

Meteor Project Docs

#buyMore()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/inventory/inventory.js>, line 22

clicking the button in the inventory box adds it to the shopping list

ArchivedReservation

new ArchivedReservation(Hour, Count)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/data/reservation.js>, line 9

Represents an archived reservation

Arguments

Hour Date

hour of reservation

Count Number

number of reservations at that hour

AvgCookTime

new AvgCookTime(time, menuItemID, cookTime, avgCookTime)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/data/avgCookTime.js>, line 7

Represents the data on cook time of an item

Arguments

time Date

time at which the data was created

menuItemID Number

id of the menu item

cookTime Number

time it took to cook the item in seconds

avgCookTime Number

average time it takes to cook the item in seconds

Bill

new Bill(billTimeCreated, billTable, billPaid)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/billsJS.js>, line 32

Represents a bill

Arguments

billTimeCreated Date

time the bill was made

billTable Number

The table ID of the bill

billPaid Boolean

If bill was paid or not

DataItems

new DataItems()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/stat.js>, line 58

Contains the statistics to be shown

Date

Date()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 342

Gets the date of the current bill and returns it in a proper format

FlowRouter

FlowRouter.route(/, /floorplan, /Reserve, /Success, /orderqueue, /manager, /waiter)

<https://github.com/Mitbits/Software-Project/tree/master/app/lib/routes.js>, line 6

A simple router for Meteor and routes the pages based on the link entered in the BlazeLayout render.

Arguments

/

BlazeLayout.render("loginPage")

/floorplan

BlazeLayout.render("floorplan")

/Reserve

BlazeLayout.render("reservationPage");

```
/Success
BlazeLayout.render("reservationSuccess");

/orderqueue
BlazeLayout.render("orderQueue");

/manager
BlazeLayout.render("menuSuggestions");

/waiter
BlazeLayout.render("waiter");
```

FlowRouter.route(/, /floorplan, /Reserve, /Success, /orderqueue, /manager, /waiter)

<https://github.com/Mitbits/Software-Project/tree/master/app/lib/routes.js>, line 6

A simple router for Meteor and routes the pages based on the link entered in the BlazeLayout render.

Arguments

```
/
BlazeLayout.render("loginPage")

/floorplan
BlazeLayout.render("floorplan")

/Reserve
BlazeLayout.render("reservationPage");

/Success
BlazeLayout.render("reservationSuccess");

/orderqueue
BlazeLayout.render("orderQueue");

/manager
BlazeLayout.render("menuSuggestions");

/waiter
BlazeLayout.render("waiter");
```

IngUsage

new IngUsage(time, ingID, quantity, avgQuantity, threshold, units)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/data/avgIngUsage.js>, line 7

Data about the ingredients left in the inventory

Arguments

```
time Date
time at which the data point was created
```

ingID Number

identifier of the ingredient

quantity Number

amount of ingredient left in the inventory

avgQuantity Number

average amount of ingredient in the inventory

threshold Number

amount of ingredient which tells us if the ingredient is low or not

units String

units of measurement of the ingredient

MenuItem

new MenuItem(itemID, itemName, itemDescription, mealType, itemPrice, cookTime, ingredients, timesOrdered, itemPopularity)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/menuitem.js>, line 72

Class representing an item on a menu.

Arguments

itemID Number

Unique item identifier

itemName String

Name of the item

itemDescription String

Describes the item

mealType ORDER_TYPE

The type of item

itemPrice Number

The cost of the item

cookTime Number

Time to make the item in minutes

ingredients inredientsArray

The ingredients for a menu item

timesOrdered Number

The number of times the item was ordered in its lifetime

itemPopularity POPULARITY

The item's popularity level

new MenuItem(itemID, itemName, itemDescription, mealType, itemPrice, cookTime, ingredients, timesOrdered, itemPopularity)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/menuitem.js>, line 72

Class representing an item on a menu.

