

[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](#)

databaseA

## Class DatabaseAController

[java.lang.Object](#)[java.lang.Thread](#)[databaseA.DatabaseAController](#)

### All Implemented Interfaces:

[java.lang.Runnable](#)

```
public class DatabaseAController
extends java.lang.Thread
```

Starts the DataBase A. This will handle requests for log in. DB A holds employee info.

### Author:

cms549

### *Nested Class Summary*

#### Nested classes/interfaces inherited from class [java.lang.Thread](#)

[java.lang.Thread.State](#), [java.lang.Thread.UncaughtExceptionHandler](#)

### *Field Summary*

#### Fields inherited from class [java.lang.Thread](#)

[MAX\\_PRIORITY](#), [MIN\\_PRIORITY](#), [NORM\\_PRIORITY](#)

### *Constructor Summary*

#### Constructors

##### Constructor and Description

[DatabaseAController](#)([java.net.Socket](#) listener)

Constructor

## Method Summary

### All Methods

### Static Methods

### Instance Methods

### Concrete Methods

#### Modifier and Type

#### Method and Description

static boolean

**addEmployee**(java.lang.String name,  
char position)

Adds an employee to the database

static java.util.ArrayList<**Employee**>

**getEmployeeListForTesting**()

Used by DatabaseLadFromFileTest.java for testing

static void

**loadEmployees**()

Loads employee list from file

static void

**main**(java.lang.String[] args)

Starts the DataBase A.

void

**run**()

Starts a new thread for the DB A controller so it  
can communicate with one tablet on one thread.

## Methods inherited from class java.lang.Thread

activeCount, checkAccess, countStackFrames, currentThread, destroy,  
dumpStack, enumerate, getAllStackTraces, getContextClassLoader,  
getDefaultUncaughtExceptionHandler, getId, getName, getPriority,  
getStackTrace, getState, getThreadGroup, getUncaughtExceptionHandler,  
holdsLock, interrupt, interrupted, isAlive, isDaemon, isInterrupted, join,  
join, join, resume, setContextClassLoader, setDaemon,  
setDefaultUncaughtExceptionHandler, setName, setPriority,  
setUncaughtExceptionHandler, sleep, sleep, start, stop, stop, suspend,  
toString, yield

## Methods inherited from class java.lang.Object

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

## Constructor Detail

### DatabaseAController

```
public DatabaseAController(java.net.Socket listener)
```

Constructor

**Parameters:**

listener - - socket to one tablet that database A will use to communicate with a tablet to log it on properly.

**Method Detail****addEmployee**

```
public static boolean addEmployee(java.lang.String name,
                                   char position)
```

Adds an employee to the database

**Parameters:**

name - - name of employee

position - - position of employee w = waiter, h = host, c = chef, m = manager, o = owner

**Returns:**

true on success, false if you did not put in a valid position or if name is null

**run**

```
public void run()
```

Starts a new thread for the DB A controller so it can communicate with one tablet on one thread. Reads the socket in and looks for a 'L' (which signifies logging in) or 'O' (logging out) Next it writes to the socket the position of the employee if they are logging in or 'o' if the id is not valid it writes L if the employee is already logged in on a log out it changes the employee with that id to logged out This ensures that an employee doesn't log on to two devices. This is also used by the host when the host needs the list of employees and their ids. The socket will hang up after it logs a user in or out.

**Specified by:**

run in interface java.lang.Runnable

**Overrides:**

run in class java.lang.Thread

**main**

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

Starts the DataBase A. Creates a new thread for each log in request.

**Parameters:**

args -

### loadEmployees

```
public static void loadEmployees()
```

Loads employee list from file

### getEmployeeListForTesting

```
public static java.util.ArrayList<Employee> getEmployeeListForTesting()
```

Used by DatabaseLadFromFileTest.java for testing

**Returns:**

list of employees

[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](#)