

Bookshop

Generated by Doxygen 1.8.13

Contents

1	Namespace Index	1
1.1	Packages	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Namespace Documentation	9
5.1	Package client	9
5.2	Package client.gui	9
5.3	Package db	9
5.4	Package server	10
5.5	Package server.data	10
5.6	Package server.remote	10

6	Class Documentation	11
6.1	client.gui.AddBooks Class Reference	11
6.1.1	Constructor & Destructor Documentation	11
6.1.1.1	AddBooks()	11
6.1.2	Member Function Documentation	11
6.1.2.1	main()	11
6.2	server.data.Book Class Reference	12
6.2.1	Constructor & Destructor Documentation	12
6.2.1.1	Book() [1/2]	12
6.2.1.2	Book() [2/2]	13
6.2.2	Member Function Documentation	13
6.2.2.1	addReview()	13
6.2.2.2	addUser()	13
6.2.2.3	getAuthor()	13
6.2.2.4	getCategory()	13
6.2.2.5	getDescription()	13
6.2.2.6	getEdition()	14
6.2.2.7	getImg()	14
6.2.2.8	getISBN()	14
6.2.2.9	getPrice()	14
6.2.2.10	getReviews()	14
6.2.2.11	getTitle()	14
6.2.2.12	getUsers()	14
6.2.2.13	removeReview()	14
6.2.2.14	removeUser()	15
6.2.2.15	setAuthor()	15
6.2.2.16	setCategory()	15
6.2.2.17	setDescription()	15
6.2.2.18	setEdition()	15
6.2.2.19	setImg()	15

6.2.2.20	setISBN()	15
6.2.2.21	setPrice()	16
6.2.2.22	setTitle()	16
6.2.2.23	toString()	16
6.3	client.Client Class Reference	16
6.3.1	Member Function Documentation	16
6.3.1.1	buyBook()	17
6.3.1.2	displayMenu()	17
6.3.1.3	displaySubMenu()	17
6.3.1.4	login()	17
6.3.1.5	main()	17
6.3.1.6	mainMenu()	17
6.3.1.7	menuBook()	17
6.3.1.8	menuReviews()	18
6.3.1.9	menuSearch()	18
6.3.1.10	menuShowBooks()	18
6.3.1.11	searchISBN()	18
6.3.1.12	searchTitle()	18
6.3.1.13	showBook()	18
6.3.1.14	showBooks()	18
6.3.1.15	signup()	19
6.4	server.ContiPerfTest Class Reference	19
6.4.1	Member Function Documentation	19
6.4.1.1	registerNewUserTest()	19
6.4.1.2	showBookThreads()	19
6.4.1.3	showBookThreads2()	20
6.4.1.4	storeBook()	20
6.4.1.5	storeBookDuration()	20
6.4.1.6	suite()	20
6.4.2	Member Data Documentation	20

6.4.2.1	rule	20
6.5	db.DAO Class Reference	20
6.5.1	Constructor & Destructor Documentation	21
6.5.1.1	DAO()	21
6.5.2	Member Function Documentation	21
6.5.2.1	deleteBook()	21
6.5.2.2	deleteReview()	21
6.5.2.3	getAllBooks()	22
6.5.2.4	getAllReviews()	22
6.5.2.5	getAllUsers()	22
6.5.2.6	retrieveBook()	22
6.5.2.7	retrieveBookByParameter()	23
6.5.2.8	retrieveReview()	23
6.5.2.9	retrieveUser()	24
6.5.2.10	storeBook()	24
6.5.2.11	storeUser()	24
6.5.2.12	updateBook()	25
6.5.2.13	updateReview()	25
6.5.2.14	updateUser()	25
6.6	server.DAOMockTest Class Reference	26
6.6.1	Member Function Documentation	26
6.6.1.1	setUp()	26
6.6.1.2	suite()	26
6.6.1.3	testAddBookInvalidRemote()	27
6.6.1.4	testAddBookValid()	27
6.6.1.5	testRegisterUserAlreadyExists()	27
6.6.1.6	testRegisterUserCorrectly()	27
6.7	db.DB Class Reference	27
6.7.1	Constructor & Destructor Documentation	28
6.7.1.1	DB() [1/2]	28