Arguments

itemID Number

Unique item identifier

itemName String

Name of the item

itemDescription String

Describes the item

mealType ORDER_TYPE

The type of item

itemPrice Number

The cost of the item

cookTime Number

Time to make the item in minutes

ingredients ingredientsArray

The ingredients for a menu item

timesOrdered Number

The number of times the item was ordered in its lifetime

itemPopularity POPULARITY

The item's popularity level

Meteor

Meteor.startup()

<https://github.com/Mitbits/Software-Project/tree/master/app/server/main.js>, line 15

The code in this file is run every time meteor starts up. The file involves the creation of the sample data needed for the project and pushing them to their respective collections

NumOrders

new NumOrders(time, numOrders, avgNumOrders)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/data/avgNumOrders.js>, line 7

Represents an data on number of orders in the order queue(completed)

Arguments

time Date

time the data point was measured/created

numOrders Number

number of orders completed by the chef

avgNumOrders Number

avg number of orders completed at this hour

ORDER_TYPE

ORDER_TYPE

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/menuItem.js>, line 19

Enum for order types.

Order

new Order(orderID, waiterID, orderItems, timePlaced, isCompleted)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/order.js>, line 62

Represents an Order

Arguments

orderID Number

Unique order identifier

waiterID Number

Waiter identifier for waiter that placed the order

orderItems Array of orderItems

Contains all items for order

timePlaced Date

Time and date the order was created

isCompleted Boolean

Indicates if an Order is completed or not

POPULARITY

POPULARITY

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/menuItem.js>, line 41

Enum for popularity levels

POPULARITY_PERIOD

POPULARITY_PERIOD

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/mealSuggestions.js>, line 17

Enum for popularity periods.

PopItem

new PopItem(menuItemID, itemName, mealType, numCooked)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/data/popularItems.js>, line 6

Represents the temporary (weekly) popular items

Arguments

menuItemID Number

ID of the menuItem

itemName String

name of the menuItem

mealType ORDER_TYPE

Order Type Enum of the menuItem

numCooked Number

number of completed orders of this item

PriorityManager

PriorityManager()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/priorityManager.js>, line 361

Starts the priority manager module

Reservation

new Reservation(firstName, lastName, phoneNum, email, seats, date, assigned)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/reservation.js>, line 9

Represents reservation records from the client-side reservation system

Arguments

firstName String

First name of reservation holder

lastName String

Last name of reservation holder

phoneNum Number

Phone Number of reservation holder

email String

Email address of reservation holder

seats Number

Number of seats reserved

date Date

Date and time of reservation

assigned Boolean

whether reservation has been assigned a table

Table

new Table(table_id, size, occupants, table_status, table_type, reservation, merged, checked_for_merge)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 59

Represents a table at the restaurant

Arguments

table_id Number

Unqiue table identifier

size Number

Size of the table in seats

occupants Number

Number of people seated at the table

table_status TableStatus

Current status of the table

table_type TableType

The type of table

reservation Reservation

Reservation for the table

merged Boolean

True if the table is a component of a merged table

checked_for_merge Boolean

True if a table is selected as a component for manual merge in the floor plan interface

TableManager

new TableManager(reservations-, reservation_intv)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 212

manages all tables of all sizes, merges and assigns tables

Arguments

reservations- Reservation

Queue of pending reservations

reservation_intv Number

threshold time difference between now and reservation date for when reservation should be assigned

TableStatus

TableStatus

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 22

Enum for the current status of a table

TableType

TableType

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 37

Enum that represents the type of a table

TimeObject Creator

TimeObject Creator

<https://github.com/Mitbits/Software-Project/tree/master/app/client/order/order.js>, line 54

Creates the time objects when the page is rendered.

