
122         },
123         {
124             "action": "REJECT",
125             "nextState": "REJECTED",
126             "roles": [
127                 "SW_FIELD_INSPECTOR"
128             ]
129         },
130         {
131             "action": "SEND_BACK_FOR_DOCUMENT_VERIFICATION",
132             "nextState": "PENDING_FOR_DOCUMENT_VERIFICATION",
133             "roles": [
134                 "SW_FIELD_INSPECTOR"
135             ]
136         }
137     ]
138 },
139 {
140     "sla": 43200000,
141     "state": "PENDING_APPROVAL_FOR_CONNECTION",
142     "applicationStatus": "PENDING_APPROVAL_FOR_CONNECTION",
143     "docUploadRequired": false,
144     "isStartState": false,
145     "isStateUpdatable": true,
146     "isTerminateState": false,
147     "actions": [
148         {
149             "action": "APPROVE_FOR_CONNECTION",
150             "nextState": "PENDING_FOR_PAYMENT",
151             "roles": [
152                 "SW_APPROVER"
```

```
153     ]
154   },
155   {
156     "action": "REJECT",
157     "nextState": "REJECTED",
158     "roles": [
159       "SW_APPROVER"
160     ]
161   },
162   {
163     "action": "SEND_BACK_FOR_FIELD_INSPECTION",
164     "nextState": "PENDING_FOR_FIELD_INSPECTION",
165     "roles": [
166       "SW_APPROVER"
167     ]
168   }
169 ]
170 },
171 {
172   "sla": 43200000,
173   "state": "PENDING_FOR_PAYMENT",
174   "applicationStatus": "PENDING_FOR_PAYMENT",
175   "docUploadRequired": false,
176   "isStartState": false,
177   "isTerminateState": false,
178   "isStateUpdatable": false,
179   "actions": [
180     {
181       "action": "PAY",
182       "nextState": "PENDING_FOR_CONNECTION_ACTIVATION",
183       "roles": [
184         "CITIZEN",
185         "SW_CEMP"
186       ]
187     }
188   ]
189 },
190 {
191   "sla": null,
192   "state": "PENDING_FOR_CONNECTION_ACTIVATION",
193   "applicationStatus": "PENDING_FOR_CONNECTION_ACTIVATION",
194   "isStateUpdatable": true,
195   "docUploadRequired": false,
196   "isStartState": false,
197   "isTerminateState": false,
198   "actions": [
199     {
200       "action": "ACTIVATE_CONNECTION",
201       "nextState": "CONNECTION_ACTIVATED",
202       "roles": [
203         "SW_CLERK"
204       ]
205     }
206   ]
207 }
```

```

206     ]
207   },
208   {
209     "sla": null,
210     "state": "CONNECTION_ACTIVATED",
211     "applicationStatus": "CONNECTION_ACTIVATED",
212     "isStateUpdatable": false,
213     "docUploadRequired": false,
214     "isStartState": false,
215     "isTerminateState": true
216   }
217 ]
218 }
219 ]
220}

```

Workflow for property creation through Water and Sewerage Module

```

1{
2  "BusinessServices": [
3    {
4      "tenantId": "pb",
5      "businessService": "NewWS1",
6      "business": "ws-services",
7      "businessServiceSla": 259200000,
8      "states": [
9        {
10       "sla": null,
11       "state": null,
12       "applicationStatus": "INWORKFLOW",
13       "docUploadRequired": false,
14       "isStartState": true,
15       "isTerminateState": false,
16       "isStateUpdatable": false,
17       "actions": [
18         {
19           "action": "OPEN",
20           "nextState": "INITIATED",
21           "roles": [
22             "CITIZEN",
23             "WS_CEMP"
24           ]
25         }
26       ]
27     },
28     {
29       "sla": null,
30       "state": "INITIATED",
31       "applicationStatus": "INWORKFLOW",
32       "docUploadRequired": false,
33       "isStartState": true,
34       "isTerminateState": false,
35       "isStateUpdatable": true,
36       "actions": [

```



```

37     {
38         "action": "SUBMIT",
39         "nextState": "APPROVED",
40         "roles": [
41             "CITIZEN",
42             "WS_CEMP"
43         ]
44     }
45 ]
46 },
47 {
48     "sla": null,
49     "state": "APPROVED",
50     "applicationStatus": "ACTIVE",
51     "docUploadRequired": false,
52     "isStartState": false,
53     "isTerminateState": true,
54     "isStateUpdatable": false,
55     "actions": null
56 }
57 ]
58 }
59 ]
60 }

```

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: Configuration of the query params to be used for the API call. It is a comma separated 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 parameter.

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

```
1ServiceMaps:
2 serviceName: Sewerage Service - rainmaker
3 version: 1.0.0
4 mappings:
5   - topic: save-sw-connection
6     configKey: INDEX
7     indexes:
8       - name: sewerage-services
9         type: general
10       id: $.id$.tenantId
11       timeStampField: $.auditDetails.createdTime
```