6.7.1.2	DB() [2/2]	28
6.7.2	Member Function Documentation	28
6.7.2.1	addBookToDb()	28
6.7.2.2	addReview()	29
6.7.2.3	averageRatingByBook()	29
6.7.2.4	averageRatingByUser()	30
6.7.2.5	buyBook()	30
6.7.2.6	deleteBook()	31
6.7.2.7	deleteReview()	31
6.7.2.8	getAllBooks()	31
6.7.2.9	getAllReviews()	32
6.7.2.10	getAllUsers()	32
6.7.2.11	getBookReviews()	32
6.7.2.12	getUserReviews()	33
6.7.2.13	registerUser() [1/2]	33
6.7.2.14	registerUser() [2/2]	33
6.7.2.15	showBookByISBN()	34
6.7.2.16	showBookByTitle()	34
6.7.2.17	showReview()	34
6.7.2.18	showUser()	35
6.8	db.IDAO Interface Reference	35
6.8.1	Member Function Documentation	36
6.8.1.1	deleteBook()	36
6.8.1.2	deleteReview()	36
6.8.1.3	getAllBooks()	36
6.8.1.4	getAllReviews()	37
6.8.1.5	getAllUsers()	37
6.8.1.6	retrieveBook()	37
6.8.1.7	retrieveBookByParameter()	38
6.8.1.8	retrieveReview()	38

6.8.1.9	retrieveUser()	38
6.8.1.10	storeBook()	39
6.8.1.11	storeUser()	39
6.8.1.12	updateBook()	39
6.8.1.13	updateReview()	40
6.8.1.14	updateUser()	40
6.9	db.IDB Interface Reference	41
6.9.1	Member Function Documentation	41
6.9.1.1	addBookToDb()	41
6.9.1.2	addReview()	42
6.9.1.3	averageRatingByBook()	42
6.9.1.4	averageRatingByUser()	42
6.9.1.5	buyBook()	43
6.9.1.6	deleteBook()	43
6.9.1.7	deleteReview()	44
6.9.1.8	getAllBooks()	44
6.9.1.9	getAllReviews()	44
6.9.1.10	getAllUsers()	45
6.9.1.11	getBookReviews()	45
6.9.1.12	getUserReviews()	45
6.9.1.13	registerUser()	46
6.9.1.14	showBookByISBN()	46
6.9.1.15	showBookByTitle()	46
6.9.1.16	showReview()	47
6.9.1.17	showUser()	47
6.10	server.remote.IRemote Interface Reference	48
6.10.1	Member Function Documentation	48
6.10.1.1	addBook() [1/2]	48
6.10.1.2	addBook() [2/2]	49
6.10.1.3	addReview()	50

6.10.1.4	averageRatingByBook()	50
6.10.1.5	averageRatingByUser()	51
6.10.1.6	bookTest()	51
6.10.1.7	buyBook()	51
6.10.1.8	deleteBook()	52
6.10.1.9	deleteReview()	52
6.10.1.10	getAllReviews()	53
6.10.1.11	getAllUsers()	53
6.10.1.12	getBookByISBN()	53
6.10.1.13	getBookByTitle()	54
6.10.1.14	getBookReviews()	54
6.10.1.15	getReview()	55
6.10.1.16	getUser()	55
6.10.1.17	getUserReviews()	56
6.10.1.18	registerUser()	56
6.10.1.19	showBooksInStore()	57
6.11	client.gui.LogIn Class Reference	57
6.11.1	Constructor & Destructor Documentation	58
6.11.1.1	LogIn()	58
6.11.2	Member Function Documentation	58
6.11.2.1	main()	58
6.12	server.remote.Remote Class Reference	58
6.12.1	Constructor & Destructor Documentation	59
6.12.1.1	Remote()	59
6.12.2	Member Function Documentation	59
6.12.2.1	addBook() [1/2]	59
6.12.2.2	addBook() [2/2]	59
6.12.2.3	addReview()	60
6.12.2.4	averageRatingByBook()	60
6.12.2.5	averageRatingByUser()	61

