

Gestor de textos i cites | Alexis Rico and Jordi Romero
P001

Generated by Doxygen 1.8.8

Thu Apr 21 2016 10:20:37

Contents

1	Gestor de textos i cites Alexis Rico and Jordi Romero	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	Author Class Reference	7
4.1.1	Detailed Description	7
4.1.2	Constructor & Destructor Documentation	7
4.1.2.1	Author	7
4.1.2.2	~Author	7
4.2	Book Class Reference	7
4.2.1	Detailed Description	8
4.2.2	Constructor & Destructor Documentation	8
4.2.2.1	Book	8
4.2.2.2	Book	8
4.2.2.3	~Book	8
4.2.3	Member Function Documentation	8
4.2.3.1	readBookContent	8
4.2.3.2	getBookTitle	8
4.2.3.3	getAuthorName	8
4.2.3.4	getBookLines	8
4.2.3.5	getBookWords	8
4.2.3.6	replaceWords	8
4.2.3.7	printAllLines	8
4.2.3.8	printLines	8
4.2.3.9	printSelectLines	8
4.2.3.10	printFrequencyTable	8
4.3	Library Class Reference	8

4.3.1	Detailed Description	9
4.3.2	Constructor & Destructor Documentation	9
4.3.2.1	Library	9
4.3.2.2	~Library	9
4.3.3	Member Function Documentation	9
4.3.3.1	readBook	9
4.3.3.2	selectBook	9
4.3.3.3	deleteBook	9
4.3.3.4	deleteQuote	9
4.3.3.5	getBook	9
4.3.3.6	getQuote	9
4.3.3.7	printAuthors	9
4.3.3.8	printBooks	9
4.3.3.9	printQuotes	9
4.3.3.10	printBooksByAuthor	9
4.3.3.11	printQuotesByAuthor	9
4.4	Quote Class Reference	9
4.4.1	Detailed Description	10
4.4.2	Constructor & Destructor Documentation	10
4.4.2.1	Quote	10
4.4.2.2	~Quote	10
4.4.3	Member Function Documentation	10
4.4.3.1	printQuoteInformation	10
5	File Documentation	11
5.1	Author.hh File Reference	11
5.1.1	Detailed Description	11
5.2	Book.hh File Reference	11
5.2.1	Detailed Description	12
5.3	Library.hh File Reference	12
5.3.1	Detailed Description	13
5.4	main.cc File Reference	13
5.4.1	Detailed Description	14
5.4.2	Function Documentation	14
5.4.2.1	readActions	14
5.4.2.2	main	15
5.4.3	Variable Documentation	15
5.4.3.1	BOOK_INSERT	15
5.4.3.2	BOOK_DELETE	15
5.4.3.3	BOOK_SELECT	15

5.4.3.4	BOOK_REPLACE_WORD	15
5.4.3.5	QUOTE_INSERT	15
5.4.3.6	QUOTE_DELETE	15
5.4.3.7	QUERY_AUTHORS	15
5.4.3.8	QUERY_BOOKS_ALL	16
5.4.3.9	QUERY_BOOKS_BY_AUTHOR	16
5.4.3.10	QUERY_CURRENT_AUTHOR	16
5.4.3.11	QUERY_CURRENT_CONTENT	16
5.4.3.12	QUERY_CURRENT_INFO	16
5.4.3.13	QUERY_CURRENT_EXPRESION	16
5.4.3.14	QUERY_CURRENT_LINES	16
5.4.3.15	QUERY_CURRENT_WORDS	16
5.4.3.16	QUERY_CURRENT_FREQUENCY	16
5.4.3.17	QUERY_CURRENT_QUOTES	16
5.4.3.18	QUERY_QUOTES_ALL	16
5.4.3.19	QUERY_QUOTES_BY_AUTHOR	16
5.4.3.20	QUERY_QUOTE_INFO	17
5.4.3.21	QUIT	17
5.5	Quote.hh File Reference	17
5.5.1	Detailed Description	17
Index		18

Chapter 1

**Gestor de textos i cites | Alexis Rico and Jordi
Romero**

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Author	7
Book	7
Library	8
Quote	
Data model for Quotes (cites) from Books	9