TimeSpent

new TimeSpent(time, spent, avgSpent)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/data/avgTimeSpent.js>, line 7

Represents the time spent by a customer in the restaurant

Arguments

time Date

time when the data point was created

spent Number

time in seconds the customer spent in the restaurant

avgSpent Number

average time in seconds the customer spends "at this hour"

addOccupants

addOccupants(numOccupants)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 183

Adds occupants to a table

Arguments

numOccupants Number

number of occupants to be added to the table

addQuantity

addQuantity(diff)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/ingredient.js>, line 57

Adds to the quantity in the inventory

Arguments

diff

addTimeHandler

addTimeHandler(event, templateInstance)

<https://github.com/Mitbits/Software-Project/tree/master/app/client/order/order.js>, line 162

Adds a specified amount of time (30secs for now) to the current time of an Order

Arguments

event

templateInstance

apps

apps()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 279

returns all the objects from the database with the mealType of Enum 1, appetizers.

arcSave

arcSave()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/data/reservation.js>, line 31

saves ArchivedReservation to db

assignPopularity

assignPopularity()

<https://github.com/Mitbits/Software-Project/tree/master/app/server/main.js>, line 327

This function will assign popularity to menu items based time period ranks

assignRanks

assignRanks()

<https://github.com/Mitbits/Software-Project/tree/master/app/server/main.js>, line 307

This function will assign ranks to menu items for each duration of time

assignTable

assignTable(table, res)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 315

assigns a reservation to a table

Arguments

table Table

Table to assign reservation to

res Reservation

Reservation to assign a table

barcode

barcode()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 356

Returns the unique ID of each bill to display

belowThreshold

belowThreshold()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/inventory/inventory.js>, line 153

Checks if the item is below the threshold

Returns

boolean

billItem

new billItem(billItemName, billItemPrice)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/billsJS.js>, line 15

Each item in a bill

Arguments

billItemName String

Name for the item on the bill

billItemPrice Number

Price of the item in the bill

bindData

bindData()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/archive.js>, line 43

pulls billing data from the database and binds it to the HTML page

change_for_merge

change_for_merge()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 145

toggles the checked `formerge` attribute

clean

clean()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 154

clears a table up and makes it clean, unmerges

click

click #addItem()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 234

Creates the order item for the selected item and pushes the order back into the orderArray

click #cancelOrder()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 186

Drops all the selected items for the order

click #clearList()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/inventory/inventory.js>, line 78

Clears the shopping list

click #merge_button()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/floorPlan/floorplan.js>, line 12

transistions interface to manual merge interface, allows for selecting tables to merge

click #payButton()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 218

Marks the bill as paid and sets the respective table to dirty

click #placeOrder()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 152

Takes all the Order items created by orderArray and fills in the rest of the fields of the order and places the order into the order collection

click #removeItem()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter.js>, line 245

Removes the respective item from the orderArray so it is not placed in the final order

click .Appetizers()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter.js>, line 83

Displays the appetizers menu and hides the rest along with the sidebar

click .Desserts()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter.js>, line 122

Displays the desserts menu and hides the rest along with the sidebar

click .Entrees()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter.js>, line 96

Displays the entrees menu and hides the rest along with the sidebar

click .FloorPlan()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter.js>, line 109

Displays the floor plan and hides the rest along with the sidebar

click .drinkicon()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter.js>, line 194

Opens the drinks modal showcasing all the menu items classified as drinks

click .gray.right.corner.label()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/floorPlan/floorplan.js>, line 145

Updates table status to Taken from Reserved

click .green.right.corner.label()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/floorPlan/floorplan.js>, line 133

Updates table status to Taken from Clean & Updates number of Occupants on table

click .minus.icon.link()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/reservation/reservation.js>, line 31

Decrement count by 1 with minimum of 1

click .minus.icon.link()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/reservation/reservation.js>, line 31

Decrement count by 1 with minimum of 1

click .particularItem()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/manager/manager.js>, line 28

This function is used to extract menulitem details for the particular menulitem clicked on by the user

click .payBill()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 203

Displays the bill for the table and hides the rest along with the sidebar

click .placeOrder()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 138

Displays the place order tab and hides the rest along with the sidebar

click .plus.icon.link()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/reservation/reservation.js>, line 15

