|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | [**Package**](http://docs.google.com/rain/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MusicDAO.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-all.html) | | |  |
| [**PREV CLASS**](http://docs.google.com/rain/MusicCatalogServlet.html)   [**NEXT CLASS**](http://docs.google.com/rain/MusicRecording.html) | [**FRAMES**](http://docs.google.com/index.html?rain/MusicDAO.html)    [**NO FRAMES**](http://docs.google.com/MusicDAO.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#3rdcrjn) | [METHOD](#lnxbz9) |

## **rain**

Class MusicDAO

[java.lang.Object](http://docs.google.com/docs/jdk_docs/api/java/lang/Object.html)  
 [rain.DAO](http://docs.google.com/rain/DAO.html)  
 **rain.MusicDAO**

public class **MusicDAO**extends [DAO](http://docs.google.com/rain/DAO.html)

This class provides a data access mechanism for the music database. Methods are available to get a list of the music categories and a list of recordings.

Usage Example:

// create the data accessor  
 MusicDAO myMusicDAO = new MusicDAO(theDataSource);  
  
 // get a list of available categories;  
 List<String> cats = myMusicDAO.getMusicCategories();  
 ...  
  
 // get a list of jazz recordings  
 List<MusicRecording> jazzRecordingList = myMusicDAO.getMusicRecordings("Jazz");  
 ...  
  
 // get a list of jazz recordings sorted by price (ascending)  
 List<MusicRecording> jazzRecordingList = myMusicDAO.getMusicRecordings("Jazz", MusicDAO.SORT\_BY\_PRICE);  
 ...

**Author:** 570 Development Team

|  |  |
| --- | --- |
| **Field Summary** | |
| static int | [**SORT\_BY\_ARTIST**](http://docs.google.com/rain/MusicDAO.html#SORT_BY_ARTIST)            Flag to sort by artist name |

|  |
| --- |
| **Fields inherited from class rain.**[**DAO**](http://docs.google.com/rain/DAO.html) |
| [dataSource](http://docs.google.com/rain/DAO.html#dataSource), [logger](http://docs.google.com/rain/DAO.html#logger), [SORT\_BY\_PRICE](http://docs.google.com/rain/DAO.html#SORT_BY_PRICE), [SORT\_BY\_STOCK\_COUNT](http://docs.google.com/rain/DAO.html#SORT_BY_STOCK_COUNT), [SORT\_BY\_TITLE](http://docs.google.com/rain/DAO.html#SORT_BY_TITLE) |

|  |  |
| --- | --- |
| **Constructor Summary** | |
| [**MusicDAO**](http://docs.google.com/rain/MusicDAO.html#MusicDAO(javax.sql.DataSource))([DataSource](http://docs.google.com/docs/jdk_docs/api/javax/sql/DataSource.html) theDataSource)            Constructs the data accessor with the given data source |

|  |  |
| --- | --- |
| **Method Summary** | |
| [List](http://docs.google.com/docs/jdk_docs/api/java/util/List.html)<[String](http://docs.google.com/docs/jdk_docs/api/java/lang/String.html)> | [**getMusicCategories**](http://docs.google.com/rain/MusicDAO.html#getMusicCategories())()            Get a list of music categories from the database |
| [MusicRecording](http://docs.google.com/rain/MusicRecording.html) | [**getMusicRecording**](http://docs.google.com/rain/MusicDAO.html#getMusicRecording(int))(int recordingId)            Returns a music recording based on the id |
| [MusicRecording](http://docs.google.com/rain/MusicRecording.html) | [**getMusicRecording**](http://docs.google.com/rain/MusicDAO.html#getMusicRecording(java.lang.String))([String](http://docs.google.com/docs/jdk_docs/api/java/lang/String.html) recordingId)            Returns a music recording based on the id |
| [List](http://docs.google.com/docs/jdk_docs/api/java/util/List.html)<[MusicRecording](http://docs.google.com/rain/MusicRecording.html)> | [**getMusicRecordings**](http://docs.google.com/rain/MusicDAO.html#getMusicRecordings(java.lang.String))([String](http://docs.google.com/docs/jdk_docs/api/java/lang/String.html) theCategory)            Get a list of video recordings from the database for the given category. |
| [List](http://docs.google.com/docs/jdk_docs/api/java/util/List.html)<[MusicRecording](http://docs.google.com/rain/MusicRecording.html)> | [**getMusicRecordings**](http://docs.google.com/rain/MusicDAO.html#getMusicRecordings(java.lang.String,%20int))([String](http://docs.google.com/docs/jdk_docs/api/java/lang/String.html) theCategory, int sortBy)            Get a list of music recordings from the database for the given category The sort order is ascending. |

|  |
| --- |
| **Methods inherited from class rain.**[**DAO**](http://docs.google.com/rain/DAO.html) |
| [cleanup](http://docs.google.com/rain/DAO.html#cleanup(java.sql.ResultSet,%20java.sql.Statement,%20java.sql.Connection)), [log](http://docs.google.com/rain/DAO.html#log(java.lang.Object)), [logError](http://docs.google.com/rain/DAO.html#logError(java.lang.Object,%20java.lang.Throwable)), [logError](http://docs.google.com/rain/DAO.html#logError(java.lang.Throwable)), [setDataSource](http://docs.google.com/rain/DAO.html#setDataSource(javax.sql.DataSource)) |

