

## View Property Details, Transfer Ownership and View History



### View Property Details

#### Objective

View Property Details page created to display the details of the successfully created Property record. Properties created can be traced by their unique Property ID number. Property details page displays data entered while creating property, operations performed on the file, assessment information, payment dues, workflow information etc.

It also has download and print options where users can download or print the Application details in PDF format.

**Property Information** **Property ID: PB-PT-2020-04-24-004869**



Property Address		
City	Door/House No.	Building/Company Name
Amritsar	NA	NA
Mohalla	Pincode	Existing Property ID
Ajit Nagar - Area1	NA	NA

#### Technical Implementation Details

View Property Details page uses a React Class component written in egov-ui-kit module. Below is the file link.

<https://github.com/egovernments/frontend/blob/master/web/rainmaker/dev-packages/egov-ui-kit-dev/src/common/propertyTax/Property/index.js>

Employee application and citizen application both refers to this component to display property details information. User navigates to this page by providing PropertyID and TenantID in the Navigation url as parameters.

path: "property-tax/property/:propertyId/:tenantId"

When the user landed to this page, PropertyID and TenantID extracted from query params and supplied them to the *fetchProperties* function to get the property data.

```
77     fetchProperties([
78       { key: "propertyIds", value: decodeURIComponent(this.props
79       { key: "tenantId", value: this.props.match.params.tenantId
```

Property component is a class based component and connected to redux. *fetchProperties* method is a redux action method. It sends the property details data to the reducer and reducer sets the data in states and returns the state. States are mapped to the props so the property details data captured in the props of the component.

```
609   const mapStateToProps = (state, ownProps) => {
610     const { app, common } = state;
611     const { urls, localizationLabels } = app;
612     const { cities } = common;
613     const { generalMDMSDataById } = state.common || {};
614     let { propertiesById, singleAssessmentByStatus = [], loading, receiptsByYr, totalBillAmo
```

Property details data received through captured in the props will be formatted into various formats via utility functions. After formatting these data, it will be supplied to the rendered component.

Property component renders two main components inside its render block.

<PTHeader />

<AssessmentList />

PTHeader: To display header data of the Property details page. Below is the file link.

<https://github.com/egovernments/frontend/blob/master/web/rainmaker/dev-packages/egov-ui-kit-dev/src/common/common/PTHeader/index.js>

AssessmentList: To display the details of the property. Below is the file link.

<https://github.com/egovernments/frontend/blob/master/web/rainmaker/dev-packages/egov-ui-kit-dev/src/common/propertyTax/AssessmentList/index.js>

It also renders footer buttons, EDIT Property and ASSESS Property. These footer buttons are configurable.

```
320         <Button
321           label={
322             <Label buttonLabel={true}
323               label={formWizardConstants[PROPERTY_FORM_PURPOSE.UPDATE].parentButton} fi
324               color="#fe7a51" />
325           }
326           onClick={() => this.onEditPropertyClick()}
327           labelStyle={{ letterSpacing: 0.7, padding: 0, color: "#fe7a51" }}
328           buttonStyle={{ border: "1px solid #fe7a51" }}
329           style={{ lineHeight: "auto", minWidth: "45%", marginRight: "10%" }}
330         />
---
```

## API Call Details

After fetching properties, few more APIs are called. Below are the details of such API calls.

**fetchAssessments:** This API returns the total no of assessment performed on the Property and the details about each assessment.

```
87     fetchAssessments([
88       { key: "propertyIds", value: decodeURIComponent(this.props.
89       { key: "tenantId", value: this.props.match.params.tenantId
```

**fetchReceipt:** This API returns the receipt details related to the Property so that user can download or print it.

```
254     this.props.fetchTotalBillAmount([
255       { key: "consumerCode", value: propertyId },
256       { key: "tenantId", value: this.props.match.params.tenantId },
257       { key: "businessService", value: 'BT' }
```

**fetchTotalBillAmount:** This API returns the data related to the total Bill amount to be paid by the user. It gives the complete breakup bill details that we are showing in the Estimate card component.

## MDMS Call Details

We have one MDMS call that returns the dropdown details used while creating the Property Tax file. This Dropdown details we are mapping to the dropdown keys present in Property data so that we display the meaningful information on the details page.

```
74     const requestBody = generalMDMSDataRequestObj(commonConfig.tena
75     fetchGeneralMDMSData(requestBody, "PropertyTax", getGeneralMDMSData)
```

## Localization Implementation

This page uses the locales from the following locale module.

- Rainmaker-common
- Rainmaker-pt
- Rainmaker-pgr
- TenantID specific locale

While navigating to the details page; at the route authentication phase all the required locale calls are made and the data stored in localStorage.

While initializing the details page; locale keys from the localStorage are stored in the local variable in the component.

```
47     const locale = getLocale() || "en_IN";
48     const localeKeys = getLocaleKeys(locale);
```

While formatting property data for each section, we are mapping these locale keys to their respective data.