```

12  jsonPath: $.SewerageConnection
13  customJsonMapping:
14    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":{}}}
15    fieldMapping:
16      - inJsonPath: $.applicationStatus.state
17        outJsonPath: $.Data.workflow.state
18      - inJsonPath: $.processInstance.action
19        outJsonPath: $.Data.workflow.action
20      - inJsonPath: $.processInstance.assignes.*.uuid
21        outJsonPath: $.Data.workflow.assignes
22      - inJsonPath: $.applicationNo
23        outJsonPath: $.Data.applicationNo
24      - inJsonPath: $.applicationStatus
25        outJsonPath: $.Data.applicationStatus
26      - inJsonPath: $.status
27        outJsonPath: $.Data.status
28      - inJsonPath: $.connectionNo
29        outJsonPath: $.Data.connectionNo
30      - inJsonPath: $.oldConnectionNo
31        outJsonPath: $.Data.oldConnectionNo
32      - inJsonPath: $.plumberInfo
33        outJsonPath: $.Data.plumberInfo
34      - inJsonPath: $.roadCuttingInfo
35        outJsonPath: $.Data.roadCuttingInfo
36      - inJsonPath: $.connectionHolders
37        outJsonPath: $.Data.connectionHolders
38      - inJsonPath: $.roadType
39        outJsonPath: $.Data.roadType
40      - inJsonPath: $.roadCuttingArea
41        outJsonPath: $.Data.roadCuttingArea
42      - inJsonPath: $.connectionExecutionDate
43        outJsonPath: $.Data.connectionExecutionDate
44      - inJsonPath: $.connectionCategory
45        outJsonPath: $.Data.connectionCategory
46      - inJsonPath: $.connectionType
47        outJsonPath: $.Data.connectionType
48      - inJsonPath: $.additionalDetails
49        outJsonPath: $.Data.additionalDetails
50      - inJsonPath: $.id
51        outJsonPath: $.Data.id
52      - inJsonPath: $.propertyId
53        outJsonPath: $.Data.propertyId
54      - inJsonPath: $.tenantId
55        outJsonPath: $.Data.tenantId
56      - inJsonPath: $.proposedWaterClosets
57        outJsonPath: $.Data.proposedWaterClosets
58      - inJsonPath: $.proposedToilets
59        outJsonPath: $.Data.proposedToilets

```

```

60 - inJsonPath: $.noOfWaterClosets
61   outJsonPath: $.Data.noOfWaterClosets
62 - inJsonPath: $.noOfToilets
63   outJsonPath: $.Data.noOfToilets
64 - inJsonPath: $.applicationType
65   outJsonPath: $.Data.applicationType
66 - inJsonPath: $.dateEffectiveFrom
67   outJsonPath: $.Data.dateEffectiveFrom
68   externalUriMapping:
69     - path: http://egov-workflow-v2.egov:8080/egov-workflow-v2/egov-wf/process/_search
70       queryParams: businessIds=$.applicationNo,history=true,tenantId=$.tenantId
71     apiRequest:
{"RequestInfo":{"apild":"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"}]}}}
72     uriResponseMapping:
73     - inJsonPath: $.ProcessInstances
74       outJsonPath: $.Data.history
75
76 - topic: update-sw-connection
77   configKey: INDEX
78   indexes:
79     - name: sewerage-services
80       type: general
81       id: $.id,$.tenantId
82       timeStampField: $.auditDetails.lastModifiedTime
83       jsonPath: $.SewerageConnection
84       customJsonMapping:
85         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": {}}}
86         fieldMapping:
87         - inJsonPath: $.applicationStatus.state
88           outJsonPath: $.Data.workflow.state
89         - inJsonPath: $.processInstance.action
90           outJsonPath: $.Data.workflow.action
91         - inJsonPath: $.processInstance.assignes.*.uuid
92           outJsonPath: $.Data.workflow.assignes
93         - inJsonPath: $.applicationNo
94           outJsonPath: $.Data.applicationNo
95         - inJsonPath: $.applicationStatus
96           outJsonPath: $.Data.applicationStatus
97         - inJsonPath: $.status
98           outJsonPath: $.Data.status
99         - inJsonPath: $.connectionNo
100          outJsonPath: $.Data.connectionNo
101         - inJsonPath: $.oldConnectionNo
102          outJsonPath: $.Data.oldConnectionNo

```