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Author.hh	Data model that hosts information about an Author	11
Book.hh	Data model that hosts information about a Book	11
Library.hh	Main structure of our Library with several dependant data childs	12
main.cc	Main program that handles the I/O operations with the end user	13
Quote.hh	Data model that hosts information about a Quote	17

Chapter 4

Class Documentation

4.1 Author Class Reference

Public Member Functions

- [Author](#) ()
- [~Author](#) ()

4.1.1 Detailed Description

Definition at line 13 of file Author.hh.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Author::Author ()

4.1.2.2 Author::~~Author ()

The documentation for this class was generated from the following file:

- [Author.hh](#)

4.2 Book Class Reference

Public Member Functions

- [Book](#) ()
- [Book](#) (string title, string author)
- [~Book](#) ()
- void [readBookContent](#) ()
- string [getBookTitle](#) () const
- string [getAuthorName](#) ()
- int [getBookLines](#) ()
- int [getBookWords](#) ()
- void [replaceWords](#) (string oldWord, string newWord)
- void [printAllLines](#) ()
- void [printLines](#) (string query)
- void [printSelectLines](#) (int start, int end)
- void [printFrequencyTable](#) ()

4.2.1 Detailed Description

Definition at line 16 of file Book.hh.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 `Book::Book ()`

4.2.2.2 `Book::Book (string title, string author)`

4.2.2.3 `Book::~Book ()`

4.2.3 Member Function Documentation

4.2.3.1 `void Book::readBookContent ()`

4.2.3.2 `string Book::getBookTitle () const`

4.2.3.3 `string Book::getAuthorName ()`

4.2.3.4 `int Book::getBookLines ()`

4.2.3.5 `int Book::getBookWords ()`

4.2.3.6 `void Book::replaceWords (string oldWord, string newWord)`

4.2.3.7 `void Book::printAllLines ()`

4.2.3.8 `void Book::printLines (string query)`

4.2.3.9 `void Book::printSelectLines (int start, int end)`

4.2.3.10 `void Book::printFrequencyTable ()`

The documentation for this class was generated from the following file:

- [Book.hh](#)

4.3 Library Class Reference

Public Member Functions

- [Library](#) ()
- [~Library](#) ()
- void [readBook](#) (string title, string author)
- void [selectBook](#) (string query)
- void [deleteBook](#) ()
- void [deleteQuote](#) (string id)
- [Book](#) [getBook](#) ()
- [Quote](#) [getQuote](#) (string id)
- void [printAuthors](#) ()
- void [printBooks](#) ()
- void [printQuotes](#) ()
- void [printBooksByAuthor](#) (string author)
- void [printQuotesByAuthor](#) (string author)

4.3.1 Detailed Description

Definition at line 18 of file Library.hh.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 `Library::Library ()`

4.3.2.2 `Library::~~Library ()`

4.3.3 Member Function Documentation

4.3.3.1 `void Library::readBook (string title, string author)`

4.3.3.2 `void Library::selectBook (string query)`

4.3.3.3 `void Library::deleteBook ()`

4.3.3.4 `void Library::deleteQuote (string id)`

4.3.3.5 `Book Library::getBook ()`

4.3.3.6 `Quote Library::getQuote (string id)`

4.3.3.7 `void Library::printAuthors ()`

4.3.3.8 `void Library::printBooks ()`

4.3.3.9 `void Library::printQuotes ()`

4.3.3.10 `void Library::printBooksByAuthor (string author)`

4.3.3.11 `void Library::printQuotesByAuthor (string author)`

The documentation for this class was generated from the following file:

- [Library.hh](#)

4.4 Quote Class Reference

Data model for Quotes (cites) from Books.

Public Member Functions

- [Quote](#) ()
Creates an empty [Quote](#).
- [~Quote](#) ()
Destructs the implicit [Quote](#).
- void [printQuoteInformation](#) ()
Print all the information of the implicit [Quote](#).

4.4.1 Detailed Description

Data model for Quotes (cites) from Books.

Definition at line 19 of file Quote.hh.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 Quote::Quote ()

Creates an empty [Quote](#).

Precondition

True

Postcondition

Returns an implicit empty quote

4.4.2.2 Quote::~~Quote ()

Destructs the implicit [Quote](#).

