

Billing Slabs

Objective

The objective of this service is to provide APIs to create update and search the billing slabs. These slabs are defined at module level. PT will have its own billing-slab APIs so will TL so will any other module. Conventionally, these APIs are part of the respective calculator service.

Requirements

- Prior Knowledge of Java/J2EE
- Prior Knowledge of SpringBoot
- Prior Knowledge of REST APIs and related concepts like path parameters, headers, JSON etc
- Prior Knowledge of Kafka and related concepts like Producer, Consumer, Topic etc.

Setup

PT

Step 1: Start the egov-mdms service present in the following repo:

<https://github.com/egovernments/core-services/tree/master/egov-mdms-service> - Connect to preview

Step 2: Start the pt-calculator service present in the following repo:

<https://github.com/egovernments/municipal-services/tree/master/pt-calculator-v2> - Connect to preview

TL

Step 1: Start the egov-mdms service present in the following repo:

<https://github.com/egovernments/core-services/tree/master/egov-mdms-service> - Connect to preview

Step 2: Start the tl-calculator service present in the following repo:

<https://github.com/egovernments/municipal-services/tree/master/tl-calculator>

- Connect to preview

Configurable Properties

Following are the properties in application.properties file in egov-user-events service which are configurable.

Property	Value	Remarks
kafka.topics.save.service	save-tl-billingslab OR save-pt-billingslab	This is the persister topic onto which calculator pushes record persistence. This is for creating slabs.
kafka.topics.update.service	update-tl-billingslab OR update-pt-billingslab	This is the persister topic onto which calculator pushes record persistence. This is for updating slabs.

Entities

BillingSlab: Model to capture the billing-slab information. This includes all the fields that are to be captured in order to define a slab. This depends on the module and also the state.

How does it work?

PT

Property Tax is calculated based on various factors that are subjected to slabs, time of payment, type of owner, ULB etc. Billing slabs are a set of factors that play a significant role in the Property Tax calculation. Currently in Punjab, the following parameters constitute each billing slab - *propertyType*, *propertySubType*, *usageCategory*, *ownershipCategory*, *subOwnershipCategory*, *areaType*, *fromPlotSize*, *toPlotSize*, *occupancyType*, *fromFloor*, *toFloor*, *unitRate*,

unBuiltUnitRate, arvPercent.

These fields keep changing from state to state based on the implementation. These factors cumulatively are mapped to a certain amount which is accounted for while calculating the property tax.

Note - please ensure that overlapping slabs are not created, the factors involving range like *fromFloor and toFloor* are checked for overlaps, have clearly defined inclusion and exclusion. Currently, ranges are inclusive, which means if a slab is defined for *fromFloor = 2 and toFloor = 5*, it is applicable for floors 2,3,4 and 5.

TL

Trade License is calculated based on various factors that are subjected to slabs, time of payment, type of owner, ULB etc. Billing slabs are a set of factors that play a significant role in the Trade license fee calculation. Currently in Punjab, the following parameters constitute each billing slab - *licenseType, structureType, tradeType, accessoryCategory, type, uom, fromUom, toUom, rate.*

These fields keep changing from state to state based on the implementation. These factors cumulatively are mapped to a certain amount which is accounted for while calculating the trade license fee.

Note - please ensure that overlapping slabs are not created, the factors involving range like *fromUom and toUom* are checked for overlaps, have clearly defined inclusion and exclusion. Currently, ranges are inclusive, which means if a slab is defined for *fromUom = 2 and toUom = 5*, it is applicable for floors 2,3,4 and 5.

The billing slab APIs for both PT and TL are present in pt-calculator and tl-calculator respectively. The calculation logic makes use of these APIs to fetch the slabs. It is to be noted that every slab has to be unique, for a given PT or TL application, there cannot be more than one billing slabs retrieved, in that case, the calculation throws an error.

APIs

Create - API to create billing slabs

Update - API to update billing slabs

Search - API to search billing slabs based on different criteria

API Contract

PT - <https://github.com/egovernments/egov-services/blob/master/docs/rainmaker/property-tax/property-calculation-service.yml>

TL - <https://github.com/egovernments/egov-services/blob/master/docs/rainmaker/trade-license/tl-calculator.yml>

Postman Collection

PT - <https://www.getpostman.com/collections/9e0899483efc051f6a89>

TL - <https://www.getpostman.com/collections/1e0e1b9b07d34e3732c7>

Configs

MDMS Data

Factors in billing slabs need to be validated against the respective MDMS data, MDMS for billing-slabs is owned by the module itself.

PT - <https://github.com/egovernments/egov-mdms-data/tree/master/data/pb/PropertyTax> - Connect to preview

TL: <https://github.com/egovernments/egov-mdms-data/tree/master/data/pb/TradeLicense> - Connect to preview

Some of the values are also defined as enums in the system, which can be found in the calculator services. Few factors like to Floor etc are absolute and are not configured anywhere.

Roleaction Configs

/_create: SUPERUSER, EMPLOYEE, <PT ADMIN roles>, <TL ADMIN roles>
/_update: SUPERUSER, EMPLOYEE, <PT ADMIN roles>, <TL ADMIN roles>
/_search: CITIZEN, SUPERUSER, EMPLOYEE, <PT ADMIN roles>, <TL ADMIN roles>