```

103 - inJsonPath: $.plumberInfo
104   outJsonPath: $.Data.plumberInfo
105 - inJsonPath: $.roadCuttingInfo
106   outJsonPath: $.Data.roadCuttingInfo
107 - inJsonPath: $.connectionHolders
108   outJsonPath: $.Data.connectionHolders
109 - inJsonPath: $.roadType
110   outJsonPath: $.Data.roadType
111 - inJsonPath: $.roadCuttingArea
112   outJsonPath: $.Data.roadCuttingArea
113 - inJsonPath: $.connectionExecutionDate
114   outJsonPath: $.Data.connectionExecutionDate
115 - inJsonPath: $.connectionCategory
116   outJsonPath: $.Data.connectionCategory
117 - inJsonPath: $.connectionType
118   outJsonPath: $.Data.connectionType
119 - inJsonPath: $.additionalDetails
120   outJsonPath: $.Data.additionalDetails
121 - inJsonPath: $.id
122   outJsonPath: $.Data.id
123 - inJsonPath: $.propertyId
124   outJsonPath: $.Data.propertyId
125 - inJsonPath: $.tenantId
126   outJsonPath: $.Data.tenantId
127 - inJsonPath: $.proposedWaterClosets
128   outJsonPath: $.Data.proposedWaterClosets
129 - inJsonPath: $.proposedToilets
130   outJsonPath: $.Data.proposedToilets
131 - inJsonPath: $.noOfWaterClosets
132   outJsonPath: $.Data.noOfWaterClosets
133 - inJsonPath: $.noOfToilets
134   outJsonPath: $.Data.noOfToilets
135 - inJsonPath: $.applicationType
136   outJsonPath: $.Data.applicationType
137 - inJsonPath: $.dateEffectiveFrom
138   outJsonPath: $.Data.dateEffectiveFrom
139 externalUriMapping:
140 - path: http://egov-workflow-v2.egov:8080/egov-workflow-v2/egov-wf/process/_search
141   queryParams: businessIds=$.applicationNo,history=true,tenantId=$.tenantId
142   apiRequest:
{"RequestInfo":{"apild":"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"}]}}}
143   uriResponseMapping:
144 - inJsonPath: $.ProcessInstances
145   outJsonPath: $.Data.history
146
147
148 - topic: update-sw-workflow
149 configKey: INDEX
150 indexes:

```

```

151 - name: sewerage-services
152   type: general
153   id: $.id,$.tenantId
154   timeStampField: $.auditDetails.lastModifiedTime
155   jsonPath: $.SewerageConnection
156   customJsonMapping:
157     indexMapping: {"Data":{"workflow":{"state": {}, "action": "", "assignes":
158     []},"applicationNo":"","applicationStatus":"","status":"","connectionNo":"","oldConnectionNo":"","plumberInfo":
159     [],"roadCuttingInfo":[],"connectionHolders":[],"roadType":"","roadCuttingArea":"","connectionExecutionDate":
160     ","connectionCategory":"","connectionType":"","additionalDetails":{"id":"","propertyId":"","tenantId":"","propos
161     edWaterClosets":"","proposedToilets":"","noOfWaterClosets":"","noOfToilets":"","applicationType":"","dateEff
162     ectiveFrom":"","history":{}}}
163     fieldMapping:
164       - inJsonPath: $.applicationStatus.state
165         outJsonPath: $.Data.workflow.state
166       - inJsonPath: $.processInstance.action
167         outJsonPath: $.Data.workflow.action
168       - inJsonPath: $.processInstance.assignes.*.uuid
169         outJsonPath: $.Data.workflow.assignes
170       - inJsonPath: $.applicationNo
171         outJsonPath: $.Data.applicationNo
172       - inJsonPath: $.applicationStatus
173         outJsonPath: $.Data.applicationStatus
174       - inJsonPath: $.status
175         outJsonPath: $.Data.status
176       - inJsonPath: $.connectionNo
177         outJsonPath: $.Data.connectionNo
178       - inJsonPath: $.oldConnectionNo
179         outJsonPath: $.Data.oldConnectionNo
180       - inJsonPath: $.plumberInfo
181         outJsonPath: $.Data.plumberInfo
182       - inJsonPath: $.roadCuttingInfo
183         outJsonPath: $.Data.roadCuttingInfo
184       - inJsonPath: $.connectionHolders
185         outJsonPath: $.Data.connectionHolders
186       - inJsonPath: $.roadType
187         outJsonPath: $.Data.roadType
188       - inJsonPath: $.roadCuttingArea
189         outJsonPath: $.Data.roadCuttingArea
190       - inJsonPath: $.connectionExecutionDate
191         outJsonPath: $.Data.connectionExecutionDate
192       - inJsonPath: $.connectionCategory
193         outJsonPath: $.Data.connectionCategory
194       - inJsonPath: $.connectionType
195         outJsonPath: $.Data.connectionType
196       - inJsonPath: $.additionalDetails
197         outJsonPath: $.Data.additionalDetails
198       - inJsonPath: $.id
199         outJsonPath: $.Data.id
200       - inJsonPath: $.propertyId
201         outJsonPath: $.Data.propertyId
202       - inJsonPath: $.tenantId
203         outJsonPath: $.Data.tenantId

```

```

199     - inJsonPath: $.proposedWaterClosets
200       outJsonPath: $.Data.proposedWaterClosets
201     - inJsonPath: $.proposedToilets
202       outJsonPath: $.Data.proposedToilets
203     - inJsonPath: $.noOfWaterClosets
204       outJsonPath: $.Data.noOfWaterClosets
205     - inJsonPath: $.noOfToilets
206       outJsonPath: $.Data.noOfToilets
207     - inJsonPath: $.applicationType
208       outJsonPath: $.Data.applicationType
209     - inJsonPath: $.dateEffectiveFrom
210       outJsonPath: $.Data.dateEffectiveFrom
211     externalUriMapping:
212     - path: http://egov-workflow-v2.egov:8080/egov-workflow-v2/egov-wf/process/_search
213       queryParam: businessIds=$.applicationNo,history=true,tenantId=$.tenantId
214     apiRequest:
{"RequestInfo":{"apild":"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"}]}}}
215     uriResponseMapping:
216     - inJsonPath: $.ProcessInstances
217       outJsonPath: $.Data.history

```

Notifications

To enable or disable notification

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.egovmment.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

