

Property Tax Document MDMS configuration

In this Document we see about MDMS configurations that are being used in the Property Create/update/Assess/Reassess flow.

In the MDMS call we will get the Dropdown values and List of Documents that is present while entering the Form .

Refer Property Actions for List of MDMS call present in Property Create flow,

<https://digit-discuss.atlassian.net/wiki/spaces/EGR/pages/406618320/Property+Actions>.

Here we see more about MDMS call for Document.

It is being used in two screens,

1. Document Required popup
2. Document Upload screen (Step 4 in Creating Property)

In both places it will make call using this MDMS criteria

```
1 "MdmsCriteria": {
2   "tenantId": "pb",
3   "moduleDetails": [
4     {
5       "moduleName": "PropertyTax",
6       "masterDetails": [
7         {
8           "name": "Documents"
9         }
10      ]
11    }
12  ]
13 }
```

Reference:

[egov-mdms-data/Documents.json at master · egovernments/egov-mdms-data](#)

1. Document Required Popup

It is a Popup shows user before applying about what all documents required for Creating a Property. Its Configuration is present in the - dev-packages\egov-pt-dev\src\ui-config\screens\specs\pt-mutation\requiredDocuments\reqDocs.js

[frontend/reqDocs.js at 0024529285b09a257154e15c5202f2f084fd6469 · egovernments/frontend](#)

2. Document Upload Screen,

It is the last screen before summary while creating a Property; here it collects document information to verify the entered User Information is Valid.

Here we see detail about each configuration in the MDMS Documents.
Each Document value will have following structure

```
1 {
2   "code": "OWNER.SPECIALCATEGORYPROOF",
3   "documentType": "OWNER",
4   "required": true,
5   "active": true,
6   "hasDropdown": true,
7   "additionalDetails": {
8     "filterCondition": {
9       "filterValue": [
10        "NONE",
11        null
12      ],
13      "jsonPath": "Properties[0].propertyDetails[0].owners",
14      "onArray": true,
15      "arrayAttribute": "ownerType"
16    },
17    "dropdownFilter": {
18      "parentArrayJsonPath": "Properties[0].propertyDetails[0].owners",
19      "parentJsonpath": "ownerType"
20    },
21    "enabledActions": {
22      "assess": {
23        "disableUpload": true,
24        "disableDropdown": true
25      },
26      "reassess": {
27        "disableUpload": true,
28        "disableDropdown": true
29      },
30      "update": {
31        "disableUpload": true,
32        "disableDropdown": true
33      },
34      "create": {
35        "disableUpload": false,
36        "disableDropdown": false
37      }
38    }
39  },
40  "dropdownData": [
41    {
42      "code": "OWNER.SPECIALCATEGORYPROOF.SERVICEDOCUMENT",
```

```

43     "active": true,
44     "parentValue": [
45       "FREEDOMFIGHTER",
46       "DEFENSE"
47     ]
48   },
49   {
50     "code": "OWNER.SPECIALCATEGORYPROOF.HANDICAPCERTIFICATE",
51     "active": true,
52     "parentValue": [
53       "HANDICAPPED"
54     ]
55   }
56 ],
57 "description":
"OWNER.SPECIALCATEGORYPROOF.SPECIALCATEGORYPROOF_DESCRIPTION"
58 },

```

In Each Document we have configurations

1. Attribute "enabledActions" we can use for disable / enable document upload. if we want document editable during a particular action i.e. for 'Update Property' it is configured as it means document type will be disabled and document upload option also will be disabled. similarly for 'assess','reassess','create'.

```

1     "update": {
2       "disableUpload": true,
3       "disableDropdown": true
4     },

```

2. Attribute "dropdownFilter" it can be used for filtering all the options for each document in document upload screen based on previous value that is being entered in previous screens. it is configured as

```

1 "dropdownFilter": {
2   "parentArrayJsonPath": "Properties[0].propertyDetails[0].owners",
3   "parentJsonpath": "ownerType"
4 },

```

here parentArrayJsonPath is the JSON path of the Array element if we need to check the values in an array
if it is not a type array then use "parentJsonpath".

```

1 "dropdownFilter": {
2   "parentJsonpath": "Properties[0].propertyDetails[0].usageCategoryMinor"
3 },

```

once the value is fetched then it will compare with each option of the dropdown

here "parentValue" is an array which will have all the values for which it should be shown.

```
1 "parentValue": [  
2   "FREEDOMFIGHTER",  
3   "DEFENSE"  
4 ]
```

3. Similar to the Dropdown filter we have option to show / hide document upload option based on previous values.

Configuration for that is as follow

```
1 "filterCondition": {  
2   "filterValue": [  
3     "NONE",  
4     null  
5   ],  
6   "jsonPath": "Properties[0].propertyDetails[0].owners",  
7   "onArray": true,  
8   "arrayAttribute": "ownerType"  
9 },
```

here "filterValue" is an array which will have all the values for which it should be shown.

"jsonPath" corresponds to the objects path where it should be compared with.

"onArray" is true if it needs to be compared with an Array of objects "arrayAttribute" the value corresponds to the array attribute from which values will be compared

Based on this document will be shown / hidden

4. "dropdownData" will have the following structure

here code corresponds to the Label of the dropdown option

active corresponds to option is active / inactive

parent value used for filtering the options present in the document upload.

```
1 "dropdownData": [  
2   {  
3     "code": "OWNER.SPECIALCATEGORYPROOF.SERVICEDOCUMENT",  
4     "active": true,  
5     "parentValue": [  
6       "FREEDOMFIGHTER",  
7       "DEFENSE"  
8     ]  
9   },
```

```
10 {
11   "code": "OWNER.SPECIALCATEGORYPROOF.HANDICAPCERTIFICATE",
12   "active": true,
13   "parentValue": [
14     "HANDICAPPED"
15   ]
16 }
17 ],
```