6.12.2.6	bookTest()	61
6.12.2.7	buyBook()	62
6.12.2.8	deleteBook()	62
6.12.2.9	deleteReview()	62
6.12.2.10	getAllReviews()	63
6.12.2.11	getAllUsers()	63
6.12.2.12	getBookByISBN()	64
6.12.2.13	getBookByTitle()	64
6.12.2.14	getBookReviews()	65
6.12.2.15	getReview()	65
6.12.2.16	getUser()	65
6.12.2.17	getUserReviews()	66
6.12.2.18	registerUser()	66
6.12.2.19	showBooksInStore()	67
6.13	server.data.Review Class Reference	67
6.13.1	Constructor & Destructor Documentation	68
6.13.1.1	Review() [1/3]	68
6.13.1.2	Review() [2/3]	68
6.13.1.3	Review() [3/3]	68
6.13.2	Member Function Documentation	68
6.13.2.1	getBook()	69
6.13.2.2	getComment()	69
6.13.2.3	getId_review()	69
6.13.2.4	getRating()	69
6.13.2.5	getUser()	69
6.13.2.6	setBook()	69
6.13.2.7	setComment()	69
6.13.2.8	setId_review()	70
6.13.2.9	setRating()	70
6.13.2.10	setUser()	70

6.13.2.11 toString()	70
6.14 server.RMCTest Class Reference	70
6.14.1 Detailed Description	71
6.14.2 Member Function Documentation	71
6.14.2.1 bookTestValidation()	71
6.14.2.2 getThings()	71
6.14.2.3 registerExistingUserTest()	71
6.14.2.4 registerNewUserTest()	71
6.14.2.5 setUp()	71
6.14.2.6 setUpClient()	71
6.14.2.7 showBooksInStoreTest()	72
6.14.2.8 showUserFailTest()	72
6.14.2.9 showUsersTest()	72
6.14.2.10 suite()	72
6.14.2.11 tearDown()	72
6.15 server.Server Class Reference	73
6.15.1 Member Function Documentation	73
6.15.1.1 main()	73
6.16 client.gui.ShowBooks Class Reference	73
6.16.1 Constructor & Destructor Documentation	73
6.16.1.1 ShowBooks()	73
6.16.2 Member Function Documentation	73
6.16.2.1 main()	74
6.17 client.gui.ShowBooksAdmin Class Reference	74
6.17.1 Constructor & Destructor Documentation	74
6.17.1.1 ShowBooksAdmin() [1/2]	74
6.17.1.2 ShowBooksAdmin() [2/2]	74
6.17.2 Member Function Documentation	74
6.17.2.1 main()	75
6.18 client.gui.ShowDescription Class Reference	75

6.18.1	Constructor & Destructor Documentation	75
6.18.1.1	ShowDescription()	75
6.19	client.gui.ShowDescriptionAdmin Class Reference	75
6.19.1	Constructor & Destructor Documentation	76
6.19.1.1	ShowDescriptionAdmin()	76
6.19.2	Member Function Documentation	76
6.19.2.1	actionPerformed()	76
6.19.2.2	main()	76
6.20	server.data.User Class Reference	77
6.20.1	Constructor & Destructor Documentation	77
6.20.1.1	User() [1/2]	77
6.20.1.2	User() [2/2]	78
6.20.2	Member Function Documentation	78
6.20.2.1	addBook()	78
6.20.2.2	addReview()	78
6.20.2.3	getAddress()	78
6.20.2.4	getBooks()	78
6.20.2.5	getEmail()	78
6.20.2.6	getMoney()	79
6.20.2.7	getName()	79
6.20.2.8	getPassword()	79
6.20.2.9	getReviews()	79
6.20.2.10	getRole()	79
6.20.2.11	removeBook()	79
6.20.2.12	removeReview()	79
6.20.2.13	setAddress()	80
6.20.2.14	setEmail()	80
6.20.2.15	setMoney()	80
6.20.2.16	setName()	80
6.20.2.17	setPassword()	80
6.20.2.18	setRole()	80
6.20.2.19	toString()	80

