Ministry of Housing and Urban Affairs
Government of India

Ministry of Housing and Urban Affairs
Government of India

National Urban Digital Mission

Building cities that work for people

EDCR Setup

Outline

1. EDCR Overview
2. EDCR Configurations and Setup
   - 2.1. How to set up a client implementation repository
   - 2.2. How to override rule processing across the state for a feature
   - 2.3. How to override rule processing district, city, and grade wise for a feature
   - 2.4. Configuration Changes to setup State and Cities
   - 2.5. Changes required for local machine workspace setup
   - 2.6. MDMS Integration

EDCR Overview

- Setting up an instance of an application server and configuring customer-specific rules, and data, etc are a few of the key activities.
- Centralized Server hosting all the ULBs within a state
- All ULBs access the software over API calls.
- Uniform code base supporting all the ULBs for the state. City-specific changes are maintained using client-specific implementation repositories.
- A separate schema for each ULB in the database.

EDCR Configurations and Setup

- eDCR Service repository will be used to define default rules. The statewide rules to be defined within the client implementation repository.

How to set up a client implementation repository

- The default client implementation project is available in the https://github.com/nugp-digit/DIGIT-NUGP/tree/master/edcr/client
- For each state have to create a separate client repository to setup the building bye-laws of the state
- After creating a client implementation repository, If you want to override rules processing for features, then create a project under egov directory like **egov-client-impl.**
- The client-specific rules are configurable in the individual client implementation module. Here the rules are fetched by state wise, district wise, ULB wise, or grade-wise.

How to override rule processing across the state for a feature

- The EG_CITY table, master data is used to decide the rules.
- If the rules are the same for a feature across the state then the filename must need to keep like Far_{Client.id}.
- Here **client id** nothing but a state name, the client.id need to update with the state name in **application-config-client.properties** file or in **egov-erp-override.properties** file in the wildfly
- The client id used with the feature class name and configures in the properties file should match, then only the system will pick features and process otherwise it won't pick the file.

How to override rule processing district, city, and grade wise for a feature

- If the rules varying across district, city, and grade wise within a state then must need to follow the following naming conventions.
- If the rules are varying from district to district, city to city, then the respective district-related rules need code under separate files with a filename like Far_{District Name}, Far_{City Name}, Far_{Grade}. eg: Far_Udalguri, Far_Dispur, Far_Corp.
- The district, city, and grade information will be reading from the eg_city table, so in this table need to update that information before starting the coding.

Configuration Changes to setup State and Cities

- The state is configured by adding property **tenant.{domain_name}=schema_name (state_name)** in egov-erp-override.properties**.**
- Each new ULB is enabled by adding a schema name and domain name in egov-erp-override.properties file
- Each ULB can be configured by adding an entry like **tenant.{domain_name}=schema_name (city_name)** in **egov-erp-override.properties** file.
- Insert data into eg_city, in the city table domain URL value should be the same as configured tenant domain_name in the **egov-erp-override.properties**.

Changes required for local machine workspace setup

- Setup your eDCR service workspace by following the instructions provided in https://github.com/UPYOG-UPYOG/UPYOG-UPYOG/blob/master/edcr/service/README.md
- After set up is done, in the local ubuntu machine need to update the domain URLs in the host file which you are going to use for scrutinizing and fetching the plan.
- Navigate to the root directory and from there open the host config file available in the location 'etc/hosts'. Map the domain URLs with a local IP address in the hosts file and save the changes.

Changes required for local machine workspace setup

- Add 'max-post-size' attribute with value 100mb in bytes in the below location under wildfly standalone.xml file,
- <server name="default-server"> <http-listener name="default" socket-binding="http" max-post-size="104857600" redirect-socket="https" enable-http2="true"/> <https-listener name="https" max-post-size="104857600" socket-binding="https" security-realm="ApplicationRealm" enable-http2="true"/> <host name="default-host" alias="localhost"> <location name="/" handler="welcome-content"/> <http-invoker security-realm="ApplicationRealm"/> </host> </server>

MDMS Integration

- If you want to fetch master data from MDMS for following ApplicationType, ServiceType, OccupancyType, SubOccupancyType, Usages then the following configurations are needed
- Enable fetch master data from MDMS by adding **mdms.enable=true** property in **application-config-client.properties** file
- By default, the master data will be fetched from the database

Note

- After setup is done APIs one must use state domain URL and in the API request the tenantID of the respective city must send to scrutinize multiple cities.
- One should not use the city domain URL to scrutinize or fetch plan if used that way, the response will be empty.
- The tenantId used should follow {state_name.city_name} naming convention, then the state_name passed in the request and city code in the state schema must be the same.
-

Thank You

"UPYOGal Transformation is more about humans than UPYOGal"