

# Property Registry Migration

In property registry migration, the data of old property structure is transferred into new property structure.

**API endpoint/property-services/property/\_migration?offset=0&limit=500**

The API endpoint migrates data from old property structure into new property structure. Here limit is the number of records which is migrated and offset is the starting point of record for example if in database table there are 100 records and offset is set as 10 and limit is set as 50 this means that starting from 10th record in database table 50 records are fetch from database table and migrated into new property object.

**NOTE** In offset 0 means first record.

Once the API is hit the migration process gets started and continues till all the record gets migrated. In this the records is migrated in a batch which equal to value of limit which is send during API calling.

## Build and Persister config

Property service build: - property-services-db:42-PT\_REGISTRY\_MIGRATION\_V2-21ab4a22

pt-services-v2 build :- pt-services-v2-db:21-PT\_V2\_Migration\_Changes-a72827dc

Persister Service build: - egov-persister:5-PERSISTER\_BULK-ee2132d

Persister Config:-

<https://github.com/egovernments/configs/blob/master/egov-persister/property-services-migration-temp-config.yml> ,  
<https://github.com/egovernments/configs/blob/master/egov-persister/assessment-persister-migration-temp.yml> and

<https://github.com/egovernments/configs/blob/master/egov-persister/migration-batch-count-persister.yml>

As there are lots of records to persist into database table, we are using bulk persister which persist 100 records at a time. For bulk persister, kafka topic in which property and assessment record are push, such topic must have "-batch" as a suffix.

For example, in the environment variable file (or application.properties file) of property registry service module, the value of PERSISTER\_SAVE\_PROPERTY\_TOPIC must be equals to save-property-registry-batch

## Input

```
{
  "RequestInfo": {
    "apiId": "Rainmaker",
    "ver": ".01",
    "ts": "",
    "action": "_create",
    "did": "1",
    "key": "",
    "msgId": "20170310130900|en_IN",
    "authToken": "6eaaa85d-5553-4f28-a097-5f8016033873",
    "userInfo": {
      "id": 24226,
      "userName": "amr001",
      "salutation": null,
      "name": "leela",
      "gender": "FEMALE",
      "mobileNumber": "9814424443",
      "emailId": "leela@llgmail.com",
      "altContactNumber": null,
      "pan": "AINP09087Y",
      "aadhaarNumber": null,
      "permanentAddress": "Maahi Street,Jalandhar",
      "permanentCity": "Phagwara",
      "permanentPinCode": "24324",
      "correspondenceAddress": "pb.amritsar",
      "correspondenceCity": "Pink city",
      "correspondencePinCode": "34353",
      "addresses": [
        {
          "pinCode": "34353",
          "city": "Pink city",
          "address": "pb.amritsar",
          "type": "CORRESPONDENCE",
          "id": 48238,
          "tenantId": "pb.amritsar",
```

```
        "userId": 24226,  
        "addressType": "CORRESPONDENCE",  
        "lastModifiedBy": null,  
        "lastModifiedDate": null  
    },  
    {  
        "pinCode": "24324",  
        "city": "Phagwara",  
        "address": "Maahi Street,Jalandhar",  
        "type": "PERMANENT",  
        "id": 48239,  
        "tenantId": "pb.amritsar",  
        "userId": 24226,  
        "addressType": "PERMANENT",  
        "lastModifiedBy": null,  
        "lastModifiedDate": null  
    }  
],  
"active": true,  
"locale": null,  
"type": "EMPLOYEE",  
"accountLocked": false,  
"accountLockedDate": 0,  
"fatherOrHusbandName": "Fathe",  
"signature": null,  
"bloodGroup": null,  
"photo": null,  
"identificationMark": null,  
"createdBy": 24113,  
"lastModifiedBy": 1,  
"tenantId": "pb.amritsar",  
"roles": [  
    {  
        "code": "TL_APPROVER",  
        "name": "TL Approver",  
        "tenantId": "pb.amritsar"  
    },  
    {  
        "code": "SUPERUSER",  
        "name": "Super User",  
        "tenantId": "pb"  
    },  
    {  
        "code": "EMPLOYEE",  
        "name": "Employee",  
        "tenantId": "pb.amritsar"  
    },  
    {  
        "code": "FEMP",  
        "name": "Field Employee",  
        "tenantId": "pb.amritsar"  
    },  
    {  
        "code": "TL_CEMP",  
        "name": "TL Counter Employee",
```

```

        "tenantId": "pb.amritsar"
      },
      {
        "code": "TL_FIELD_INSPECTOR",
        "name": "TL Field Inspector",
        "tenantId": "pb.amritsar"
      },
      {
        "code": "SUPERUSER",
        "name": "Super User",
        "tenantId": "pb.amritsar"
      },
      {
        "code": "TL_CREATOR",
        "name": "TL Creator",
        "tenantId": "pb.amritsar"
      },
      {
        "code": "COLL_OPERATOR",
        "name": "Collection Operator",
        "tenantId": "pb.amritsar"
      },
      {
        "code": "TL_DOC_VERIFIER",
        "name": "TL doc verifier",
        "tenantId": "pb.amritsar"
      }
    ],
    "uuid": "11b0e02b-0145-4de2-bc42-c97b96264807",
    "createdDate": "25-09-2018 10:47:41",
    "lastModifiedDate": "01-12-2019 21:57:20",
    "dob": "1/2/2002",
    "pwdExpiryDate": "24-12-2018 10:47:41"
  }
}
}
}