7 File Documentation	81
7.1 src/main/java/client/Client.java File Reference	81
7.2 src/main/java/client/gui/AddBooks.java File Reference	81
7.3 src/main/java/client/gui/LogIn.java File Reference	81
7.4 src/main/java/client/gui/ShowBooks.java File Reference	82
7.5 src/main/java/client/gui/ShowBooksAdmin.java File Reference	82
7.6 src/main/java/client/gui/ShowDescription.java File Reference	82
7.7 src/main/java/client/gui/ShowDescriptionAdmin.java File Reference	82
7.8 src/main/java/db/DAO.java File Reference	83
7.9 src/main/java/db/DB.java File Reference	83
7.10 src/main/java/db/IDAO.java File Reference	83
7.11 src/main/java/db/IDB.java File Reference	83
7.12 src/main/java/server/data/Book.java File Reference	84
7.13 src/main/java/server/data/Review.java File Reference	84
7.14 src/main/java/server/data/User.java File Reference	84
7.15 src/main/java/server/remote/IRemote.java File Reference	84
7.16 src/main/java/server/remote/Remote.java File Reference	85
7.17 src/main/java/server/Server.java File Reference	85
7.18 src/test/java/server/ContiPerfTest.java File Reference	85
7.19 src/test/java/server/DAOMockTest.java File Reference	85
7.20 src/test/java/server/RMITest.java File Reference	85
Index	87

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

client	9
client.gui	9
db	9
server	10
server.data	10
server.remote	10

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

client.gui.AddBooks	11
client.Client	16
server.ContiPerfTest	19
server.DAOMockTest	26
db.IDAO	35
db.DAO	20
db.IDB	41
db.DB	27
server.remote.IRemote	48
server.remote.Remote	58
client.gui.LogIn	57
server.RMITest	70
server.Server	73
client.gui.ShowBooks	73
client.gui.ShowBooksAdmin	74
client.gui.ShowDescription	75
ActionListener	
client.gui.ShowDescriptionAdmin	75
Serializable	
server.data.Book	12
server.data.Review	67
server.data.User	77
UnicastRemoteObject	
server.remote.Remote	58

Chapter 3

Class Index

3.1 Class List

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

client.gui.AddBooks	11
server.data.Book	12
client.Client	16
server.ContiPerfTest	19
db.DAO	20
server.DAOMockTest	26
db.DB	27
db.IDAO	35
db.IDB	41
server.remote.IRemote	48
client.gui.LogIn	57
server.remote.Remote	58
server.data.Review	67
server.RMITest	70
server.Server	73
client.gui.ShowBooks	73
client.gui.ShowBooksAdmin	74
client.gui.ShowDescription	75
client.gui.ShowDescriptionAdmin	75
server.data.User	77

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

src/main/java/client/ Client.java	81
src/main/java/client/gui/ AddBooks.java	81
src/main/java/client/gui/ LogIn.java	81
src/main/java/client/gui/ ShowBooks.java	82
src/main/java/client/gui/ ShowBooksAdmin.java	82
src/main/java/client/gui/ ShowDescription.java	82
src/main/java/client/gui/ ShowDescriptionAdmin.java	82
src/main/java/db/ DAO.java	83
src/main/java/db/ DB.java	83
src/main/java/db/ IDAO.java	83
src/main/java/db/ IDB.java	83
src/main/java/server/ Server.java	85
src/main/java/server/data/ Book.java	84
src/main/java/server/data/ Review.java	84
src/main/java/server/data/ User.java	84
src/main/java/server/remote/ IRemote.java	84
src/main/java/server/remote/ Remote.java	85
src/test/java/server/ ContiPerfTest.java	85
src/test/java/server/ DAOMockTest.java	85
src/test/java/server/ RMITest.java	85