Increments count by 1 with maximum of 4

click .plus.icon.link()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/reservation/reservation.js>, line 15

Increments count by 1 with maximum of 4

click .red.right.corner.label()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/floorPlan/floorplan.js>, line 108

Updates table status to Clean from Dirty

click .remove.icon.link()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/inventory/inventory.js>, line 62

Removes the selected item from the shopping list

click .small.send.icon.link()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/inventory/inventory.js>, line 87

Sends the order, which automatically updates the inventory

click .tableID()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 70

Adds a red dashed border around the selected table and calls setTableID

click .ui.yellow.button()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/floorPlan/floorplan.js>, line 85

Updates table status to dirty from taken

click div

click div#tap()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/order/order.js>, line 15

Catches all the events that occur on the page

clickCookTimelitems

clickCookTimelitems()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/kitchen.js>, line 56

Handles the click events on different menu items select feature

clickIngredients

clickIngredients()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/kitchen.js>, line 65

Handles the click events on the different ingredients select feature

clickMenu

clickMenu()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/stat.js>, line 42

Handles the click events on the menu

cookTime

cookTime(order)

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 506

returns the cookTime for the given order

Arguments

order

create menu

create menu()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/stat.js>, line 10

Creates the menu based on the statistics to be shown (based on the Dataitems object)

createBillItem

createBillItem(mBillItemName, mBillItemPrice)

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 47

Creates an a bill item with the following parameters

Arguments

mBillItemName

mBillItemPrice

createChart

createChart()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/chart.js>, line 6

creates the actual HTML graphics object for the chart using the d3 timerseries api - used for two data series

createChartSingleY

createChartSingleY()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/chart.js>, line 22

creates the actual HTML graphics object for the chart using the d3 timerseries api - used for only one data series

createCookTimeDisplay

createCookTimeDisplay()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/kitchen.js>, line 76

Pulls appropriate data of cook times from database and creates the chart to be displayed

createDisplay

createDisplay()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/kitchen.js>, line 120

Called by the master controller in statjs

createIngUsageDisplay

createIngUsageDisplay()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/kitchen.js>, line 90

Pulls appropriate data of ingredient usage from database and creates the chart to be displayed

createNumOrdersDisplay

createNumOrdersDisplay()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/kitchen.js>, line 105

Pulls appropriate data of number of orders from database and creates the chart to be displayed

createNumOrdersGraph

createNumOrdersGraph()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/kitchen.js>, line 46

creates the number of orders graph

createOrderItem

createOrderItem(mItemID, mPriority, mMenuItemID, mSpecialRequests)

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 29

Creates an order item with the following parameters

Arguments

mItemID

mPriority

mMenuItemID

mSpecialRequests

createOrderItems

createOrderItems()

<https://github.com/Mitbits/Software-Project/tree/master/app/server/main.js>, line 139

Generates an array of random orderItems using the getRandomNumber and createOrderItem functions.

Returns

Array

createPieChart

createPieChart()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/chart.js>, line 36

creates the actual HTML graphics object for the chart using the c3 api - used for pie charts only

createPopItemsCharts

createPopItemsCharts()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/general.js>, line 15

helper to create popular items graphs

createPopItemsDisplay

createPopItemsDisplay()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/general.js>, line 38

Pulls appropriate data of popular items from database and creates charts to be displayed for each menu item type

createTimeSpentDisplay

createTimeSpentDisplay()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/general.js>, line 24

Pulls appropriate data of time spent from database and creates the chart to be displayed

createTimeSpentGraph

createTimeSpentGraph()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/general.js>, line 8

helper to create time spent graph

datePickerFtn

datePickerFtn()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/archive.js>, line 9

shows the datepicker for the transaction viewing date

decimalPrice

decimalPrice()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/inventory/inventory.js>, line 204

Converts the price to proper decimal places and returns the value

decimalPrice()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/inventory/inventory.js>, line 204

Converts the price to proper decimal places and returns the value

desserts

desserts()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 293

returns all the objects from the database with the mealType of Enum 3, desserts.