```

## Output

```

{
  "propertID": "Message",
  "PT-107-000343": "SUCCESS",
  "PT-107-000344": "SUCCESS",
  "PT-107-000345": "SUCCESS"
  .....
  .....
}

```

## Postman Collection

<https://www.getpostman.com/collections/3b8f340c10286d620b34>

Below is the mapping of fields between old property structures to new property structure with some remarks

Fields of new property structure	Fields of old property structure	Remarks
Properties.id	---	Set as random UUID
Properties.propertyId	Properties.propertyId	
Properties.surveyId	---	
Properties.linkedProperties	---	
Properties.tenantId	Properties.tenantId	
Properties.accountId	RequestInfo.userInfo.uuid	
Properties.oldPropertyId	Properties.oldPropertyId	
Properties.status	Properties.status	

Properties.acknowledgementNumber	Properties.acknowledgementNumber	
Properties.propertyType	Properties.propertyDetails[0].propertyType	
Properties.ownershipCategory	Properties.propertyDetails[0].ownershipCategory,  Properties.propertyDetails[0].subOwnershipCategory	Append the value of two jsonpath with “.”
Properties.creationReason	Properties.creationReason	
Properties.usageCategory	Properties.propertyDetails[0].usageCategoryMajor,  Properties.propertyDetails[0].usageCategoryMinor	
Properties.noOfFloors	Properties.propertyDetails[0].noOfFloors	
Properties.landArea	Properties.propertyDetails[0].landArea	
Properties.superBuiltUpArea	Properties.propertyDetails[0].builtUpArea	
Properties.source	Properties.propertyDetails[0].source	

Properties.channel	Properties.propertyDetails[0].channel	
Properties.documents	Properties.propertyDetails[0].documents	All field of document in new property object matches field of document in old property object
Properties.additionalDetails	Properties.propertyDetails[0].additionalDetails	
Properties.auditDetails	Properties.propertyDetails[0].auditDetails	
Properties.address.tenantId	Properties.address.tenantId	
Properties.address.doorNo	Properties.address.doorNo	
Properties.address.plotNo	---	
Properties.address.id	Properties.address.id	
Properties.address.landmark	Properties.address.landmark	
Properties.address.city	Properties.address.city	

Properties.address.district	---	
Properties.address.region	---	
Properties.address.state	---	
Properties.address.country	---	
Properties.address.pincode	Properties.address.pincode	
Properties.address.buildingName	Properties.address.buildingName	
Properties.address.street	Properties.address.street	
Properties.address.locality	Properties.address.locality	All field of locality in new property object matches field of locality in old property object
Properties.address.geoLocation.latitude	Properties.address.latitude	



Properties.address.geoLocation.longitude	Properties.address.longitude	
Properties.address.additionalDetails	Properties.address.additionalDetails	
Properties.owners	Properties.propertyDetails[0].owners	The owner object in new property structure is same as the owner object in old property structure with two extra field ownerInfoUuid and status.
Properties.owners.ownerInfoUuid	Properties.propertyDetails[0].owners.ownerInfoUuid	
Properties.owners.status	<b>ACTIVE</b>	
Properties.institution.id	Properties.propertyDetails[0].institution.id	
Properties.institution.tenantid	Properties.propertyDetails[0].institution.tenantid	
Properties.institution.name	Properties.propertyDetails[0].institution.name	

Properties.institution.type	Properties.propertyDetails[0].institution.type	
Properties.institution.designation	Properties.propertyDetails[0].institution.designation	
Properties.institution.nameOfAuthorizedPers	---	
Properties.institution.aadditionalDetails	Properties.propertyDetails[0].institution.aadditionalDetails	
Properties.unit	Properties.propertyDetails[0].unit[*]	The unit object in new property structure is same as old except two field construction Detail and auditDetails
Properties.unit.auditDetails	---	
Properties.unit.constructionDetail.builtUpArea	Properties.propertyDetails[0].unit[*].unitArea	

The migrated data is pushed into eg\_pt\_property, eg\_pt\_owner, eg\_pt\_document, eg\_pt\_institution, eg\_pt\_address, eg\_pt\_unit .