|  |
| --- |
| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/docs/jdk_docs/api/java/lang/Object.html) |
| [clone](http://docs.google.com/docs/jdk_docs/api/java/lang/Object.html#clone()), [equals](http://docs.google.com/docs/jdk_docs/api/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/docs/jdk_docs/api/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/docs/jdk_docs/api/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/docs/jdk_docs/api/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/docs/jdk_docs/api/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/docs/jdk_docs/api/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/docs/jdk_docs/api/java/lang/Object.html#toString()), [wait](http://docs.google.com/docs/jdk_docs/api/java/lang/Object.html#wait()), [wait](http://docs.google.com/docs/jdk_docs/api/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/docs/jdk_docs/api/java/lang/Object.html#wait(long,%20int)) |

|  |
| --- |
| **Field Detail** |

### SORT\_BY\_ARTIST

public static final int **SORT\_BY\_ARTIST**

Flag to sort by artist name

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#rain.MusicDAO.SORT_BY_ARTIST)

|  |
| --- |
| **Constructor Detail** |

### MusicDAO

public **MusicDAO**([DataSource](http://docs.google.com/docs/jdk_docs/api/javax/sql/DataSource.html) theDataSource)

Constructs the data accessor with the given data source

**Throws:** [DAOException](http://docs.google.com/rain/DAOException.html) - error accessing the data source

|  |
| --- |
| **Method Detail** |

### getMusicCategories

public [List](http://docs.google.com/docs/jdk_docs/api/java/util/List.html)<[String](http://docs.google.com/docs/jdk_docs/api/java/lang/String.html)> **getMusicCategories**()  
 throws [DAOException](http://docs.google.com/rain/DAOException.html)

Get a list of music categories from the database

**Returns:**a list of categories **Throws:** [DAOException](http://docs.google.com/rain/DAOException.html) - error accessing the data source

### getMusicRecordings

public [List](http://docs.google.com/docs/jdk_docs/api/java/util/List.html)<[MusicRecording](http://docs.google.com/rain/MusicRecording.html)> **getMusicRecordings**([String](http://docs.google.com/docs/jdk_docs/api/java/lang/String.html) theCategory)  
 throws [DAOException](http://docs.google.com/rain/DAOException.html)

Get a list of video recordings from the database for the given category.

The list is sorted by artist name.

**Returns:**a list of MusicRecording objects **Throws:** [DAOException](http://docs.google.com/rain/DAOException.html) - error accessing the data source**See Also:**[getMusicRecordings(String theCategory, int sortBy)](http://docs.google.com/rain/MusicDAO.html#getMusicRecordings(java.lang.String,%20int))

### getMusicRecordings

public [List](http://docs.google.com/docs/jdk_docs/api/java/util/List.html)<[MusicRecording](http://docs.google.com/rain/MusicRecording.html)> **getMusicRecordings**([String](http://docs.google.com/docs/jdk_docs/api/java/lang/String.html) theCategory,  
 int sortBy)  
 throws [DAOException](http://docs.google.com/rain/DAOException.html)

Get a list of music recordings from the database for the given category The sort order is ascending.

**Returns:**a list of sorted MusicRecording objects **Throws:** [SQLException](http://docs.google.com/docs/jdk_docs/api/java/sql/SQLException.html) - thrown for SQL errors [DAOException](http://docs.google.com/rain/DAOException.html) - error accessing the data source**See Also:**[DAO.SORT\_BY\_TITLE](http://docs.google.com/rain/DAO.html#SORT_BY_TITLE), [DAO.SORT\_BY\_PRICE](http://docs.google.com/rain/DAO.html#SORT_BY_PRICE), [DAO.SORT\_BY\_STOCK\_COUNT](http://docs.google.com/rain/DAO.html#SORT_BY_STOCK_COUNT), [SORT\_BY\_ARTIST](http://docs.google.com/rain/MusicDAO.html#SORT_BY_ARTIST)

### getMusicRecording

public [MusicRecording](http://docs.google.com/rain/MusicRecording.html) **getMusicRecording**([String](http://docs.google.com/docs/jdk_docs/api/java/lang/String.html) recordingId)  
 throws [DAOException](http://docs.google.com/rain/DAOException.html)

Returns a music recording based on the id

**Throws:** [DAOException](http://docs.google.com/rain/DAOException.html) - error accessing the data source**See Also:**[getMusicRecording(int recordingId)](http://docs.google.com/rain/MusicDAO.html#getMusicRecording(int))

### getMusicRecording

public [MusicRecording](http://docs.google.com/rain/MusicRecording.html) **getMusicRecording**(int recordingId)  
 throws [DAOException](http://docs.google.com/rain/DAOException.html)

Returns a music recording based on the id

**Throws:** [DAOException](http://docs.google.com/rain/DAOException.html) - error accessing the data source

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | [**Package**](http://docs.google.com/rain/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MusicDAO.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-all.html) | | |  |
| [**PREV CLASS**](http://docs.google.com/rain/MusicCatalogServlet.html)   [**NEXT CLASS**](http://docs.google.com/rain/MusicRecording.html) | [**FRAMES**](http://docs.google.com/index.html?rain/MusicDAO.html)    [**NO FRAMES**](http://docs.google.com/MusicDAO.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#3rdcrjn) | [METHOD](#lnxbz9) |