```

1[
2  {
3    "code": "SW_INITIATE_INITIATED_SMS_MESSAGE",
4    "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>.",
5     "module": "rainmaker-ws",
6     "locale": "en_IN"
7 },
8 {
9     "code": "SW_INITIATE_INITIATED_APP_MESSAGE",
10    "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>",
11    "module": "rainmaker-ws",
12    "locale": "en_IN"
13 },
14 {
15    "code": "SW_REJECT_REJECTED_SMS_MESSAGE",
16    "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>.",
17    "module": "rainmaker-ws",
18    "locale": "en_IN"
19 },
20 {
21    "code": "SW_REJECT_REJECTED_APP_MESSAGE",
22    "message": "Dear <Owner Name>, Your Application <Application number> for a New <Service>
Connection has been rejected. Click here for more details <View History Link>",
23    "module": "rainmaker-ws",
24    "locale": "en_IN"
25 },
26 {
27    "code": "SW_EDIT_SMS_MESSAGE",
28    "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>.",
29    "module": "rainmaker-ws",
30    "locale": "en_IN"
31 },
32 {
33    "code": "SW_EDIT_IN_APP_MESSAGE",
34    "message": "Dear <Owner Name>, Your Application <Application number> for a New <Service>
Connection has been edited. Click here for more details <View History Link>.",
35    "module": "rainmaker-ws",
36    "locale": "en_IN"
37 },
38 {
39    "code": "SW_VERIFY_AND_FORWARD_PENDING_FOR_FIELD_INSPECTION_SMS_MESSAGE",
40    "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>.",
41    "module": "rainmaker-ws",
42    "locale": "en_IN"
43 },
44 {
45    "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_M
ESSAG",
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_M
ESSAGE",
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",

```

```

82     "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>",
83     "module": "rainmaker-ws",
84     "locale": "en_IN"
85 },
86 {
87     "code": "SEWERAGE_CONNECTION_BILL_GENERATION_APP_MESSAGE",
88     "message": "Dear <Owner Name>, Your <Service> Bill has been generated. Please pay the
amount due: <bill amount> by due date <Due Date>.",
89     "module": "rainmaker-ws",
90     "locale": "en_IN"
91 },
92 {
93     "code": "SW_APPROVE_FOR_CONNECTION_PENDING_FOR_PAYMENT_SMS_MESSAGE",
94     "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.",
95     "module": "rainmaker-ws",
96     "locale": "en_IN"
97 },
98 {
99     "code": "SW_APPROVE_FOR_CONNECTION_PENDING_FOR_PAYMENT_APP_MESSAGE",
100    "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>",
101    "module": "rainmaker-ws",
102    "locale": "en_IN"
103 },
104 {
105     "code": "SW_PAY_PENDING_FOR_CONNECTION_ACTIVATION_SMS_MESSAGE",
106     "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>.",
107     "module": "rainmaker-ws",
108     "locale": "en_IN"
109 },
110 {
111     "code": "SW_PAY_PENDING_FOR_CONNECTION_ACTIVATION_APP_MESSAGE",
112     "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>",
113     "module": "rainmaker-ws",
114     "locale": "en_IN"
115 },
116 {
117     "code": "SW_ACTIVATE_CONNECTION_CONNECTION_ACTIVATED_SMS_MESSAGE",
118     "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>",
119     "module": "rainmaker-ws",
120     "locale": "en_IN"
121 },

```



```
14  "isStartState": true,
15  "isTerminateState": false,
16  "isStateUpdatable": false,
17  "actions": [
18    {
19      "action": "INITIATE",
20      "nextState": "INITIATED",
21      "roles": [
22        "SW_CEMP"
23      ]
24    }
25  ],
26 },
27 {
28   "sla": null,
29   "state": "INITIATED",
30   "applicationStatus": "INITIATED",
31   "docUploadRequired": false,
32   "isStartState": false,
33   "isTerminateState": false,
34   "isStateUpdatable": true,
35   "actions": [
36     {
37       "action": "SUBMIT_APPLICATION",
38       "nextState": "PENDING_FOR_APPROVAL",
39       "roles": [
40         "SW_CEMP"
41       ]
42     }
43   ],
44 },
45 {
46   "sla": 86400000,
47   "state": "PENDING_FOR_APPROVAL",
48   "applicationStatus": "PENDING_FOR_APPROVAL",
49   "docUploadRequired": false,
50   "isStartState": false,
51   "isStateUpdatable": true,
52   "isTerminateState": false,
53   "actions": [
54     {
55       "action": "APPROVE_CONNECTION",
56       "nextState": "APPROVED",
57       "roles": [
58         "SW_APPROVER"
59       ]
60     },
61     {
62       "action": "REJECT",
63       "nextState": "REJECTED",
64       "roles": [
65         "SW_APPROVER"
66       ]
67     }
68   ]
69 }
```