Chapter 5

Namespace Documentation

5.1 Package client

Packages

- package [gui](#)

Classes

- class [Client](#)

5.2 Package client.gui

Classes

- class [AddBooks](#)
- class **BookTableModel**
- class [Login](#)
- class **ReviewTableModel**
- class [ShowBooks](#)
- class [ShowBooksAdmin](#)
- class [ShowDescription](#)
- class [ShowDescriptionAdmin](#)

5.3 Package db

Classes

- class [DAO](#)
- class [DB](#)
- interface [IDAO](#)
- interface [IDB](#)

5.4 Package server

Packages

- package [data](#)
- package [remote](#)

Classes

- class [ContiPerfTest](#)
- class [DAOMockTest](#)
- class [RMITest](#)
- class [Server](#)

5.5 Package server.data

Classes

- class [Book](#)
- class [Review](#)
- class [User](#)

5.6 Package server.remote

Classes

- interface [IRemote](#)
- class [Remote](#)

Chapter 6

Class Documentation

6.1 client.gui.AddBooks Class Reference

Public Member Functions

- [AddBooks](#) (String email)

Static Public Member Functions

- static void [main](#) (String[] args)

6.1.1 Constructor & Destructor Documentation

6.1.1.1 AddBooks()

```
client.gui.AddBooks.AddBooks (
    String email )
```

Create the application.

6.1.2 Member Function Documentation

6.1.2.1 main()

```
static void client.gui.AddBooks.main (
    String [] args ) [static]
```

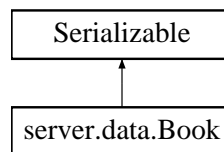
Launch the application.

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

- src/main/java/client/gui/[AddBooks.java](#)

6.2 server.data.Book Class Reference

Inheritance diagram for server.data.Book:



Public Member Functions

- [Book](#) (int ISBN, String title, String category, String edition, String author, double price, String description, String img)
- [Book](#) (int ISBN, String title, String author, double price)
- void [addReview](#) ([Review](#) review)
- void [removeReview](#) ([Review](#) review)
- List< [Review](#) > [getReviews](#) ()
- void [addUser](#) ([User](#) user)
- void [removeUser](#) ([User](#) user)
- List< [User](#) > [getUsers](#) ()
- int [getISBN](#) ()
- void [setISBN](#) (int ISBN)
- String [getTitle](#) ()
- void [setTitle](#) (String title)
- String [getCategory](#) ()
- void [setCategory](#) (String category)
- String [getEdition](#) ()
- void [setEdition](#) (String edition)
- String [getAuthor](#) ()
- void [setAuthor](#) (String author)
- double [getPrice](#) ()
- void [setPrice](#) (double price)
- String [getDescription](#) ()
- void [setDescription](#) (String description)
- String [getImg](#) ()
- void [setImg](#) (String img)
- String [toString](#) ()

6.2.1 Constructor & Destructor Documentation

6.2.1.1 [Book\(\)](#) [1/2]

```

server.data.Book.Book (
    int ISBN,
    String title,
    String category,
    String edition,
    String author,
    double price,
    String description,
    String img )
  
```

6.2.1.2 Book() [2/2]

```
server.data.Book.Book (
    int ISBN,
    String title,
    String author,
    double price )
```

6.2.2 Member Function Documentation

6.2.2.1 addReview()

```
void server.data.Book.addReview (
    Review review )
```