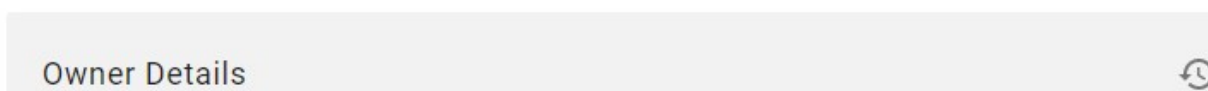
```
370     const getAddressInfo = (addressObj, extraItems) => {
371         return (
372             addressObj && [
373                 {
374                     heading: getTranslatedLabel("PT_PROPERTY_ADDRESS_SUB_HEADER", localizati
375                     // iconAction: "action",
376                     iconName: "home",
377                     items: [
378                         {
379                             key: getTranslatedLabel("PT_PROPERTY_ADDRESS_CITY", localizationLabe
380                             value: addressObj.city || "NA",
381                         },
382                         {
383                             key: getTranslatedLabel("PT_PROPERTY_ADDRESS_HOUSE_NO", localizationLabe
```

getTranslatedLabel: This method is responsible for localization of labels. It accepts the label keys and localization keys as parameters and returns the translated label. This function is written at framework level and is being used across application.

## Transfer Ownership

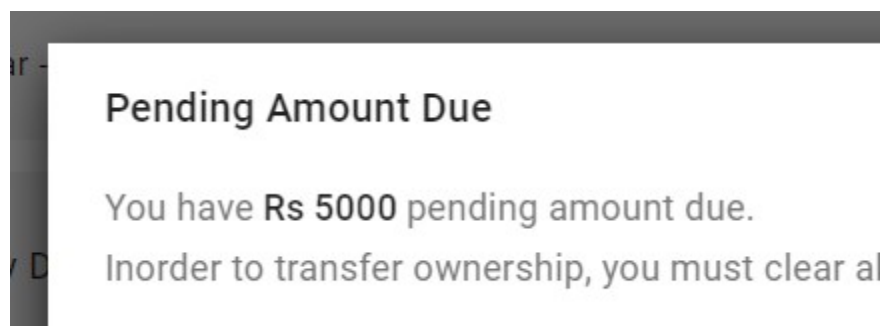
### Objective

Transfer ownership operation is responsible for transferring the property to another person. Property mutation flow starts from transfer ownership operation.



This operation checks below two conditions.

- If the Total Due amount is more than zero, then a notification popup opens up stating that the total dues amount with PROCEED TO PAY option. Once user clicks on the PROCEED TO PAY button it navigates to the common pay screen where user makes the payment.



- If the Total Dues amount is zero then after clicking on TRANSFER OWNERSHIP button then user navigates to Mutation screens.

### Technical Implementation Details

TRANSFER OWNERSHIP is a component which is rendered by OwnerInfo card component.

```
314      <TransferOwnership ownershipTransfer={ownershipTransfer} openDialog={this.openDialog} />
```

Pending Amount Due popup is also a component rendered by OwnerInfo card component.

```
347      {this.state.pendingAmountDue && (  
348        <PendingAmountDialog  
349          open={this.state.pendingAmountDue}  
350          amount={totalBillAmountDue}  
351          tenantId={properties.tenantId}  
352          consumerCode={properties.propertyId}
```

OwnerInfo card component file path:

<https://github.com/egovernments/frontend/tree/master/web/rainmaker/dev-packages/egov-ui-kit-dev/src/common/propertyTax/Property/components/OwnerInfo> - Connect to preview

Transfer Ownership button component file path:

<https://github.com/egovernments/frontend/tree/master/web/rainmaker/dev-packages/egov-ui-kit-dev/src/common/propertyTax/Property/components/ActionItems> - Connect to preview

Pending Amount Due dialog component file path:

<https://github.com/egovernments/frontend/tree/master/web/rainmaker/dev-packages/egov-ui-kit-dev/src/common/propertyTax/Property/components/PendingAmountDue> - Connect to preview

## API Call details

No API call made.

## MDMS Call Details

No MDMS call made.