Precondition

An implicit [Quote](#)

Postcondition

Deletes the implicit quote

4.4.3 Member Function Documentation

4.4.3.1 void Quote::printQuoteInformation ()

Print all the information of the implicit [Quote](#).

Precondition

An implicit [Quote](#)

Postcondition

Prints all the information

The documentation for this class was generated from the following file:

- [Quote.hh](#)

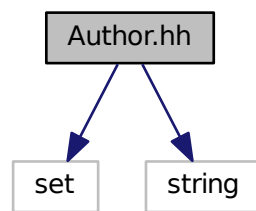
Chapter 5

File Documentation

5.1 Author.hh File Reference

Data model that hosts information about an [Author](#).

Include dependency graph for Author.hh:



Classes

- class [Author](#)

5.1.1 Detailed Description

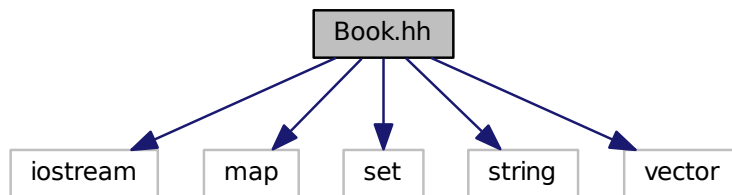
Data model that hosts information about an [Author](#).

Definition in file [Author.hh](#).

5.2 Book.hh File Reference

Data model that hosts information about a [Book](#).

Include dependency graph for Book.hh:



Classes

- class [Book](#)

5.2.1 Detailed Description

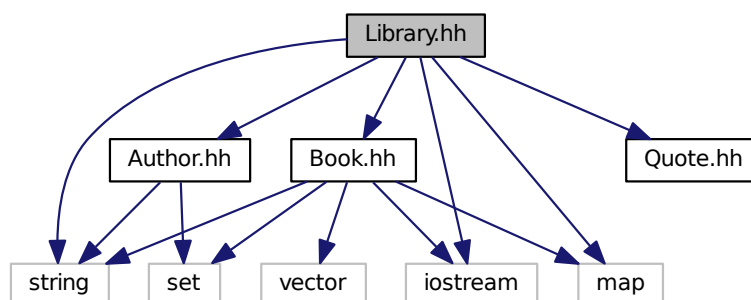
Data model that hosts information about a [Book](#).

Definition in file [Book.hh](#).

5.3 Library.hh File Reference

Main structure of our [Library](#) with several dependant data childs.

Include dependency graph for Library.hh:



Classes

- class [Library](#)

5.3.1 Detailed Description

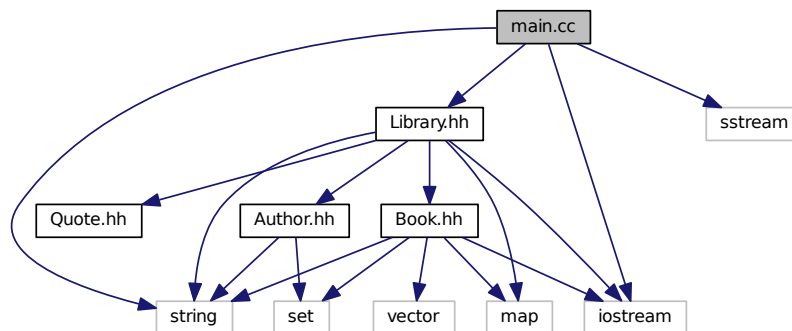
Main structure of our [Library](#) with several dependant data childs.

Definition in file [Library.hh](#).

5.4 main.cc File Reference

Main program that handles the I/O operations with the end user.

Include dependency graph for main.cc:



Functions

- void [readActions](#) ([Library](#) &library)
- int [main](#) ()

Main Procedure of [main.cc](#).