6.2.2.2 addUser()

```
void server.data.Book.addUser (
    User user )
```

6.2.2.3 getAuthor()

```
String server.data.Book.getAuthor ( )
```

6.2.2.4 getCategory()

```
String server.data.Book.getCategory ( )
```

6.2.2.5 getDescription()

```
String server.data.Book.getDescription ( )
```

6.2.2.6 getEdition()

```
String server.data.Book.getEdition ( )
```

6.2.2.7 getImg()

```
String server.data.Book.getImg ( )
```

6.2.2.8 getISBN()

```
int server.data.Book.getISBN ( )
```

6.2.2.9 getPrice()

```
double server.data.Book.getPrice ( )
```

6.2.2.10 getReviews()

```
List<Review> server.data.Book.getReviews ( )
```

6.2.2.11 getTitle()

```
String server.data.Book.getTitle ( )
```

6.2.2.12 getUsers()

```
List<User> server.data.Book.getUsers ( )
```

6.2.2.13 removeReview()

```
void server.data.Book.removeReview (
    Review review )
```

6.2.2.14 removeUser()

```
void server.data.Book.removeUser (
    User user )
```

6.2.2.15 setAuthor()

```
void server.data.Book.setAuthor (
    String author )
```

6.2.2.16 setCategory()

```
void server.data.Book.setCategory (
    String category )
```

6.2.2.17 setDescription()

```
void server.data.Book.setDescription (
    String description )
```

6.2.2.18 setEdition()

```
void server.data.Book.setEdition (
    String edition )
```

6.2.2.19 setImg()

```
void server.data.Book.setImg (
    String img )
```

6.2.2.20 setISBN()

```
void server.data.Book.setISBN (
    int ISBN )
```

6.2.2.21 setPrice()

```
void server.data.Book.setPrice (
    double price )
```

6.2.2.22 setTitle()

```
void server.data.Book.setTitle (
    String title )
```

6.2.2.23 toString()

```
String server.data.Book.toString ( )
```

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

- [src/main/java/server/data/Book.java](#)

6.3 client.Client Class Reference

Static Public Member Functions

- static void [displayMenu](#) (String[] options)
- static void [displaySubMenu](#) (String[] options)
- static void [mainMenu](#) (IRemote server)
- static void [menuShowBooks](#) (IRemote server)
- static void [menuSearch](#) (IRemote server)
- static void [menuBook](#) (IRemote server)
- static void [menuReviews](#) (IRemote server)
- static void [login](#) (IRemote server)
- static void [signUp](#) (IRemote server)
- static void [searchISBN](#) (IRemote server)
- static void [searchTitle](#) (IRemote server)
- static void [showBooks](#) (IRemote server)
- static void [showBook](#) (IRemote server, int numBook)
- static void [buyBook](#) (IRemote server)
- static void [main](#) (String[] args)

6.3.1 Member Function Documentation

6.3.1.1 buyBook()

```
static void client.Client.buyBook (
    IRemote server ) [static]
```

6.3.1.2 displayMenu()

```
static void client.Client.displayMenu (
    String [] options ) [static]
```

6.3.1.3 displaySubMenu()

```
static void client.Client.displaySubMenu (
    String [] options ) [static]
```

6.3.1.4 login()

```
static void client.Client.logIn (
    IRemote server ) [static]
```

6.3.1.5 main()

```
static void client.Client.main (
    String [] args ) [static]
```

6.3.1.6 mainMenu()

```
static void client.Client.mainMenu (
    IRemote server ) [static]
```

6.3.1.7 menuBook()

```
static void client.Client.menuBook (
    IRemote server ) [static]
```

6.3.1.8 menuReviews()

```
static void client.Client.menuReviews (
    IRemote server ) [static]
```

6.3.1.9 menuSearch()

```
static void client.Client.menuSearch (
    IRemote server ) [static]
```

6.3.1.10 menuShowBooks()

```
static void client.Client.menuShowBooks (
    IRemote server ) [static]
```

6.3.1.11 searchISBN()

```
static void client.Client.searchISBN (
    IRemote server ) [static]
```