## Localization Implementation

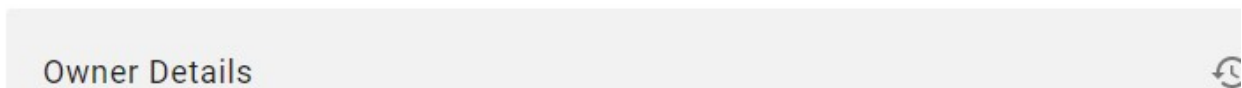
This component uses the locale keys available in the rainmaker-pt locale module. For example:

```
{
  "code": "PT_PENDING_AMOUNT_DUE",
  "message": "Pending Amount Due",
  "module": "rainmaker-pt",
  "locale": "en_IN"
},
```

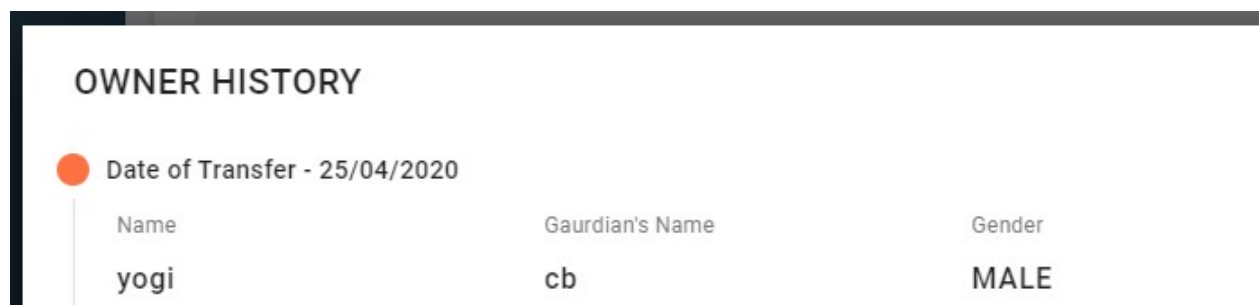
## View History

### Objective

View history operation shows the details about the property being transferred to the owners. When property is created it adds the owner information from the create property details. When the same property is transferred to the different owners by doing TRANSFER OWNERSHIP operation, It adds the owner details to whom the property was transferred.



View History is a popup which opens when the user clicks on the VIEW HISTORY button. It shows Ownership History of the property.



### Technical Implementation Details

VIEW HISTORY is a component which is rendered by OwenerInfo card component.

```
313 <ViewHistory viewHistory={viewHistory} openDialog={this.openDialog} />
```

OWNER HISTORY is a popup screen which is rendered by OwnerInfo card component.

```

357         {this.state.viewHistory && (
358             <ViewHistoryDialog
359                 open={this.state.viewHistory}
360                 ownershipInfo={this.state.ownershipInfo}
361                 closeDialogue={() => this.closeDialogue("viewHistory"

```

OwnerInfo card component file path:

<https://github.com/egovernments/frontend/tree/master/web/rainmaker/dev-packages/egov-ui-kit-dev/src/common/propertyTax/Property/components/OwnerInfo - Connect to preview>

View History button component file path:

<https://github.com/egovernments/frontend/tree/master/web/rainmaker/dev-packages/egov-ui-kit-dev/src/common/propertyTax/Property/components/ActionItems - Connect to preview>

Owner History Dialog component file path:

<https://github.com/egovernments/frontend/tree/master/web/rainmaker/dev-packages/egov-ui-kit-dev/src/common/propertyTax/Property/components/ViewHistory - Connect to preview>

## Localization Implementation

It uses rainmaker-pt locale module to show translated labels. For example:

```

{
  "code": "PT_OWNER_HISTORY",
  "message": "OWNER HISTORY",
  "module": "rainmaker-pt",
  "locale": "en_IN"
},

```

## API Call Role Action mapping

S.No.	API	Action id	Roles
1	/egov-mdms-service/v1/_search	954	LOA_CREATOR,SUPERUSER,WO_CREATOR,AE_CREATOR,WORKS_MASTER_CREATOR,



2	/user/_search	604	SUPERUSER,SYS_INTEGRATOR_FINANCE,SYS_INTEGRATOR_WATER_SEW,EGF_BILL_CREATOR,EMPLOYEE,EMPLOYEE_FINANCE,
3	/localization/messages/v1/_search	1531	SUPERUSER,EMPLOYEE,CITIZEN,GRO,DGRO,
4	/property-services/property/_search	1897	CITIZEN, PT_CEMP,PT_DOC_VERIFIER,PT_FIELD_INSPECTOR,PT_APPROVER
5	/property-services/assessment/_search	1945	SUPERUSER,PT_CEMP,CITIZEN
6	/collection-services/payments/PT/_search	2029	PTCEMP, CITIZEN
7	/billing-service/bill/v2/_fetchbill	1862	EMPLOYEE,CITIZEN
8	/filestore/v1/files/url	1528	SUPERUSER,EMPLOYEE,CITIZEN,GRO,DGRO,
9	/access/v1/actions/mdms/_get	870	CSR, GRO,DGRO,EMPLOYEE,CITIZEN,PGR-ADMIN,EMPLOYEE_FINANCE,TL_APPROVER,TL_CEMP,SUPERUSER,COLL_OPERATOR,BANK_COLL_OPERATOR,CSC_COLL_OPERATOR,DMA OFFICE OFFICER,ULB OFFICER,COMMISSIONER,