

## Water & Sewerage Public Domain Search and Open Payment

### Objective

To provide facility to search properties without logging into the application and making the payment of pending due amount. This feature is developed only for citizen application.

### Implementation Details

This feature has a separate search screen. Citizen can enter the search criteria into the search form and search the properties. The properties obtained by the search operation will be listed in the search table. Below are the details of search criteria.

Field	Input type	Validation rule
ULB	Dropdown	Mandatory
Locality	Dropdown	Mandatory
Property ID	Input text	Optional
Owner mobile no	Input text	Optional
Consumer no	Input text	Optional

1. At least one optional field is required along with mandatory inputs to search the properties.
2. Locality dropdown list depends on the ULB selected.
3. After successful search, the search results will be displayed in the table as below.

Search Results for Water & Sewerage Connections (2)				
Service	Consumer No	Owner Name	Address	Connection Sta
WATER	WS/107/2020-21/068789	chander rao	Aggarsain Chowk to Mal Godown - Both Sides - Area3,Amritsar	Active
WATER	WS/107/2020-21/068801	chander rao	Aggarsain Chowk to Mal Godown - Both Sides - Area3,Amritsar	Active

### URI to access the screen

URI needs to add in the url after citizen - "/withoutAuth/wns/public-search"

<https://egov-micro-qa.egovernments.org/citizen/withoutAuth/wns/public-search>

### PAY option

User can have PAY button in the following scenarios


1. Property should be ACTIVE
2. Due amount should be more than 0
3. If the property is ACTIVE and Advance Payment is enabled then PAY option should present even if the due amount is 0.

On Clicking of PAY button, citizen navigates to common PAY screen. Common PAY screen is existing feature developed and being used across the entire module where payment is required. Below is the URL to navigate on common pay screen using routing method provided by react. Consumer code and tenant id are dynamic data. Below is the one such example.

```
/withoutAuth/egov-common/pay?consumerCode=WS_AP/107/2020-21/068817&tenantId=pb.amritsar&businessService=WS
```

In Public search common pay screen, side bar is hidden since user doesn't have access to enter other modules without authorization.

Citizen has to pay through payment gateway and all the required service are white-listed for the payment.



Payment Information

Application No. WS\_AP/107/2020-21/068817

Payment Collection Details

Fee Details

Road Cutting Charge	14760
One Time Fee	3690
Tax and Cess	952.5
Other Charge	325
Scrutiny Fee	250
Form Fee	25
Fee Round Off	0.5
Total Amount	20003

After successful payment, citizen navigates to acknowledgment screen. In this screen, the items which required authorization have been hidden. Here user can generate receipt pdf of the payment done. receipt pdf api is white listed.

Payment Information

Application No. WS\_AP/107/2020-21/068817



Payment has been collected successfully!

A notification regarding Payment Collection has been sent to the registered Mobile No. of the user/owner.

## Technical Implementation Details

Open Search screen is developed as part of WNS module. Below is the search screen configuration file link.

<https://github.com/egovernments/frontend/blob/master/web/rainmaker/dev-packages/egov-wns-dev/src/ui-config/screens/specs/wns/public-search.js>

Search screen table, API calls and other utility functions are written in the files under *publicSearchResource* folder. Below is the file description.

<https://github.com/egovernments/frontend/blob/master/web/rainmaker/dev-packages/egov-wns-dev/src/ui-config/screens/specs/wns/publicSearchResource/search-resources.js> - It has the screen configuration like input text, dropdowns, buttons etc.

<https://github.com/egovernments/frontend/blob/master/web/rainmaker/dev-packages/egov-wns-dev/src/ui-config/screens/specs/wns/publicSearchResource/search-methods.js> - It has the functions related to search API calls, reset forms etc.

<https://github.com/egovernments/frontend/blob/master/web/rainmaker/dev-packages/egov-wns-dev/src/ui-config/screens/specs/wns/publicSearchResource/publicSearchUtils.js> - This file has utility methods supporting api calls, form validation etc.

<https://github.com/egovernments/frontend/blob/master/web/rainmaker/dev-packages/egov-wns-dev/src/ui->

[config/screens/specs/wns/publicSearchResource/searchApplicationResult.js](#) - This file has functions that are responsible for making table definitions.

## MDMS Call details

We have to make few MDMS calls to get the required configuration related data. Below is the description of MDMS calls used in open search implementation.