6.3.1.12 searchTitle()

```
static void client.Client.searchTitle (
    IRemote server ) [static]
```

6.3.1.13 showBook()

```
static void client.Client.showBook (
    IRemote server,
    int numBook ) [static]
```

6.3.1.14 showBooks()

```
static void client.Client.showBooks (
    IRemote server ) [static]
```


6.3.1.15 signUp()

```
static void client.Client.signUp (
    IRemote server ) [static]
```

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

- src/main/java/client/[Client.java](#)

6.4 server.ContiPerfTest Class Reference

Public Member Functions

- void [storeBook](#) ()
- void [registerNewUserTest](#) ()
- void [storeBookDuration](#) ()
- void [showBookThreads](#) ()
- void [showBookThreads2](#) ()

Static Public Member Functions

- static junit.framework.Test [suite](#) ()

Public Attributes

- ContiPerfRule [rule](#) = new ContiPerfRule()

6.4.1 Member Function Documentation

6.4.1.1 registerNewUserTest()

```
void server.ContiPerfTest.registerNewUserTest ( )
```

6.4.1.2 showBookThreads()

```
void server.ContiPerfTest.showBookThreads ( )
```

6.4.1.3 showBookThreads2()

```
void server.ContiPerfTest.showBookThreads2 ( )
```

6.4.1.4 storeBook()

```
void server.ContiPerfTest.storeBook ( )
```

6.4.1.5 storeBookDuration()

```
void server.ContiPerfTest.storeBookDuration ( )
```

6.4.1.6 suite()

```
static junit.framework.Test server.ContiPerfTest.suite ( ) [static]
```

6.4.2 Member Data Documentation

6.4.2.1 rule

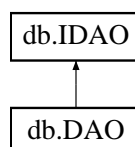
```
ContiPerfRule server.ContiPerfTest.rule = new ContiPerfRule()
```

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

- [src/test/java/server/ContiPerfTest.java](#)

6.5 db.DAO Class Reference

Inheritance diagram for db.DAO:



Public Member Functions

- [DAO](#) ()
- boolean [storeUser](#) ([User](#) u)
- [User](#) [retrieveUser](#) (String email)
- boolean [updateUser](#) ([User](#) u)
- [Review](#) [retrieveReview](#) (int id_review)
- boolean [updateReview](#) ([Review](#) review)
- boolean [storeBook](#) ([Book](#) b)
- [Book](#) [retrieveBook](#) (int ISBN)
- boolean [updateBook](#) ([Book](#) b)
- [Book](#) [retrieveBookByParameter](#) (String title)
- List< [User](#) > [getAllUsers](#) ()
- List< [Review](#) > [getAllReviews](#) ()
- List< [Book](#) > [getAllBooks](#) ()
- void [deleteReview](#) ([Review](#) r)
- void [deleteBook](#) ([Book](#) b)

6.5.1 Constructor & Destructor Documentation

6.5.1.1 DAO()

```
db.DAO.DAO ( )
```

6.5.2 Member Function Documentation

6.5.2.1 deleteBook()

```
void db.DAO.deleteBook (
    Book b )
```

Eliminates a book from the database

Parameters

<i>b</i>	the book to eliminate
----------	-----------------------

Implements [db.IDAO](#).

6.5.2.2 deleteReview()

```
void db.DAO.deleteReview (
    Review r )
```

Eliminates a Review from the database

Parameters

<i>r</i>	the review to eliminate
----------	-------------------------

Implements [db.IDAO](#).

6.5.2.3 getAllBooks()

```
List<Book> db.DAO.getAllBooks ( )
```

get all the Books of the Database

Returns

A List<Book> of all users

Implements [db.IDAO](#).

6.5.2.4 getAllReviews()

```
List<Review> db.DAO.getAllReviews ( )
```

get all the Reviews of the Database

Returns

A List<Review> of all users

Implements [db.IDAO](#).

