# Data Structures Documentation

This document provides comprehensive information about all data structures used in the application.

## Table of Contents

1. [Analysis Data Structure](#analysis-data-structure)
2. [Properties Data Structure](#properties-data-structure)
3. [Categories Data Structure](#categories-data-structure)
4. [Users Data Structure](#users-data-structure)
5. [Transactions Data Structure](#transactions-data-structure)
6. [Partner Contributions Data Structure](#partner-contributions-data-structure)
7. [Loans Data Structure](#loans-data-structure)

## Analysis Data Structure

The analysis data structure represents various types of property investment analyses.

### Core Fields

* id: UUID string
* user\_id: string
* created\_at: ISO 8601 datetime string
* updated\_at: ISO 8601 datetime string
* analysis\_type: string
* analysis\_name: string

### Property Details

* address: string
* square\_footage: integer
* lot\_size: integer
* year\_built: integer
* bedrooms: integer
* bathrooms: float

### Financial Details

* purchase\_price: integer
* after\_repair\_value: integer
* renovation\_costs: integer
* renovation\_duration: integer
* furnishing\_costs: integer // One-time expense for PadSplit furnishing
* cash\_to\_seller: integer
* closing\_costs: integer
* assignment\_fee: integer
* marketing\_costs: integer

### Income Details

* monthly\_rent: integer

### Expense Details

* property\_taxes: integer
* insurance: integer
* hoa\_coa\_coop: integer
* management\_fee\_percentage: float
* capex\_percentage: float
* vacancy\_percentage: float
* repairs\_percentage: float
* utilities: integer
* internet: integer
* cleaning: integer
* pest\_control: integer
* landscaping: integer
* padsplit\_platform\_percentage: float

### Loan Details

#### Initial Loan

* initial\_loan\_name: string
* initial\_loan\_amount: integer
* initial\_loan\_interest\_rate: float
* initial\_loan\_term: integer
* initial\_loan\_down\_payment: integer
* initial\_loan\_closing\_costs: integer

#### Refinance Loan

* refinance\_loan\_name: string
* refinance\_loan\_amount: integer
* refinance\_loan\_interest\_rate: float
* refinance\_loan\_term: integer
* refinance\_loan\_down\_payment: integer
* refinance\_loan\_closing\_costs: integer

#### Additional Loans (1-3)

* loan1\_loan\_name: string
* loan1\_loan\_amount: integer
* loan1\_loan\_interest\_rate: float
* loan1\_loan\_term: integer
* loan1\_loan\_down\_payment: integer
* loan1\_loan\_closing\_costs: integer
* loan2\_loan\_name: string
* loan2\_loan\_amount: integer
* loan2\_loan\_interest\_rate: float
* loan2\_loan\_term: integer
* loan2\_loan\_down\_payment: integer
* loan2\_loan\_closing\_costs: integer
* loan3\_loan\_name: string
* loan3\_loan\_amount: integer
* loan3\_loan\_interest\_rate: float
* loan3\_loan\_term: integer
* loan3\_loan\_down\_payment: integer
* loan3\_loan\_closing\_costs: integer

### Comps Integration Data

* comps\_data: object (null if comps haven't been run)
  + last\_run: ISO 8601 datetime string
  + run\_count: integer (number of times comps have been run in current session)
  + estimated\_value: integer
  + value\_range\_low: integer
  + value\_range\_high: integer
  + comparables: array of objects
    - id: string
    - formattedAddress: string
    - city: string
    - state: string
    - zipCode: string
    - propertyType: string
    - bedrooms: integer
    - bathrooms: float
    - squareFootage: integer
    - yearBuilt: integer
    - price: integer
    - listingType: string
    - listedDate: ISO 8601 datetime string
    - removedDate: ISO 8601 datetime string (null if still active)
    - daysOnMarket: integer
    - distance: float
    - correlation: float