Also for the assessment object, its field is set from old property object and the migrated data is pushed into following table eg\_pt\_asmt\_assessment, eg\_pt\_asmt\_unitusage, eg\_pt\_asmt\_document.

## Note

1) During migration kafka producer request size and consumer fetch max byte property has to increase to process faster. Otherwise storing data in DB will be slow.

2) Before migration disables the notification service otherwise after migration user will get the unnecessary SMS on mobile.

## Indexing

As part of property registry migration, from now on two new indexes will be used for property and property assessment records respectively.

The new indexer config can be found from <https://github.com/egovernments/configs/blob/master/egov-indexer/property-services.yml>

Also detailed information about new indexes can be obtained from [Property Service Re-indexing](#)

## Bulk reindexing records directly from DB

**Note:** the property service build should be **property-services-db:159-PT\_SERVICES\_FEATURE-8527613a** or newer

After all the properties are migrated in the DB, to directly index data from DB, run commands mentioned next through playground pod, or use other method to hit indexer service with same request, ex:- port forwarding indexer service port to local

**Note:-** After reindexing is done, for checking correctness of reindexing please match count of following DB tables with count of records on new indexes, total

count from eg\_pt\_asmt\_assessment → "property-assessments" index distinct  
propertyids in eg\_pt\_property → "property-services" index

## 1. Property Index

index name- "property-services"

```
curl -X POST \
http://egov-indexer:8080/egov-indexer/index-operations/_legacyindex \
H 'Cache-Control: no-cache' \
H 'Content-Type: application/json' \
H 'Postman-Token: d380bebb-383f-1b7c-76d1-10c1dc07dd06' \
d '{
  "RequestInfo": {
    "apiId": "string",
    "ver": "string",
    "ts": null,
    "action": "string",
    "did": "string",
    "key": "string",
    "msgId": "string",
    "authToken": "b843ef27-1ac6-49b8-ab71-cd0c22f4e50e",
    "correlationId": "e721639b-c095-40b3-86e2-acecb2cb6efb",
    "userInfo": {
      "id": 23299,
      "uuid": "e721639b-c095-40b3-86e2-acecb2cb6efb",
      "userName": "9337682030",
      "name": "Abhilash Seth",
      "type": "EMPLOYEE",
      "mobileNumber": "9337682030",
      "emailId": "abhilash.seth@gmail.com",
      "roles": [
        {
          "id": 281,
          "name": "Employee"
        }
      ]
    }
  },
  "apiDetails": {
    "uri": "http://property-services:8080/property-services/property/_plainsearch",
    "paginationDetails": {
      "offsetKey": "offset",
```

```

"sizeKey": "limit",
"maxPageSize": 50,
"startingOffset": 0
},
"responseJsonPath": "$.Properties"
},
"legacyIndexTopic": "property-registry-legacyIndex",
"tenantId": "pb.amritsar"
}'

```

## 2. Assessment Index

index name- "property-assessments"

```

curl -X POST \
http://egov-indexer:8080/egov-indexer/index-operations/_legacyindex \
H 'Cache-Control: no-cache' \
H 'Content-Type: application/json' \
H 'Postman-Token: d380bebb-383f-1b7c-76d1-10c1dc07dd06' \
d '{
  "RequestInfo": {
    "apiId": "string",
    "ver": "string",
    "ts": null,
    "action": "string",
    "did": "string",
    "key": "string",
    "msgId": "string",
    "authToken": "b843ef27-1ac6-49b8-ab71-cd0c22f4e50e",
    "correlationId": "e721639b-c095-40b3-86e2-acecb2cb6efb",
    "userInfo": {
      "id": 23299,
      "uuid": "e721639b-c095-40b3-86e2-acecb2cb6efb",
      "userName": "9337682030",
      "name": "Abhilash Seth",
      "type": "EMPLOYEE",
      "mobileNumber": "9337682030",
      "emailId": "abhilash.seth@gmail.com",
      "roles": [
        {
          "id": 281,
          "name": "Employee"
        }
      ]
    }
  },
  "apiDetails": {

```

```
"uri":"http://property-services:8080/property-  
services/assessment/_plainsearch",  
"paginationDetails":{  
  "offsetKey":"offset",  
  "sizeKey":"limit",  
  "maxPageSize":50,  
  "startingOffset": 0  
},  
"responseJsonPath":"$.Assessments"  
},  
"legacyIndexTopic":"pt-assessment-legacyIndex",  
"tenantId":"pb.amritsar"  
}'
```