doneButtonHandler

doneButtonHandler(event, templateInstance)

<https://github.com/Mitbits/Software-Project/tree/master/app/client/order/order.js>, line 86

Allows user to get rid of an order from the order queue and use that data to update new existing values in database based on the algorithm

Arguments

event

templateInstance

drinks

drinks()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 272

returns all the objects from the database with the mealType of Enum 0, drinks.

eighteenPercent

eighteenPercent()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 388

Returns 18% of the total cost to suggest as a tip

Returns

number

entrees

entrees()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 286

returns all the objects from the database with the mealType of Enum 2, entrees.

fifteenPercent

fifteenPercent()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 363

Returns 15% of the total cost to suggest as a tip

Returns

number

generateReceipt

generateReceipt()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/billsJS.js>, line 57

Generates the bill

generateStats

generateStats(itemCountArray, timePeriod)

<https://github.com/Mitbits/Software-Project/tree/master/app/server/main.js>, line 274

Calculates the cost, profit, revenue of the menu items, and inserts a popularItem object into the database

Arguments

itemCountArray

Count of items sold within a particular time period

timePeriod

getCookTime

getCookTime()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/order.js>, line 149

Gets the cook time of mltemID

Returns

Number

Status of database write operation

getIngredientPrice

getIngredientPrice()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/menuItem.js>, line 136

Gets the price for an individual ingredient identified as mlngID

Returns

Number

Status of database write operation

getNumOccupants

getNumOccupants()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 202

Gets the number of occupants at a table

getRandomInt

getRandomInt()

<https://github.com/Mitbits/Software-Project/tree/master/app/server/dataDriver.js>, line 188

Creates a random integer that is in range [min, max]

Returns

Number

an integer between and including min, max

getRandomNumber

getRandomNumber(min, max)

<https://github.com/Mitbits/Software-Project/tree/master/app/server/main.js>, line 120

Generates a random number

Arguments

min Number

Lower bound of the random number

max

Upper bound of the random number

Returns

Number

getTables

getTables(num, size, table_type, table_status)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 287

gets num tables if possible of tabletype and tablestatus

Arguments

num Number

Number of tables to retreive

size Number

Size of tables to retrieve

table_type TableType

type of tables to get

table_status TableStatus

status of the tables to get

Returns

array of table id's

getTime

getTime(event, \$timeObj)

<https://github.com/Mitbits/Software-Project/tree/master/app/client/order/order.js>, line 194

Gets the current time being shown to the user

Arguments

event

\$timeObj Object

Returns

Object

getTotalCost

getTotalCost()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 316

calculates and returns the total cost of all items on the bill to display.

Returns

number

incrementTimesOrdered

incrementTimesOrdered()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/menuItem.js>, line 118

Increments the timesOrdered attribute by one

Returns

Number

Status of database write operation

incrementTimesUsed

incrementTimesUsed()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/ingredient.js>, line 75

Increments the number of items each item was used

ingredients

ingredients()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/inventory/inventory.js>, line 108

Returns the ingredients in the order of most frequently used

initAvgCookTime

initAvgCookTime()

<https://github.com/Mitbits/Software-Project/tree/master/app/server/dataDriver.js>, line 11

Creates random data of cook time of an item for an entire year and saves it to the database

initAvgIngUsage

initAvgIngUsage()

<https://github.com/Mitbits/Software-Project/tree/master/app/server/dataDriver.js>, line 43

Creates random data of usage of an ingredient for an entire year and saves it to the database

initAvgNumOrders

initAvgNumOrders()

<https://github.com/Mitbits/Software-Project/tree/master/app/server/dataDriver.js>, line 85

Creates random data of number of orders of a menu item for an entire year and saves it to the database

initAvgTimeSpent

initAvgTimeSpent()

<https://github.com/Mitbits/Software-Project/tree/master/app/server/dataDriver.js>, line 115

Creates random data of time spent by a customer at the restaurant every hour for an entire year and saves it to the database

initData

initData()