### Example Comps Data

{

"comps\_data": {

"last\_run": "2024-09-28T13:21:51.018Z",

"run\_count": 1,

"estimated\_value": 221000,

"value\_range\_low": 208000,

"value\_range\_high": 233000,

"comparables": [

{

"id": "unique-comp-id",

"formattedAddress": "123 Main St",

"city": "San Antonio",

"state": "TX",

"zipCode": "78244",

"propertyType": "Single Family",

"bedrooms": 4,

"bathrooms": 2,

"squareFootage": 1747,

"yearBuilt": 1986,

"price": 229900,

"listingType": "Standard",

"listedDate": "2024-04-03T00:00:00.000Z",

"removedDate": "2024-05-26T00:00:00.000Z",

"daysOnMarket": 53,

"distance": 0.2994,

"correlation": 0.9822

}

]

}

}

### Background Notes

1. Store required data in JSON as raw data (str, int, bool, float). Calculated and derived values are created and calculated at run-time.
2. Currency and percentage values are only formatted for the Reports Tab output and reports downloaded as PDF.
3. Currency format follows "$X,XXX.XX" pattern
4. Percentage format follows "XX.XX%" pattern
5. Date-times are in ISO 8601 format
6. Fields made available to the user depend upon the analysis type selected. Assume null values unless specified otherwise by the user.
7. Currency and percentage formats are performed at the frontend only for the Reports Tab and when generating reports to download as PDF.

## Properties Data Structure

This schema defines the structure for real estate property data, including purchase details, financing, income, expenses, and partnership information.

### Root Object Properties

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| address | string | Full property address including street, city, state, zip, and country |
| purchase\_price | number | Total purchase price in USD |
| purchase\_date | string | Date of purchase (YYYY-MM-DD format) |
| down\_payment | number | Down payment amount in USD |
| primary\_loan\_rate | number | Primary loan interest rate as percentage |
| primary\_loan\_term | number | Primary loan term in months |
| primary\_loan\_amount | number | Primary loan amount in USD |
| primary\_loan\_start\_date | string | Primary loan start date (YYYY-MM-DD format) |
| secondary\_loan\_amount | number | Secondary/seller financing amount in USD |
| secondary\_loan\_rate | number | Secondary loan interest rate as percentage |
| secondary\_loan\_term | number | Secondary loan term in months |
| closing\_costs | number | Total closing costs in USD |
| renovation\_costs | number | Total renovation costs in USD |
| marketing\_costs | number | Total marketing costs in USD |
| holding\_costs | number | Total holding costs in USD |

### Monthly Income Object

"monthly\_income": {

"rental\_income": number,

"parking\_income": number,

"laundry\_income": number,

"other\_income": number,

"income\_notes": string

}

### Monthly Expenses Object

"monthly\_expenses": {

"property\_tax": number,

"insurance": number,

"repairs": number,

"capex": number,

"property\_management": number,

"hoa\_fees": number,

"utilities": {

"water": number,

"electricity": number,

"gas": number,

"trash": number

},

"other\_expenses": number,

"expense\_notes": string

}

### Partners Array

"partners": [

{

"name": string,

"equity\_share": number,

"is\_property\_manager": boolean

}

]

### Data Types

* All numerical values should include decimal points (e.g., 100.0 instead of 100)
* Dates should be in ISO 8601 format: YYYY-MM-DD
* Percentages are stored as decimal numbers (e.g., 5.75 for 5.75%)
* Boolean values should be true or false
* Strings should be enclosed in double quotes

## Categories Data Structure

The categories data structure defines the available categories for income and expense transactions.