- MDMS call to get the Tenants data. Tenants' data was supplied to ULB dropdown after screen initialization.

```
41 const getMDMSData = async dispatch => {
42   const mdmsBody = {
43     MdmsCriteria: {
44       tenantId: commonConfig.tenantId,
45       moduleDetails: [
46         {
47           moduleName: "tenant",
48           masterDetails: [
49             {
50               name: "tenants"
51             },
52             { name: "citymodule" }
53           ]
54         }
55       ]
56     }
57   };
58   try {
59     const payload = await httpRequest(
60       "post",
61       "/egov-mdms-service/v1/_search",
62       "_search",
63       [],
64       mdmsBody
65     );
66     payload.MdmsRes.tenant.tenants =
67       payload.MdmsRes.tenant.citymodule[1].tenants;
68     // console.log("payload--", payload)
69     dispatch(prepareFinalObject("searchScreenMdmsData", payload.MdmsRes));
```

## API White listing

Since this is open search and citizen can search the properties without logging in the application, certain APIs were required to be white listed. These APIs can return the result without authenticating the user. Below is the list of APIs that were white listed

1. `"/ws-services/wc/_search"`
2. `"/sw-services/swc/_search"`
3. `"/property-services/property/_search"`
4. `"/billing-service/bill/v2/_fetchbill"`

5. "/egov-location/location/v11/boundary/\_search?hierarchyTypeCode=REVENUE&boundaryType=Locality"
6. "pg-service/transaction/v1/\_create"
7. "pg-service/transaction/v1/\_update" + txnQuery
8. "/collection-services/payments/\_search"
9. "/pdf-service/v1/\_create"

## API call details

### 1. Boundary calls to get the Locality data

```
70         let payload = await httpRequest(  
71             "post",  
72             "/egov-location/location/v11/boundary/_search?hierarchyTypeCode=REVENUE&boundaryType=Locality",  
73             "_search",  
74             [{ key: "tenantId", value: action.value }],  
75             {}  
76         );  
77         const mohallaData = getMohallaData(payload, action.value);  
78         applyMohallaData(mohallaData, action.value, dispatch);
```

### 2. Search API Call for Water

URL: "/ws-services/wc/\_search"

method: "post"

QueryParams: [{key:"tenantId", value:"pb.amritsar"}, {key:"locality", value:"SUN02"}, {key:"searchType", value:"CONNECTION"},{key:"mobileNumber", value:"1234567890"}]

### 3. Search API Call for Sewarage

URL: "/sw-services/swc/\_search"

method: "post"

QueryParams: [{key:"tenantId", value:"pb.amritsar"}, {key:"locality", value:"SUN02"}, {key:"searchType", value:"CONNECTION"},{key:"mobileNumber", value:"1234567890"}]

### 4. Search API Call

URL: "/property-services/property/\_search"

method: "post"

QueryParams: [{key:"tenantId", value:"pb.amritsar"}, {key:"locality", value:"SUN02"}, {key:"name", value:"test"}]

## 5. Fetch Bill Call

URL: `/billing-service/bill/v2/_search`

method: `"post"`

QueryParams: [{key:"tenantId", value:"pb.amritsar"}, {key:"service", value:"WS"}, {key:"consumerCode", value:"WS\_AP/107/2020-21/068817"}]

### Localization Implementation

Search screen locale keys are pushed under rainmaker-pt locale module. Table header locale keys are in rainmaker-common locale module. Localization calls are made before screen initialization and all the locale keys are stored in local Storage.

While selecting the ULB, again a locale call is made for locality dropdown. This call returns the locale keys of every locality dropdown options.

URL: `/localization/messages/v1/_search`

method: `"post"`

queryParams: [{key:"module", value:"rainmaker-common,rainmaker-ws"}, {key:"locale", value:"en\_IN"}, {key:"tenantId", value:"pb"}]

### API Call Role Action Mapping

S.No.	API	Action id	Roles
1	/ws-services/wc/_search		CITIZEN
2	/sw-services/swc/_search		CITIZEN
3	/egov-location/location/v11/boundary/_search	1429	CITIZEN,
4	/billing-service/bill/v2/_fetchbill		CITIZEN
5	/localization/messages/v1/_search	1531	CITIZEN,
6	/property-services/property/_search	1897	CITIZEN
7	/pg-service/transaction/v1/_create	1571	CITIZEN
8	/pg-service/transaction/v1/_update	1572	CITIZEN
9	/collection-services/payments/_search	1864	CITIZEN
10	/pdf-service/v1/_create	1834	CITIZEN