<https://github.com/Mitbits/Software-Project/tree/master/app/server/dataDriver.js>, line 156

Checks if the data used for statistics exists otherwise creates data

initPopItems

initPopItems()

<https://github.com/Mitbits/Software-Project/tree/master/app/server/dataDriver.js>, line 138

Creates random data of popular items for the week and saves it to the database

inventoryItem

new inventoryItem(invID, invName, invUnits, invQuantity, invPrice, invPerUnit, invThreshold, invTimesUsed)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/ingredient.js>, line 15

- One item in inventory

Arguments

invID Number

itemID in inventory

invName String

Name of item in inventory

invUnits String

Units in inventory

invQuantity Number

Quantity of the item in inventory

invPrice Number

Price of item in inventory

invPerUnit Number

Number of items per unit

invThreshold Number

Threshold for the item in inventory

invTimesUsed Number

Times the item was used in inventory

isClean

isClean()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/floorPlan/floorplan.js>, line 258

Checks if current table is Clean

Returns

Boolean

isDirty

isDirty()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/floorPlan/floorplan.js>, line 246

Checks if current table is Dirty

Returns

Boolean

isMergedTable

isMergedTable()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/floorPlan/floorplan.js>, line 228

filters out tables that are can't be used for manual merge

Returns

Boolean

- whether table is allowed for manual merge

isReserved

isReserved()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/floorPlan/floorplan.js>, line 269

Checks if current table is Reserved

Returns

Boolean

isTaken

isTaken()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/floorPlan/floorplan.js>, line 280

Checks if current table is Taken

Returns

Boolean

isToday

isToday()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/reservation.js>, line 61

Checks if the date of the Reservation corresponds to the current server date

Returns

boolean

True if date matches, false otherwise

isTopOne

isTopOne()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 518

Returns true if the item is first on the top list

Returns

boolean

isTopThree

isTopThree()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 548

Returns true if the item is third on the top list

Returns

boolean

isTopTwo

isTopTwo()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 533

Returns true if the item is second on the top list

Returns

boolean

itemDescription

itemDescription(order)

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 497

returns the description of the menu item for the given order

Arguments

order

itemName

itemName(order)

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 479

returns the name of the menu item for the given order

Arguments

order

itemPrice

itemPrice(order)

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 488

returns the price of the menu item for the given order

Arguments

order

itemsExclusive

itemsExclusive()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/manager/manager.js>, line 167

Returns the revenue of each menu item

itemsExclusive()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/manager/manager.js>, line 167

Returns the revenue of each menu item

itemsExclusive()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/manager/manager.js>, line 167

Returns the revenue of each menu item

itemsExclusive()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/manager/manager.js>, line 167

Returns the revenue of each menu item

itemsExclusive()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/manager/manager.js>, line 167

Returns the revenue of each menu item

itemsHigh

itemsHigh()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/manager/manager.js>, line 102

Returns all the menu items in the high category

itemsLow

itemsLow()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/manager/manager.js>, line 116

Returns all the menu items in the low category

itemsMedium

itemsMedium()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/manager/manager.js>, line 109

Returns all the menu items in the medium category

itemsRank

itemsRank()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/manager/manager.js>, line 123

Returns the rank of each respective menu item

listIngredients

listIngredients()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/kitchen.js>, line 30

Displays the ingredients select feature

listMenuItems

listMenuItems()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/kitchen.js>, line 14

Displays the menuitems select feature

manager_save

manager_save()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 239

saves changes to new manager

menuItems

menuItems()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/stat.js>, line 22

Switches between different statistics on the menu

mergeTable

mergeTable(res)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 335

merges the tables in "tables" and assigns "res" to the resulting table

Arguments

res Reservation

Reservation to assign to merged table

minID

minID

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/order.js>, line 13

Validation parameter for unique ID numbers used in various classes.

nextClick

nextClick()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/archive.js>, line 32

handles the click event to show next transaction on the same day

nextTableId

nextTableId()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 45

looks for next highest table id

notMergeComponent

notMergeComponent()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/floorPlan/floorplan.js>, line 237

filters out tables that are merge components

Returns

Boolean

- whether table is a merge component

orderItem

new orderItem(itemID, priority, menuitemId, specialRequests, isCompleted)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/order.js>, line 27

- Represents an item for an order.