{

"income": [

"Application Fees",

"Common Area Revenue",

"Escrow Refund",

"Insurance Refund",

"Late Fees",

"Loan Repayment",

"Other Income",

"Parking/Storage Fees",

"Pet Rent",

"Rent",

"Security Deposit",

"Utility Reimbursement"

],

"expense": [

"Asset Acquisition",

"Association Dues",

"Bank/Financial Fees",

"Capital Expenditures",

"Cleaning",

"Furnishing",

"Insurance",

"Landscaping",

"Legal/Professional Fees",

"Maintenance",

"Marketing/Advertising",

"Mortgage",

"Other Expense",

"Permit/License Fees",

"Pest Control",

"Property Management Fees",

"Property Tax",

"Repairs",

"Security",

"Supplies",

"Utilities"

]

}

## Users Data Structure

The users data structure stores user information, credentials, and roles.

### User Object Structure

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| first\_name | string | User's first name |
| last\_name | string | User's last name |
| name | string | User's full name |
| email | string | User's email address (used as key in the JSON object) |
| phone | string | User's phone number with country code |
| password | string | Hashed password using PBKDF2 with SHA-256 |
| role | string | User's role (Admin or User) |
| property\_access | array | List of properties the user has access to with access levels and equity shares |

### Property Access Object Structure

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| property\_id | string | Property identifier (typically the full address) |
| access\_level | string | Access level ("owner", "manager", "editor", "viewer") |
| equity\_share | number | User's equity share percentage in the property (optional) |

### Example User Object

"bjmar867@gmail.com": {

"first\_name": "BJ",

"last\_name": "Marshall",

"name": "BJ Marshall",

"email": "bjmar867@gmail.com",

"phone": "+14435465716",

"password": "pbkdf2:sha256:600000$RW1h6Oalhp4zutww$bfea68cc108126cb1673d76bd8b86c3551d3d8533e1b66412218ce6b1b79d2af",

"role": "Admin",

"property\_access": [

{

"property\_id": "1911 Grinnalds Avenue, Baltimore, MD 21223, United States of America, Baltimore, Maryland, 21223",

"access\_level": "owner",

"equity\_share": 100.0

},

{

"property\_id": "454 Guilford Avenue, Hagerstown, MD 21740, United States of America, Hagerstown, Maryland, 21740",

"access\_level": "manager",

"equity\_share": 0.0

}

]

}

### Access Level Hierarchy

* **owner**: Full access to property, including financial management and partner equity
* **manager**: Can manage property transactions and maintenance, but cannot modify partner equity
* **editor**: Can edit property details and add transactions, but cannot manage reimbursements
* **viewer**: Read-only access to property information

## Transactions Data Structure

The transactions data structure records all financial transactions related to properties.

### Transaction Object Structure

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| property\_id | string | Property identifier (typically the full address) |
| type | string | Transaction type ("income" or "expense") |
| category | string | Transaction category from categories.json |
| description | string | Description of the transaction |
| amount | number | Transaction amount |
| date | string | Transaction date (YYYY-MM-DD format) |
| collector\_payer | string | Name of the person who collected or paid |
| documentation\_file | string | Filename of the supporting documentation |
| reimbursement | object | Information about reimbursement status |
| id | string | Unique transaction identifier |

### Reimbursement Object

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| date\_shared | string | Date when reimbursement was shared (YYYY-MM-DD format) |
| share\_description | string | Description of the sharing status |
| reimbursement\_status | string | Status of the reimbursement (e.g., "completed") |

### Example Transaction Object

{

"property\_id": "1911 Grinnalds Avenue, Baltimore, MD 21223, United States of America, Baltimore, Maryland, 21223",

"type": "expense",

"category": "Mortgage",

"description": "2024.05.01 Shellpoint Mortgage",

"amount": 603.32,

"date": "2024-05-01",

"collector\_payer": "BJ Marshall",

"documentation\_file": "trans\_79\_2024\_Grinnalds\_Shellpoint\_Mortgage\_Payments.png",

"reimbursement": {

"date\_shared": "2024-05-01",

"share\_description": "Auto-completed - Single owner property",

"reimbursement\_status": "completed"

},

"id": "79"

}

### Notes

* The documentation\_file references a file stored in the /data/uploads/ directory
* Transaction IDs are assigned sequentially as strings
* The property\_id uses the full address as an identifier

## Partner Contributions Data Structure

The partner contributions data structure tracks financial contributions and distributions between partners for properties.