Variables

- const string [BOOK_INSERT](#) = "afegir text"
- const string [BOOK_DELETE](#) = "eliminar text"
- const string [BOOK_SELECT](#) = "triar text"
- const string [BOOK_REPLACE_WORD](#) = "substitueix"
- const string [QUOTE_INSERT](#) = "afegir cita"
- const string [QUOTE_DELETE](#) = "eliminar cita"
- const string [QUERY_AUTHORS](#) = "tots autors ?"
- const string [QUERY_BOOKS_ALL](#) = "tots textos ?"
- const string [QUERY_BOOKS_BY_AUTHOR](#) = "textos autor"
- const string [QUERY_CURRENT_AUTHOR](#) = "autor ?"
- const string [QUERY_CURRENT_CONTENT](#) = "contingut ?"
- const string [QUERY_CURRENT_INFO](#) = "info ?"
- const string [QUERY_CURRENT_EXPRESION](#) = "frases"
- const string [QUERY_CURRENT_LINES](#) = "nombre de frases ?"
- const string [QUERY_CURRENT_WORDS](#) = "nombre de paraules ?"
- const string [QUERY_CURRENT_FREQUENCY](#) = "taula de freqüencies ?"
- const string [QUERY_CURRENT_QUOTES](#) = "cites ?"
- const string [QUERY_QUOTES_ALL](#) = "totes cites ?"

- const string `QUERY_QUOTES_BY_AUTHOR` = "cites autor"
- const string `QUERY_QUOTE_INFO` = "info cita"
- const string `QUIT` = "sortir"

5.4.1 Detailed Description

Main program that handles the I/O operations with the end user.

Definition in file [main.cc](#).

5.4.2 Function Documentation

5.4.2.1 void readActions (Library & library)

Definition at line 44 of file [main.cc](#).

```

44         {
45     string input;
46     while (getline(cin, input)) {
47         if (input.substr(0, BOOK_INSERT.length()) == BOOK_INSERT) {
48             string title, author;
49             title = input.erase(input.length() - 1, 1).substr(input.find_first_of("\"") + 1);
50             getline(cin, input);
51             author = input.erase(input.length() - 1, 1).substr(input.find_first_of("\"") + 1);
52             library.readBook(title, author);
53         } else if (input.substr(0, BOOK_DELETE.length()) ==
BOOK_DELETE) {
54             library.deleteBook();
55         } else if (input.substr(0, BOOK_SELECT.length()) ==
BOOK_SELECT) {
56             library.selectBook(input.erase(input.length() - 1, 1).substr(input.find_first_of("{")
+ 1));
57         } else if (input.substr(0, BOOK_REPLACE_WORD.length()) ==
BOOK_REPLACE_WORD) {
58             string oldWord, newWord;
59             // TODO: Substring from input
60             library.getBook().replaceWords(oldWord, newWord);
61         } else if (input.substr(0, QUOTE_INSERT.length()) ==
QUOTE_INSERT) {
62             // TODO
63         } else if (input.substr(0, QUOTE_DELETE.length()) ==
QUOTE_DELETE) {
64             string reference;
65             // TODO: Substring reference from input
66             library.deleteQuote(reference);
67         } else if (input.substr(0, QUERY_AUTHORS.length()) ==
QUERY_AUTHORS) {
68             // TODO
69         } else if (input.substr(0, QUERY_BOOKS_ALL.length()) ==
QUERY_BOOKS_ALL) {
70             // TODO
71         } else if (input.substr(0, QUERY_BOOKS_BY_AUTHOR.length()) ==
QUERY_BOOKS_BY_AUTHOR) {
72             // TODO
73         } else if (input.substr(0, QUERY_CURRENT_AUTHOR.length()) ==
QUERY_CURRENT_AUTHOR) {
74             // TODO
75         } else if (input.substr(0, QUERY_CURRENT_CONTENT.length()) ==
QUERY_CURRENT_CONTENT) {
76             // TODO
77         } else if (input.substr(0, QUERY_CURRENT_INFO.length()) ==
QUERY_CURRENT_INFO) {
78             // TODO
79         } else if (input.substr(0, QUERY_CURRENT_EXPRESION.length()) ==
QUERY_CURRENT_EXPRESION) {
80             // TODO
81         } else if (input.substr(0, QUERY_CURRENT_LINES.length()) ==
QUERY_CURRENT_LINES) {
82             // TODO
83         } else if (input.substr(0, QUERY_CURRENT_WORDS.length()) ==
QUERY_CURRENT_WORDS) {
84             // TODO
85         } else if (input.substr(0, QUERY_CURRENT_FREQUENCY.length()) ==
QUERY_CURRENT_FREQUENCY) {
86             // TODO
87         } else if (input.substr(0, QUERY_CURRENT_QUOTES.length()) ==
QUERY_CURRENT_QUOTES) {

