



Ministry of Housing and Urban Affairs
Government of India



Ministry of Housing and Urban Affairs
Government of India



ET Panache

National Urban Digital Mission

Building cities that work for people



Master Data Setup

Outline

1. How MDMS works
2. How to configure a tenant
3. How to add a new Master
4. State level masters v/s ULB Level masters
5. MDMS in a multi environment setup

How MDMS works

- MDMS stands for Master Data Management Service. MDMS is One of the applications in the UPYOG core group of services.
- This service aims to reduce the time spent by developers on writing codes to store and fetch master data (primary data needed for module functionality) which doesn't have any business logic associated with them.
- The MDMS service reads the data from a set of JSON files from a pre-specified location.
- It can either be an online location (readable JSON files from online) or offline (JSON files stored in local memory).
- Using MDMS we can configure data for [logical calculations](#) or decision making.

How MDMS works Contd..

- The JSON files will be in a prescribed format and store the data on a map.
- The **tenantID** of the file serves as a key and a map of master data details as values.
- Once the data is stored in the map the same can be retrieved by making an API request to the MDMS service. Filters can be applied in the request to retrieve data based on the existing fields of JSON. Ex,

```
{  
  "tenantId": "uk",  
  "moduleName": "BillingService",  
  "{$MasterName}": [ ]  
}
```


How to configure a tenant

- Tenant represents a body in a system. In the municipal system, a state and its ULBs (Urban local bodies) are tenants. ULB represents a city or a town in a state. Tenant configuration is done in MDMS.
- Tenant is added in tenant.json. In MDMS, file **tenant.json**, under **tenant** folder holds the details of state and ULBs to be added in that state.

```
{  
  "tenantId": "uk", //<ReplaceWithDesiredTenantId>  
  "moduleName": "tenant",  
  "tenants": [ {  
    "code": "uk.citya", //<state.ulbname>  
    "name": "City A", //<name of the ulb>  
    "description": "City A", //<ulb description>
```

How to configure a tenant

```
"imageId": null,  
"domainUrl": "", //<ulb website url>  
"type": "CITY",  
"twitterUrl": null,  
"facebookUrl": null,  
"emailId": "complaints.citya@gmail.com", //<ulb email id>  
"OfficeTimings": {  
  "Mon - Sat": "10.00 AM - 5.00 PM"  
},  
"city": {  
  "name": "City A",  
  "localName": null,
```

How to configure a tenant

```
"districtCode": "CITYA",  
"districtName": null,  
"regionName": null,  
"ulbGrade": "Municipal Corporation",  
"longitude": 78.0322,  
"latitude": 30.3165,  
"shapeFileLocation": null,  
"captcha": null,  
"code": "248430"  
}, "address": "City A Municipal Corporation Address",  
"contactNumber": "91 (135) 2653572"}]}
```


How to add a new Master

- For creating a new master in MDMS, create the JSON file with the master data and configure the newly created master in the master config file.
- The new JSON file needs to contain 3 keys as shown in the below code snippet.

```
{  
  "tenantId": "< TENANT ID >",  
  "moduleName": "< MODULE NAME >",  
  "< MASTER NAME >": []  
}
```

How to add a new Master

- The new master can be created for Statewide or ULB wise. Tenant id and config in the master config file determines this.
- The Master config file is structured as below. Each key in the Master config is a module and each key in the module is a master.

```
{  
  "<module1>": {  
    "<master1>": {},  
    "<master2>": {},  
    ...  
  },  
  ...  
}
```

How to add a new Master

- Each master contain the following data and keys are self-explanatory

```
"master": {  
  "masterName": "<>",  
  "isStateLevel": true,  
  "uniqueKeys": []  
}
```

State level masters v/s ULB Level masters

- MDMS supports the [configuration](#) of data at different levels.
- While enable a state there can be data that is common to all the ULB's of the state
- The data further can be configured at each module level as state-specific or ULB's specific.
- State Level Masters are maintained in a common folder.
- ULB Level Masters are maintained in separate folders named after the ULB.
- Module Specific State Level Masters are maintained by a folder named after the specific module that is placed outside the common folder.

Master Data Collection Templates

- To help implementation parties and end-users in providing the required data in minimal interaction and iterations and ensure the quality, consistency and shape of data needed to configure into the system.

Data gathering activities.

- State Team
- Onsite Team/ Implementation Team
- ULB Team (Nodal and DEO)
- Implementation Partner

Master Data Collection Templates

State Level Setup : Key configurations at the state level include -

- Sms configuration
- eMail configuration
- Payment gateway integration
- Domain name, ssl configurations
- Department, designation, Pos machine integration
- Google Map, google play store account
- Localization

ULB Level Setup : Key configurations at the state level include -

- Boundary, Hierarchy, Bank account, Cross boundary map

Module specific master data templates

Property Tax Module

- State Level: Property type, sub type, usage, construction type, ownership type, ownership special categories, document types, road type etc.
- ULB level : [Tax rate](#), Unit rate, interest rate, penalty rate, rebate rates, mutation fee



Thank You

“UPYOGal Transformation is more about humans than UPYOGal”