```

67     }
68   ]
69 },
70 {
71   "sla": null,
72   "state": "REJECTED",
73   "applicationStatus": "REJECTED",
74   "isStateUpdatable": false,
75   "docUploadRequired": false,
76   "isStartState": false,
77   "isTerminateState": true
78 },
79 {
80   "sla": null,
81   "state": "APPROVED",
82   "applicationStatus": "APPROVED",
83   "isStateUpdatable": false,
84   "docUploadRequired": false,
85   "isStartState": false,
86   "isTerminateState": true
87 }
88 ]
89 }
90 ]
91}

```

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 be 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

- eGov 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.

```

1{
2  "tenantId": "pb",
3  "moduleName": "sw-services-calculation",
4  "SCBillingSlab": [
5    {
6      "id": "1",
7      "buildingType": "RESIDENTIAL",
8      "calculationAttribute": "No. of water closets",
9      "connectionType": "Non Metered",
10     "minimumCharge": 0,
11     "slabs": [
12       {
13         "from": 0,
14         "to": 1000000000,
15         "charge": 15
16       }
17     ]
18   },
19   {
20     "id": "2",
21     "buildingType": "RESIDENTIAL",
22     "calculationAttribute": "No. of toilets",
23     "connectionType": "Non Metered",
24     "minimumCharge": 0,
25     "slabs": [
26       {
27         "from": 0,
28         "to": 1000000000,
29         "charge": 15
30       }
31     ]
32   },
33   {
34     "id": "3",
35     "buildingType": "NONRESIDENTIAL",
36     "calculationAttribute": "No. of water closets",
37     "connectionType": "Non Metered",
38     "minimumCharge": 0,
39     "slabs": [

```

```
40  {
41    "from": 0,
42    "to": 1000000000,
43    "charge": 30
44  }
45 ]
46 },
47 {
48   "id": "4",
49   "buildingType": "NONRESIDENTIAL",
50   "calculationAttribute": "No. of toilets",
51   "connectionType": "Non Metered",
52   "minimumCharge": 0,
53   "slabs": [
54     {
55       "from": 0,
56       "to": 1000000000,
57       "charge": 30
58     }
59   ]
60 },
61 {
62   "id": "5",
63   "buildingType": "Commercial",
64   "calculationAttribute": "No. of water closets",
65   "connectionType": "Non Metered",
66   "minimumCharge": 0,
67   "slabs": [
68     {
69       "from": 0,
70       "to": 1000000000,
71       "charge": 30
72     }
73   ]
74 },
75 {
76   "id": "6",
77   "buildingType": "Commercial",
78   "calculationAttribute": "No. of toilets",
79   "connectionType": "Non Metered",
80   "slabs": [
81     {
82       "from": 0,
83       "to": 1000000000,
84       "charge": 30
85     }
86   ]
87 },
88 {
89   "id": "7",
90   "buildingType": "Government",
91   "calculationAttribute": "No. of water closets",
92   "connectionType": "Non Metered",
```



```
93  "slabs": [
94    {
95      "from": 0,
96      "to": 1000000000,
97      "charge": 30
98    }
99  ],
100 },
101 {
102   "id": "8",
103   "buildingType": "Government",
104   "calculationAttribute": "No. of toilets",
105   "connectionType": "Non Metered",
106   "slabs": [
107     {
108       "from": 0,
109       "to": 1000000000,
110       "charge": 30
111     }
112   ],
113 },
114 {
115   "id": "9",
116   "buildingType": "Partly Commercial",
117   "calculationAttribute": "No. of water closets",
118   "connectionType": "Non Metered",
119   "slabs": [
120     {
121       "from": 0,
122       "to": 1000000000,
123       "charge": 25
124     }
125   ],
126 },
127 {
128   "id": "10",
129   "buildingType": "Partly Commercial",
130   "calculationAttribute": "No. of toilets",
131   "connectionType": "Non Metered",
132   "slabs": [
133     {
134       "from": 0,
135       "to": 1000000000,
136       "charge": 25
137     }
138   ],
139 },
140 {
141   "id": "11",
142   "buildingType": "RESIDENTIAL",
143   "calculationAttribute": "Flat",
144   "connectionType": "Non Metered",
145   "minimumCharge": 100,
```

```

146   "slabs": []
147 },
148 {
149   "id": "12",
150   "buildingType": "NONRESIDENTIAL",
151   "calculationAttribute": "Flat",
152   "connectionType": "Non Metered",
153   "minimumCharge": 250,
154   "slabs": []
155 },
156 {
157   "id": "13",
158   "buildingType": "Commercial",
159   "calculationAttribute": "Flat",
160   "connectionType": "Non Metered",
161   "minimumCharge": 250,
162   "slabs": []
163 },
164 {
165   "id": "14",
166   "buildingType": "Government",
167   "calculationAttribute": "Flat",
168   "connectionType": "Non Metered",
169   "minimumCharge": 350,
170   "slabs": []
171 },
172 {
173   "id": "15",
174   "buildingType": "Partly commercial",
175   "calculationAttribute": "Flat",
176   "connectionType": "Non Metered",
177   "minimumCharge": 200,
178   "slabs": []
179 }
180 ]
181}

```

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:

```
1{
2  "tenantId": "pb",
3  "moduleName": "sw-services-calculation",
4  "Interest": [
5    {
6      "rate": 5,
7      "minAmount": null,
8      "applicableAfterDays": 0,
9      "flatAmount": null,
10     "maxAmount": null,
11     "fromFY": "2019-20",
12     "startingDay": "1/01/2019"
13   }
14 ]
15}
```

Penalty

Below is a sample of master data JSON for penalty:

```
1{
2  "tenantId": "pb",
3  "moduleName": "sw-services-calculation",
4  "Penalty": [
5    {
6      "rate": 10,
7      "minAmount": null,
8      "applicableAfterDays": 0,
9      "flatAmount": null,
10     "fromFY": "2019-20",
11     "startingDay": "1/01/2019"
12   }
13 ]
14}
```

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.

```
1{
2  "category": "TAX",
3  "service": "SW",
4  "name": "Sewerage adhoc rebate",
5  "code": "SW_TIME_ADHOC_REBATE",
6  "isDebit": false,
7  "isActualDemand": true,
8  "order": "5",
9  "isRequired": false
10 },
11 {
12  "category": "TAX",
13  "service": "SW",
14  "name": "Sewerage adhoc penalty",
15  "code": "SW_TIME_ADHOC_PENALTY",
16  "isDebit": false,
17  "isActualDemand": true,
18  "order": "6",
19  "isRequired": false
20 },
```

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

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

```

14     },
15     "tenantId": "pb.amritsar"
16   }
17 ],

```

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:

```

1 "demandDetails": [
2   {
3     "id": "77ba1e93-a535-409c-b9d1-a312c409bd45",
4     "demandId": "687c3176-305b-461d-9cec-2fa26a30c88f",
5     "taxHeadMasterCode": "SEWERAGE_CHARGE",
6     "taxAmount": 120,
7     "collectionAmount": 0,
8     "additionalDetails": null,
9     "auditDetails": {
10      "createdBy": "04956309-87cd-4526-b4e6-48123abd4f3d",
11      "lastModifiedBy": "04956309-87cd-4526-b4e6-48123abd4f3d",
12      "createdTime": 1583675275873,
13      "lastModifiedTime": 1583675298705
14    },
15     "tenantId": "pb.amritsar"
16   },
17   {
18     "id": "0d83f4b0-6442-11ea-bc55-0242ac130003 ",
19     "demandId": "687c3176-305b-461d-9cec-2fa26a30c88f",
20     "taxHeadMasterCode": "SEWERAGE_CHARGE",
21     "taxAmount": 30,
22     "collectionAmount": 0,
23     "additionalDetails": null,
24     "auditDetails": {
25      "createdBy": "04956309-87cd-4526-b4e6-48123abd4f3d",
26      "lastModifiedBy": "04956309-87cd-4526-b4e6-48123abd4f3d",
27      "createdTime": 1583675275873,
28      "lastModifiedTime": 1583675298705
29    },
30     "tenantId": "pb.amritsar"
31   }
32 ],

```

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_ROUND OFF in demand detail can be greater than 0.5 but the sum of all SW_ROUND OFF 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.

The screenshot displays the 'Water & Sewerage' service interface. At the top, there are two main navigation options: 'Pay Water and Sewerage Bill' (indicated by a rupee symbol icon) and 'My Connections (29)' (indicated by a house icon). Below these, a light blue header bar reads 'Water & Sewerage'. The main content area is titled 'Search Water & Sewerage Connection' with a subtext 'Provide at least one parameter to search for a connection'. It features several input fields for searching: 'City *' with a dropdown menu labeled 'Select City', 'Property ID' with a text input labeled 'Enter Property ID', 'Mobile No.' with a text input labeled '+91 | Enter your mobile No', 'Consumer number' with a text input labeled 'Enter Consumer Number', and 'Old Consumer Number' with a text input labeled 'Enter Old Consumer Number'.

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.

Service
Consumer number
Status
Owner Name
Address
Due
PAY

b) Create new application

Apply For New Connection

My Applications (295)

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

Apply for New Water and Sewerage Connection

1

2

3

Connection DetailsDocumentsSummary

Property Details

Property ID *

Enter Property ID

SEARCH

Connection Details

Apply For *

☒ Water ☐ Sewerage

No. of taps proposed *

Enter No. of taps

Pipe Size proposed (in inches) *

Select Size

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

Service
Application number
Owner Name
Due
Status
[VIEW DETAILS](#)

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).

Task Status

Date

05/03/2020

Updated By

WSEMP2

Status

Pending For Payment

Pay

TAKE ACTION



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.

INR 105

06/02/2020 - 1

Consumer No


Owner Name :

Amount Paid :

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.

AMRITSAR MUNICIPAL CORPORATION

Amritsar

ENGLISH

SEARCH

Home

Complaints

Trade License

Reports

Fire Noc

Citizen Engagement

Property Tax

Dashboard

Miscellaneous Collection

Water & Sewerage

Search Water & Sewerage Application

Provide at least one parameter to search for an application

Consumer No.

Enter Consumer No.

Application Number.

SW_AP/107/2020-21/062040

Owner Mobile No.

+91 | Enter your mo

Application Type

Select Application Type

Application status

Select Application Status

From Date

dd-mm-yyyy

To Date

dd-mm-yyyy

RESET


SEARCH

Search Results for Water & Sewerage Application (1)

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)

AMRITSAR MUNICIPAL CORPORATION

Amritsar

ENGLISH

SEARCH

Home

Complaints

Trade License

Reports

Fire Noc

Citizen Engagement

Property Tax

Dashboard

Miscellaneous Collection

Water & Sewerage

Application for New Water and Sewerage Connection

Sewerage Application No. SW_AP/107/2020-21/062040

1

Connection Details

2

Documents

3

Additional Details

Required Documents

Only one file can be uploaded for one document.If multiple files need to be uploaded then please combine all files in a pdf and then upload

1

Identity Proof *

Select Documents *

Select Documents

2

Address Proof *

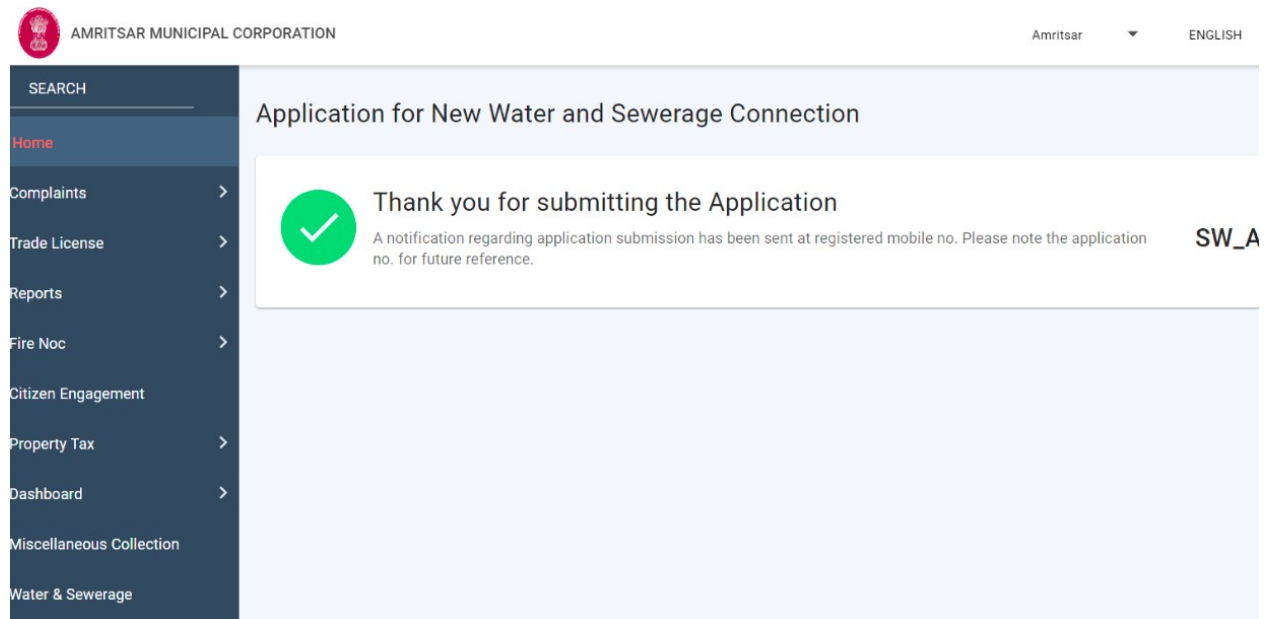
Select Documents *

Select Documents

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.

AMRITSAR MUNICIPAL CORPORATION | Amritsar | ENGLISH | DIGIT

Additional Details

Connection Details			
Connection Type: Non Metered	Number of Taps: 10	Water Source: SURFACE	Water Sub Source: LAKE
Pipe Size: 1			

Plumber Details			
Plumber Provided By: ULB	Plumber Licence No.: 10	Plumber Name: Shyam Sundar	Plumber Mobile No.: 6789098765

Road Cutting Charges	
Road Type: BRICKPAVING	Area (in sq ft): 10

Activation Details	
Connection Execution Date: 27/03/2020	

TAKE ACTION | **Verify and Forward**, **Send Back**, **Reject**, **Edit**