```

```

88         // TODO
89     } else if (input.substr(0, QUERY_QUOTES_ALL.length()) ==
QUERY_QUOTES_ALL) {
90         library.printQuotes();
91     } else if (input.substr(0, QUERY_QUOTES_BY_AUTHOR.length()) ==
QUERY_QUOTES_BY_AUTHOR) {
92         string author;
93         // TODO: Substring author from input
94         library.printQuotesByAuthor(author);
95     } else if (input.substr(0, QUERY_QUOTE_INFO.length()) ==
QUERY_QUOTE_INFO) {
96         string reference;
97         // TODO: Substring reference from input
98         library.getQuote(reference).printQuoteInformation();
99     } else if (input.substr(0, QUIT.length()) == QUIT) {
100         // TODO
101     }
102 }
103 }

```

5.4.2.2 int main ()

Main Procedure of [main.cc](#).

Definition at line 106 of file main.cc.

```

106     {
107         Library mainLibrary;
108         readActions(mainLibrary);
109     }

```

5.4.3 Variable Documentation

5.4.3.1 const string BOOK_INSERT = "afegir text"

Definition at line 17 of file main.cc.

5.4.3.2 const string BOOK_DELETE = "eliminar text"

Definition at line 18 of file main.cc.

5.4.3.3 const string BOOK_SELECT = "triar text"

Definition at line 19 of file main.cc.

5.4.3.4 const string BOOK_REPLACE_WORD = "substitueix"

Definition at line 20 of file main.cc.

5.4.3.5 const string QUOTE_INSERT = "afegir cita"

Definition at line 22 of file main.cc.

5.4.3.6 const string QUOTE_DELETE = "eliminar cita"

Definition at line 23 of file main.cc.

5.4.3.7 const string QUERY_AUTHORS = "tots autors ?"

Definition at line 25 of file main.cc.

5.4.3.8 `const string QUERY_BOOKS_ALL = "tots textos ?"`

Definition at line 26 of file main.cc.

5.4.3.9 `const string QUERY_BOOKS_BY_AUTHOR = "textos autor"`

Definition at line 27 of file main.cc.

5.4.3.10 `const string QUERY_CURRENT_AUTHOR = "autor ?"`

Definition at line 29 of file main.cc.

5.4.3.11 `const string QUERY_CURRENT_CONTENT = "contingut ?"`

Definition at line 30 of file main.cc.

5.4.3.12 `const string QUERY_CURRENT_INFO = "info ?"`

Definition at line 31 of file main.cc.

5.4.3.13 `const string QUERY_CURRENT_EXPRESION = "frases"`

Definition at line 32 of file main.cc.

5.4.3.14 `const string QUERY_CURRENT_LINES = "nombre de frases ?"`

Definition at line 33 of file main.cc.

5.4.3.15 `const string QUERY_CURRENT_WORDS = "nombre de paraules ?"`

Definition at line 34 of file main.cc.

5.4.3.16 `const string QUERY_CURRENT_FREQUENCY = "taula de frequencies ?"`

Definition at line 35 of file main.cc.

5.4.3.17 `const string QUERY_CURRENT_QUOTES = "cites ?"`

Definition at line 36 of file main.cc.

5.4.3.18 `const string QUERY_QUOTES_ALL = "totes cites ?"`

Definition at line 38 of file main.cc.

5.4.3.19 `const string QUERY_QUOTES_BY_AUTHOR = "cites autor"`

Definition at line 39 of file main.cc.

5.4.3.20 `const string QUERY_QUOTE_INFO = "info cita"`

Definition at line 40 of file main.cc.

5.4.3.21 `const string QUIT = "sortir"`

Definition at line 42 of file main.cc.

5.5 Quote.hh File Reference

Data model that hosts information about a [Quote](#).

Classes

- class [Quote](#)

Data model for Quotes (cites) from Books.

5.5.1 Detailed Description

Data model that hosts information about a [Quote](#).

Definition in file [Quote.hh](#).

Index

Author, [7](#)
 Author, [7](#)

Book, [7](#)
 Book, [8](#)

Library, [8](#)
 Library, [9](#)

Quote, [9](#)
 Quote, [10](#)