JavaScript is disabled on your browser.

[Skip navigation links](#1fob9te)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/MediaRentalManagerInt.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* Prev Class
* Next Class
* [Frames](http://docs.google.com/index.html?mediaRentalManager/MediaRentalManagerInt.html)
* [No Frames](http://docs.google.com/MediaRentalManagerInt.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* Constr |
* [Method](#3znysh7)
* Detail:
* Field |
* Constr |
* [Method](#2et92p0)

mediaRentalManager

## Interface MediaRentalManagerInt

* public interface MediaRentalManagerInt  
  Interface that defines the functionality expected from the media rental manager. The two possible media we can have are movies and music albums. A movie has a title, a number of copies that are available for rent, and a rating (e.g., "PG"). An album has a title, a number of copies, an artist, and a list of songs (String with title of songs separated by commas).  
    
  **IMPORTANT:**The database of the media rental manager, must define and use two ArrayList. One stores the media (both Movies and Album information) and one stores Customer information. You will lose significant credit if you do not define and use these ArrayList objects.Author: cmsc131

### Method SummaryAll Methods Instance Methods Abstract Methods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + void | * + [addAlbum](http://docs.google.com/mediaRentalManager/MediaRentalManagerInt.html#addAlbum-java.lang.String-int-java.lang.String-java.lang.String-)(java.lang.String title, int copiesAvailable, java.lang.String artist, java.lang.String songs) Adds the specified album to the database. |
| * + void | * + [addCustomer](http://docs.google.com/mediaRentalManager/MediaRentalManagerInt.html#addCustomer-java.lang.String-java.lang.String-java.lang.String-)(java.lang.String name, java.lang.String address, java.lang.String plan) Adds the specified customer to the database. |
| * + void | * + [addMovie](http://docs.google.com/mediaRentalManager/MediaRentalManagerInt.html#addMovie-java.lang.String-int-java.lang.String-)(java.lang.String title, int copiesAvailable, java.lang.String rating) Adds the specified movie to the database. |
| * + boolean | * + [addToQueue](http://docs.google.com/mediaRentalManager/MediaRentalManagerInt.html#addToQueue-java.lang.String-java.lang.String-)(java.lang.String customerName, java.lang.String mediaTitle) Adds the specified media title to the queue associated with a customer. |
| * + java.lang.String | * + [getAllCustomersInfo](http://docs.google.com/mediaRentalManager/MediaRentalManagerInt.html#getAllCustomersInfo--)() Returns information about the customers in the database. |
| * + java.lang.String | * + [getAllMediaInfo](http://docs.google.com/mediaRentalManager/MediaRentalManagerInt.html#getAllMediaInfo--)() Returns information about all the media (movies and albums) that are part of the database. |
| * + java.lang.String | * + [processRequests](http://docs.google.com/mediaRentalManager/MediaRentalManagerInt.html#processRequests--)() Processes the requests queue of each customer. |
| * + boolean | * + [removeFromQueue](http://docs.google.com/mediaRentalManager/MediaRentalManagerInt.html#removeFromQueue-java.lang.String-java.lang.String-)(java.lang.String customerName, java.lang.String mediaTitle) Removes the specified media title from the customer's queue. |
| * + boolean | * + [returnMedia](http://docs.google.com/mediaRentalManager/MediaRentalManagerInt.html#returnMedia-java.lang.String-java.lang.String-)(java.lang.String customerName, java.lang.String mediaTitle) This is how a customer returns a rented media. |
| * + java.util.ArrayList<java.lang.String> | * + [searchMedia](http://docs.google.com/mediaRentalManager/MediaRentalManagerInt.html#searchMedia-java.lang.String-java.lang.String-java.lang.String-java.lang.String-)(java.lang.String title, java.lang.String rating, java.lang.String artist, java.lang.String songs) Returns a SORTED ArrayList with media titles that satisfy the provided parameter values. |
| * + void | * + [setLimitedPlanLimit](http://docs.google.com/mediaRentalManager/MediaRentalManagerInt.html#setLimitedPlanLimit-int-)(int value) This set the number of media associated with the LIMITED plan. |

### Method Detail

#### addCustomer void addCustomer(java.lang.String name, java.lang.String address, java.lang.String plan) Adds the specified customer to the database. The address is a physical address (not e-mail). The plan options available are: **LIMITED** and **UNLIMITED**. LIMITED defines a default maximum of two media that can be rented.Parameters: name - address - plan -

#### addMovie void addMovie(java.lang.String title, int copiesAvailable, java.lang.String rating) Adds the specified movie to the database. The possible values for rating are "PG", "R", "NR".Parameters: title - copiesAvailable - rating -

#### addAlbum void addAlbum(java.lang.String title, int copiesAvailable, java.lang.String artist, java.lang.String songs) Adds the specified album to the database. The songs String includes a list of the title of songs in the album (song titles are separated by commas).Parameters: title - copiesAvailable - artist - songs -

#### setLimitedPlanLimit void setLimitedPlanLimit(int value) This set the number of media associated with the LIMITED plan.Parameters: value -

#### getAllCustomersInfo java.lang.String getAllCustomersInfo() Returns information about the customers in the database. The information is presented sorted by customer name. See the public tests for the format to use.Returns:

#### getAllMediaInfo java.lang.String getAllMediaInfo() Returns information about all the media (movies and albums) that are part of the database. The information is presented sorted by media title. See the public tests for the format to use.Returns:

#### addToQueue boolean addToQueue(java.lang.String customerName, java.lang.String mediaTitle) Adds the specified media title to the queue associated with a customer.Parameters: customerName - mediaTitle - Returns: false if the mediaTitle is already part of the queue (it will not be added)

#### removeFromQueue boolean removeFromQueue(java.lang.String customerName, java.lang.String mediaTitle) Removes the specified media title from the customer's queue.Parameters: customerName - mediaTitle - Returns: false if removal failed for any reason (e.g., customerName not found)

* + - processRequests  
      java.lang.String processRequests()  
      Processes the requests queue of each customer. The customers will be processed in alphabetical order. For each customer, the requests queue will be checked and media will be added to the rented queue, if the plan associated with the customer allows it, and if there is a copy of the media available. For UNLIMITED plans the media will be added to the rented queue always, as long as there are copies associated with the media available. For LIMITED plans, the number of entries moved from the requests queue to the rented queue will depend on the number of currently rented media, and whether copies associated with the media are available.  
        
      For each media that is rented, the following message will be generated:  
      "Sending [mediaTitle] to [customerName]"Returns:

#### returnMedia boolean returnMedia(java.lang.String customerName, java.lang.String mediaTitle) This is how a customer returns a rented media. This method will remove the item from the rented queue and adjust any other values that are necessary (e.g., copiesAvailable)Parameters: customerName - mediaTitle - Returns:

#### searchMedia java.util.ArrayList<java.lang.String> searchMedia(java.lang.String title, java.lang.String rating, java.lang.String artist, java.lang.String songs) Returns a SORTED ArrayList with media titles that satisfy the provided parameter values. If null is specified for a parameter, then that parameter should be ignore in the search. Providing null for all parameters will return all media titles.Parameters: title - rating - artist - songs - Returns:

[Skip navigation links](#z337ya)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/MediaRentalManagerInt.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* Prev Class
* Next Class
* [Frames](http://docs.google.com/index.html?mediaRentalManager/MediaRentalManagerInt.html)
* [No Frames](http://docs.google.com/MediaRentalManagerInt.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* Constr |
* [Method](#3znysh7)
* Detail:
* Field |
* Constr |
* [Method](#2et92p0)