

Tabs Documentation v1.0

Contributors: Klein Harrigan, Young Chen, Gigi Cheang, John Ma, Tyler Cong, Crystal Huang

Package Frontend

Class Name	DescBoxData
Description	Data for a box data element

Class Name	DescBoxList
Description	List of desc box data to be passed as an argument for a React component

Class Name	FolderItemData
Description	Data for a folder item to be passed into a React component

Class Name	InventoryCategory
Description	Data for an inventory category

Class Name	DescBox
Description	A rounded box with a gradient to display a description
Arguments	None
Return	DescBox React element
Throws	None

Class Name	FolderSvg
Description	Folder-like container to be shown to the user

Arguments	data: The data to be shown in the container
Return	FolderSvg React element
Throws	None

Class Name	FloatingActionButton
Description	Action button with styling to look like it's floating
Arguments	myFunction: Callback when the button is pressed
Return	FloatingActionButton React element
Throws	None

Class Name	FolderBackdropTextInputField
Description	Input text field styled to fit the theme of a folder backdrop
Arguments	title: Title of text field hint: Hint text for the text field
Return	FolderBackdropTextInputField React element
Throws	None

Class Name	FolderBackdropListFragment
Description	Folder backdrop element for the folder-like background
Arguments	None
Return	FolderBackdropListFragment React element
Throws	None

Class Name	FolderBackdropActionButton
Description	Button styled to fit the theme of a folder backdrop
Arguments	title: Text to display on the button

Return	FolderBackdropActionButton React element
Throws	None

Class Name	RentScreen
Description	Creates the screen to display rent information
Arguments	None
Return	RentScreen React element
Throws	None

Package Backend

Class Name	InventoryDocument (extends Document, is an Interface)
Description	Represents the format of inventory objects in the database
Fields	houseId: String

Class Name	CategoryDocument (extends Document, is an Interface)
Description	Represents the format of category objects in the database
Fields	inventoryId: String categoryId: String categoryName: String categoryDesc: String

Class Name	CategoryDocument
Description	Represents the format of category objects in the database
Fields	inventoryId: String categoryId: String categoryName: String categoryDesc: String

Class Name	ItemDocument (extends Document, is an Interface)
Description	Represents the format of item objects in the database
Fields	name: String categoryId: String quantity: String expiration: String tags: String[]

Class Name	BillDocument (extends Document, is an Interface)
Description	Represents the format of Bill object in the database

Fields	billId: String, houseId: String, name: String, amount: Number, split: [String], dateCreated: String, dateDue: String, status: String
---------------	---

Function	getBill
Description	Get bill corresponding to house Id
Arguments	houseId: String — the id of the house
Return	Bill — a bill object matching the query
Throws	None

Function	addBill
Description	Adds a bill object
Arguments	billId: String, houseId: String, name: String, amount: Number, split: [String], dateCreated: String, dateDue: String, status: String
Return	Bill — the bill object added
Throws	None

Class Name	DebtDocument (extends Document, is an Interface)
Description	Represents the format of Debt object in the database
Fields	debtId: String, debtTo: String, debtFrom: String, amount: Number, description: String, dateCreated: String,

Function	getDebtsTo
Description	Get debts owed to the user corresponding to the id of the user
Arguments	debtTo: String — the id of the user
Return	Debt[] — a list of debt objects matching the query
Throws	None

Function	getDebtsFrom
Description	Get debts owed by the user corresponding to the id of the user
Arguments	debtFrom: String — the id of the user
Return	Debt[] — a list of debt objects matching the query
Throws	None

Function	addDebt
Description	Adds a debt object
Arguments	debtId: String debtTo: String debtFrom: String amount: Float description: String dateCreated: String
Return	Debt — the debt object added
Throws	None

Function	findInventory
Description	Gets the id of the inventories of a given house
Arguments	houseId: String
Return	Inventory[] — a list of inventory objects matching the query
Throws	None

Function	createInventory
Description	Creates an inventory for a given house
Arguments	houseId: String
Return	String — the id of the created inventory
Throws	None

Function	findCatsByInvId
Description	Gets a list of categories that belong to a given inventory
Arguments	inventoryId: String — the inventory id to match
Return	Category[] — a list of category objects matching the query
Throws	None

Function	addCat
Description	Adds a category to a given inventory
Arguments	inventoryId: String categoryName: String categoryDesc: String
Return	String — the id of the created category
Throws	None

Function	changeCatName
Description	Changes a given category's name
Arguments	categoryId: String categoryName: String
Return	Boolean — true if operation was successful, false otherwise
Throws	None

Function	changeCatDesc
Description	Changes a given category's description
Arguments	categoryId: String categoryDesc: String
Return	Boolean — true if operation was successful, false otherwise
Throws	None

Function	findItem
Description	Gets the item object given the item id
Arguments	itemId: String
Return	Item — the item object matching the item id
Throws	None

Function	findItemsByCategories
Description	Gets a list of item object given a category id

Arguments	categoryId: String
Return	Item[] — an array of item objects matching the item id
Throws	None

Function	createItem
Description	Adds an item to a given category
Arguments	categoryId: String name: String expiration: String
Return	String — the id of the created item
Throws	None

Function	addItem
Description	Increments the given item's quantity by 1
Arguments	itemId:String
Return	String — "Success" if operation was successful, "Failure" otherwise
Throws	None

Function	subtractItem
Description	Decrements the given item's quantity by 1
Arguments	itemId:String
Return	String — "Success" if operation was successful, "Failure" otherwise
Throws	None

Function	modifyItemName
Description	Modifies the name of a given item
Arguments	itemId: String name: String
Return	String — name if operation was successful, empty string otherwise

Throws	None
---------------	------

Function	modifyItemCategory
Description	Modifies the category of a given item
Arguments	itemId:String categoryId:String
Return	String — category if operation was successful, empty string otherwise
Throws	None