

[OVERVIEW](#) [PACKAGE](#) [CLASS](#) [USE](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#) [ALL CLASSES](#)[SUMMARY: NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

dataBaseC

Class DatabaseCController

java.lang.Object

dataBaseC.DatabaseCController

```
public class DatabaseCController
extends java.lang.Object
```

Starts the DB C. Chef will use this to continually update the inventory by giving it the name of the dish it started. This will also update the chef when low inventory is met. Waiter will use this when logging on to get the menu.

Author:

cms549

Field Summary

Fields

Modifier and Type	Field and Description
static Sender	chefSender Chef sender that is connected to DB C
static Menu	menu Holds the menu.
static java.util.HashMap<java.lang.Integer, Sender >	waiterSenders List of all the waiter senders that are connected to DB C.

Constructor Summary

Constructors

Constructor and Description
DatabaseCController()

Method Summary

All Methods Static Methods Concrete Methods

Modifier and Type	Method and Description
static int	addDishtoMenu (java.lang.String type, java.lang.String dishname, double price) Adds a dish to the menu
static boolean	addIngredientToInventory (java.lang.String ingredientName, java.lang.Double amountLeft, java.lang.String unitOfAmount, java.lang.Double threshold) Adds ingredient to inventory returns false if ingredient already exists
static void	decrementDish (java.lang.String dishName) Decrements all ingredients in the dish and if low inventory happens will generate the proper notifications.
static void	loadMenuFromFile () Load Menu From menu file
static void	main (java.lang.String[] args) Starts the Listener thread for each socket trying to connect with it.
static void	recordTicket (java.lang.String tick) Adds the ticket to the records
static void	sendLowInventoryNotifications (Ingredient i) Sends chef a low inventory notification also send waiter interfaces a message that will take off the dishes with ingredient i in their menu.

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

waiterSenders

```
public static java.util.HashMap<java.lang.Integer, Sender> waiterSenders
```

List of all the waiter senders that are connected to DB C. Links id of sender socket (generated by this class) to the sender

chefSender

```
public static Sender chefSender
```

Chef sender that is connected to DB C

menu

```
public static Menu menu
```

Holds the menu. Waiter will need this when logging on. See Menu.java

Constructor Detail**DatabaseCController**

```
public DatabaseCController()
```

Method Detail**addIngredientToInventory**

```
public static boolean addIngredientToInventory(java.lang.String ingredientName,  
                                              java.lang.Double amountLeft,  
                                              java.lang.String unitOfAmount,  
                                              java.lang.Double threshold)
```

Adds ingredient to inventory returns false if ingredient already exists

Parameters:

ingredientName - - name of ingredient to add to inventory

amountLeft - - amount of ingredient you have at the time

unitOfAmount - - unit that the amountLeft is measured in

threshold- - the amount at which you wish to be notified when inventory is low for this ingredient if you put it a threshold>=amount left or threshold = 0 then this will fail

Returns:

true on success, false on failure

main

```
public static void main(java.lang.String[] args)
```

Starts the Listener thread for each socket trying to connect with it. The listener will make the sender thread. Sockets should be from chefs or waiters.

Parameters:

args -

addDishtoMenu

```
public static int addDishtoMenu(java.lang.String type,  
                                java.lang.String dishname,  
                                double price)
```

Adds a dish to the menu

Parameters:

type - - type of dish. (Appetizer, Dessert, Entree)

dishname - - name of the dish.

price - - price of the dish.

Returns:

0 on success and -1 on failure/duplicate dish

recordTicket

```
public static void recordTicket(java.lang.String tick)
```

Adds the ticket to the records

Parameters:

tick - - string representation of ticket from waiter interface

decrementDish

```
public static void decrementDish(java.lang.String dishName)
```

Decrements all ingredients in the dish and if low inventory happens will generate the proper notifications.

Parameters:

dishName - - name of dish you are dec ingredients for

sendLowInventoryNotifications

```
public static void sendLowInventoryNotifications(Ingredient i)
```

Sends chef a low inventory notification also send waiter interfaces a message that will take off the dishes with ingredient *i* in their menu.

Parameters:

i - - ingredient that is low

loadMenuFromFile

```
public static void loadMenuFromFile()
```

Load Menu From menu file

[OVERVIEW](#) [PACKAGE](#) [CLASS](#) [USE](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)

[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#) [ALL CLASSES](#)

[SUMMARY: NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)