Arguments

itemID Number

Unique identifier representing an item within an order

priority Number

Priority number for item in the order queue

menuitemId Number

Indicates the menuItem this item represents

specialRequests String

Contains any requests by the customer to notify the chef about a particular item

isCompleted Boolean

Indicates whether an orderItem is completed or not

orderList

orderList()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 308

returns all the objects that are part of this table's bill

payBill

payBill()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/billsJS.js>, line 64

Sets the bill status to paid

placeOrder

placeOrder()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/order.js>, line 113

Makes an order available to the order queue by storing an 'order' document in the database

Returns

Number

Status of database write operation

popReservation

popReservation(res)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 272

removes reservation from queue and deletes it

Arguments

res Reservation

Reservation to pop from queue

popularItem

new popularItem(rank, period, cost, profit, revenue)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/mealSuggestions.js>, line 27

Class representing popular items and their attributes.

Arguments

rank Number

The item's overall rank

period POPULARITY_PERIOD

The item's popularity period

cost Number

The total cost of the item in its period

profit Number

The total profit of the item in its period

revenue Number

The total revenue of the item in its period

precisionQuantity

precisionQuantity()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/inventory/inventory.js>, line 181

Converts the quantity of an item to proper decimal places and returns the value

prevClick

prevClick()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/archive.js>, line 22

handles the click event to show previous transaction on the same day

pushReservation

pushReservation(reservation)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 263

pushes new reservation onto queue

Arguments

reservation Reservation

remote_delete

remote_delete()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/reservation.js>, line 78

Deletes a Reservation matching the phone number

reservationBarChart

reservationBarChart()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/reservation.js>, line 8

creates a bar chart of archived reservations

reservation_save

reservation_save()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/reservation.js>, line 53

Saves/Writes reservation into the database

Returns

Number

Status of database write operation

reservations

reservations()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/reservation/reservation.js>, line 114

Returns all Reservations from the database

Returns

Reservation Collection

reservations()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/reservation/reservation.js>, line 114

Returns all Reservations from the database

Returns

Reservation Collection

resetTimer

resetTimer(event, templateInstance)

<https://github.com/Mitbits/Software-Project/tree/master/app/client/order/order.js>, line 258

Resets the timer

Arguments

event

templateInstance

restime

restime()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/floorPlan/floorplan.js>, line 291

Takes current reservation date from database and converts it into 12 Hour Standard Time

Returns

12 Hour Time Format

selected

selected()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 301

returns all the objects from the orderArray reactive variable

setCookTime

setCookTime()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/order.js>, line 140

Sets the actualCookTime attribute of mItemID to mTime

Returns

Number

Status of database write operation

setCookTime()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/order.js>, line 140

Sets the actualCookTime attribute of mItemID to mTime

Returns

Number

Status of database write operation

setCookTime()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/order.js>, line 140

Sets the actualCookTime attribute of mItemID to mTime

Returns

Number

Status of database write operation

setDataToHTML

setDataToHTML()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/stat/archive.js>, line 70

creates the HTML for the bill display

setItemCompleted

setItemCompleted()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/order.js>, line 130

Sets the isCompleted attribute of mItemID to mIsCompleted

Returns

Number

menuItemID of item set

setItemPopularity

setItemPopularity()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/menuitem.js>, line 145

Sets the itemPopularity attribute of this

Returns

Number

Status of database write operation

setOccupantLimit

setOccupantLimit(numOccupants)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 193

Set maximum number of occupants

Arguments

numOccupants Number

limit to set for the number of occupants for a table

setOrderCompleted

setOrderCompleted()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/order.js>, line 121