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.

AMRITSAR MUNICIPAL CORPORATION | Amritsar | ENGLISH | DIGIT

Application for New Water and Sewerage Connection

Application Verified and Forwarded Successfully

A notification regarding above application status has been sent to registered Mobile No.

Application Number, **SW_AP/107/2020-21/062040**

GO TO HOME

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.

AMRITSAR MUNICIPAL CORPORATION

Amritsar
ENGLISH

SEARCH
Home
Complaints
Trade License
Reports
Fire Noc
Citizen Engagement
Property Tax
Dashboard
Miscellaneous Collection
Water & Sewerage
Collapse

Payment Information

Consumer code WS_AP/107/2019-20/061950

Payment Collection Details

Fee Details		Total Amount
Tax and Cess	455	Rs 9555
Other Charge	325	
Scrutiny Fee	250	
Form Fee	25	
Road Cutting Charge	300	
Security Charge	8000	
One Time Fee	200	
Arrears	0	
Total Amount	9555	

Capture Payment

GENERATE RECEIPT >

AMRITSAR MUNICIPAL CORPORATION

Amritsar
ENGLISH

Complaints
Trade License
Reports
Fire Noc
Citizen Engagement
Property Tax
Dashboard
Miscellaneous Collection
Water & Sewerage
Collapse

Task Details

Sewerage Application No. SW_AP/107/2020-21/062040

DOWNLOAD
PRINT

Task Status

VIEW HISTORY

Date	Updated By	Status	Current Owner	Comments
03/04/2020	WSEMP2	Pending For Connection Activation	NA	

Fee Estimate

Application Fee	475	Total Amount Rs 10500
Service Fee	9525	
Tax	500	
Total Amount	Rs 10500	

VIEW BREAKUP
ADD REBATE/PENALTY

Activate Connection
Edit

TAKE ACTION


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

The screenshot displays the Amritsar Municipal Corporation portal. The header includes the logo, "AMRITSAR MUNICIPAL CORPORATION", and navigation links for "Amritsar", "ENGLISH", and "DIGIT". A left sidebar contains a "SEARCH" bar and a list of menu items: Home, Complaints, Trade License, Reports, Fire Noc, Citizen Engagement, Property Tax, Dashboard, Miscellaneous Collection, and Water & Sewerage. The main content area is titled "Payment Information" and shows a "Consumer code WS_AP/107/2019-20/061950". A green checkmark icon indicates "Payment has been collected successfully!". Below this, a message states: "A notification regarding Payment Collection has been sent to property owner at registered Mobile No.". To the right, the "Payment Receipt No." is "03/2019-20/004127". At the bottom right, there is an orange "HOME" button.

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

The screenshot displays the Amritsar Municipal Corporation portal. The header includes the logo, "AMRITSAR MUNICIPAL CORPORATION", and navigation links for "Amritsar", "ENGLISH", and "DIGIT". A left sidebar contains a "SEARCH" bar and a list of menu items: Home, Complaints, Trade License, Reports, Fire Noc, Citizen Engagement, Property Tax, Dashboard, Miscellaneous Collection, and Water & Sewerage. The main content area is titled "Application for New Water and Sewerage Connection". A green checkmark icon indicates "Connection Activated Successfully". Below this, a message states: "A notification regarding above application status has been sent to registered Mobile No.". To the right, the "Application Number." is "SW_AP/107/2020-21/062040". At the bottom right, there is an orange "GO TO HOME" button.

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.


AMRITSAR MUNICIPAL CORPORATION
Amritsar
ENGLISH

SEARCH
Home
Complaints
Trade License
Reports
Fire Noc
Citizen Engagement
Property Tax
Dashboard
Miscellaneous Collection
Water & Sewerage

Task Details

Sewerage Application No. SW_AP/107/2020-21/062040


Task Status

Date	Updated By	Status	Current Owner	Comi
03/04/2020	WSEMP2	Pending for Document Verification	NA	

Fee Estimate

Application Fee	475
Service Fee	9525
Tax	500

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.


AMRITSAR MUNICIPAL CORPORATION
Amritsar
ENGLISH

SEARCH
Home
Complaints
Trade License
Reports
Fire Noc
Citizen Engagement
Property Tax
Dashboard
Miscellaneous Collection

Search Water & Sewerage Application

Provide at least one parameter to search for an application

Consumer No. Enter Consumer No.	Application Number. SW_AP/107/2020-21/062040	Owner Mobile No. +91 Enter your mo
Application Type Select Application Type	Application status Select Application Status	From Date dd-mm-yyyy
To Date dd-mm-yyyy		

RESET
SEARCH

Search Results for Water & Sewerage Application (1)

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.



SEARCH

Home

Complaints >

Trade License >

Reports >

Fire Noc >

Citizen Engagement

Property Tax >

Dashboard >

Miscellaneous Collection

Water & Sewerage

Connections Details

Consumer No: SW/107/2020-21/062041

Service Details

Service	Connection Execution Date	Unit of Measurement	Number of Connections
SEWERAGE	03/05/2020		10

Property Details

Property Details

Property Type	Property Usage Type	Plot Size (in sq meters)
NA	RESIDENTIAL	1000

Property Location Details

Property ID	City	Plot / House / Survey No.	Building / Jumboc
PB-PT-2020-02-28-019537	amritsar	417/j	jumboc

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