### Partner Contribution Object Structure

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| property\_id | string | Property identifier (typically the full address) |
| partner\_name | string | Name of the partner making the contribution or receiving the distribution |
| amount | number | Amount of the contribution or distribution |
| contribution\_type | string | Type of transaction ("contribution" or "distribution") |
| date | string | Date of the contribution or distribution (YYYY-MM-DD format) |
| notes | string | Optional notes about the contribution or distribution |

### Example Partner Contribution Object

{

"property\_id": "454 Guilford Avenue, Hagerstown, MD 21740, United States of America, Hagerstown, Maryland, 21740",

"partner\_name": "BJ Marshall",

"amount": 5000.00,

"contribution\_type": "contribution",

"date": "2024-05-15",

"notes": "Initial capital contribution for renovation"

}

### Notes

* Contributions represent money invested into a property by a partner
* Distributions represent money taken out of a property by a partner
* The contribution history helps track the changing equity positions of partners over time
* This data structure complements the partner information in the properties.json file

## Loans Data Structure

The loans data structure provides comprehensive tracking of loans associated with properties, including detailed information about loan terms, status, and payment details.

### Loan Object Structure

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| id | string | Unique identifier for the loan (UUID) |
| property\_id | string | Property identifier the loan is associated with |
| loan\_type | string | Type of loan (initial, refinance, additional, heloc, seller\_financing, private) |
| amount | string | Loan amount with decimal precision |
| interest\_rate | string | Annual interest rate as a percentage |
| term\_months | number | Loan term in months |
| start\_date | string | Date when the loan started (YYYY-MM-DD format) |
| is\_interest\_only | boolean | Whether the loan is interest-only |
| balloon\_payment | object | Optional balloon payment details |
| lender | string | Name of the lender |
| loan\_number | string | Loan identification number from the lender |
| status | string | Current status of the loan (active, paid\_off, refinanced, defaulted) |
| refinanced\_from\_id | string | ID of the loan this refinanced (if applicable) |
| notes | string | Additional notes about the loan |
| name | string | Optional name for the loan |
| monthly\_payment | string | Monthly payment amount |
| current\_balance | string | Current loan balance |
| last\_updated | string | Date when the loan was last updated (YYYY-MM-DD format) |
| created\_at | string | ISO 8601 datetime when the loan was created |
| updated\_at | string | ISO 8601 datetime when the loan was last updated |

### Balloon Payment Object

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| due\_date | string | Date when the balloon payment is due (YYYY-MM-DD format) |
| amount | string | Amount of the balloon payment |
| term\_months | number | Number of months until the balloon payment is due |

### Example Loan Object

{

"id": "550e8400-e29b-41d4-a716-446655440000",

"property\_id": "454 Guilford Avenue, Hagerstown, MD 21740, United States of America, Hagerstown, Maryland, 21740",

"loan\_type": "initial",

"amount": "112800.00",

"interest\_rate": "5.73%",

"term\_months": 360,

"start\_date": "2022-08-03",

"is\_interest\_only": false,

"lender": "PennyMac",

"loan\_number": "12345678",

"status": "active",

"notes": "Primary mortgage for property purchase",

"name": "Primary Mortgage",

"monthly\_payment": "658.23",

"current\_balance": "109876.54",

"last\_updated": "2024-04-01",

"created\_at": "2022-08-03T12:00:00.000Z",

"updated\_at": "2024-04-01T15:30:00.000Z"

}

### Notes

* The loan data structure supports multiple loans per property
* Loan amounts and interest rates are stored with precise decimal handling
* The structure supports tracking loan status changes over time
* Refinanced loans maintain a reference to the original loan
* The system can generate amortization schedules and calculate remaining balances