Sets the isCompleted attribute to mIsCompleted

Returns

Number

Status of database write operation

setPriorityValue

setPriorityValue(mPrtyVal)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/priorityManager.js>, line 348

Sets the priority value of an order item

Arguments

mPrtyVal

the priority value that you want to assign

Returns

Number

Status of database write operation

setTableID

setTableID(tableID)

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 59

sets the tableID to the selected table

Arguments

tableID

shoppingArray

shoppingArray()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/inventory/inventory.js>, line 116

Returns all the items in the shopping list

startCountDown

startCountDown(\$timeObj, duration, resetTimeText, timer)

<https://github.com/Mitbits/Software-Project/tree/master/app/client/order/order.js>, line 235

Starts the count down and appropriately updates the screen

Arguments

\$timeObj Object

duration Number

Time length (seconds)

resetTimeText String

sets screen display to appropriate time (mm:ss (units))

timer CountDownTimer

holds data needed for numerous times

startPollReservations

startPollReservations()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 403

polls for incoming reservations to assign them tables

startTimer

startTimer(event, templateInstance)

<https://github.com/Mitbits/Software-Project/tree/master/app/client/order/order.js>, line 180

When the user clicks on an Order this function will start the timer for that order

Arguments

event

templateInstance

submit

submit #Register-form(event)

<https://github.com/Mitbits/Software-Project/tree/master/app/client/Login/loginJS.js>, line 6

Gets all the values from each field in the registration form and uses the Meteor accounts package to create an account

Arguments

event

submit #login-form(event)

<https://github.com/Mitbits/Software-Project/tree/master/app/client/Login/loginJS.js>, line 53

Gets all the values from the login form and then logs the user in with the Meteor login and redirects them to their respective page.

Arguments

event

submit form

submit form(event)

<https://github.com/Mitbits/Software-Project/tree/master/app/client/reservation/reservation.js>, line 46

Obtains all Elements from reservation page and assigns them into reserve. reserve is then pushed into the database as a reservation.

Arguments

event

subtractQuantity

subtractQuantity(diff)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/ingredient.js>, line 66

Subtracts from the quantity in the inventory

Arguments

diff

tables

tables()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/floorPlan/floorplan.js>, line 202

Returns all Tables from the database

Returns

Table Collection

timeToString

timeToString(min, sec)

<https://github.com/Mitbits/Software-Project/tree/master/app/client/order/order.js>, line 212

Converts numerical time to text form

Arguments

min Number

Minute Input

sec Number

Second Input

Returns

string

timerExpired

timerExpired(\$timeObj)

<https://github.com/Mitbits/Software-Project/tree/master/app/client/order/order.js>, line 274

Highlights the row red when the timer time expires

Arguments

\$timeObj Object

A jquery object of the encapsulates the timer div

topThreeDrinks

topThreeDrinks()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 435

Returns the top three drinks, Appetizers, Entrees and Desserts in the menu

totalCost

totalCost()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/inventory/inventory.js>, line 124

Returns the total cost of all items in the shopping list

twentyPercent

twentyPercent()

<https://github.com/Mitbits/Software-Project/tree/master/app/client/waiter/waiter.js>, line 413

Returns 20% of the total cost to suggest as a tip

Returns

number

unmergeTable

unmergeTable(table_merged_id)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 382

breaks merged table up into its components

Arguments

table_merged_id Number

Id of merged table

updatePriorityManager

updatePriorityManager()

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/priorityManager.js>, line 9

Starts the priority manager, initializes the different variables needed for the function, operates on those variables to calculate the priorities

Returns

Array of orderItemss

An array of order items in the correct order

updateTableStatus

updateTableStatus(toStatus)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 174

Changes the current table status to the specified status

Arguments

toStatus

verifyReservation

verifyReservation(time)

<https://github.com/Mitbits/Software-Project/tree/master/app/imports/api/table.js>, line 247

Checks if a reservation can be added to the wait list.

Arguments

time Number

Time to check the reservations