6.5.2.5 getAllUsers()

```
List<User> db.DAO.getAllUsers ( )
```

get all the Users of the Database

Returns

A List<User> of all users

Implements [db.IDAO](#).

6.5.2.6 retrieveBook()

```
Book db.DAO.retrieveBook (
    int ISBN )
```

Get a Book from the database by its ISBN

Parameters

<i>ISBN</i>	
-------------	--

Returns

The Book

Implements [db.IDAO](#).

6.5.2.7 retrieveBookByParameter()

```
Book db.DAO.retrieveBookByParameter (
    String title )
```

Get a Book from the database by its title

Parameters

<i>title</i>	of the book
--------------	-------------

Returns

The book

Implements [db.IDAO](#).

6.5.2.8 retrieveReview()

```
Review db.DAO.retrieveReview (
    int id_review )
```

Get a Review from the database by its ID

Parameters

<i>id_review</i>	
------------------	--

Returns

The Review

Implements [db.IDAO](#).

6.5.2.9 retrieveUser()

```
User db.DAO.retrieveUser (
    String email )
```

Get a User form the database by its email

Parameters

<i>email</i>	Email of the user you want to get
--------------	-----------------------------------

Returns

The User

Implements [db.IDAO](#).

6.5.2.10 storeBook()

```
boolean db.DAO.storeBook (
    Book b )
```

Store a Book in the database

Parameters

<i>b</i>	the Book you want to store
----------	----------------------------

Returns

true of false to tell if it worked

Implements [db.IDAO](#).

6.5.2.11 storeUser()

```
boolean db.DAO.storeUser (
    User u )
```

Store a user in the dataBase

Parameters

<i>u</i>	User you want to store
----------	------------------------

Returns

true of false to tell if it worked

Implements [db.IDAO](#).

6.5.2.12 updateBook()

```
boolean db.DAO.updateBook (
    Book b )
```

Updates the row in the database of a given Book

Parameters

<i>b</i>	the book
----------	----------

Returns

true of false to tell if it worked

Implements [db.IDAO](#).

6.5.2.13 updateReview()

```
boolean db.DAO.updateReview (
    Review r )
```

Updates the row in the database of a given Review

Parameters

<i>r</i>	the review to update
----------	----------------------

Returns

true of false to tell if it worked

Implements [db.IDAO](#).

6.5.2.14 updateUser()

```
boolean db.DAO.updateUser (
    User u )
```

Updates the row in the database of a given User

Parameters

<i>u</i>	User you are updating
----------	-----------------------

Returns

true of false to tell if it worked

Implements [db.IDAO](#).

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

- [src/main/java/db/DAO.java](#)

6.6 server.DAOMockTest Class Reference

Public Member Functions

- void [setUp](#) () throws Exception
- void [testRegisterUserCorrectly](#) ()
- void [testRegisterUserAlreadyExists](#) ()
- void [testAddBookValid](#) () throws RemoteException
- void [testAddBookInvalidRemote](#) () throws RemoteException

Static Public Member Functions

- static junit.framework.Test [suite](#) ()

6.6.1 Member Function Documentation

6.6.1.1 [setUp\(\)](#)

```
void server.DAOMockTest.setUp ( ) throws Exception
```

6.6.1.2 [suite\(\)](#)

```
static junit.framework.Test server.DAOMockTest.suite ( ) [static]
```


6.6.1.3 testAddBookInvalidRemote()

```
void server.DAOMockTest.testAddBookInvalidRemote ( ) throws RemoteException
```

6.6.1.4 testAddBookValid()

```
void server.DAOMockTest.testAddBookValid ( ) throws RemoteException
```

6.6.1.5 testRegisterUserAlreadyExists()

```
void server.DAOMockTest.testRegisterUserAlreadyExists ( )
```

6.6.1.6 testRegisterUserCorrectly()

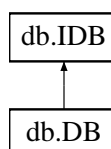
```
void server.DAOMockTest.testRegisterUserCorrectly ( )
```

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

- [src/test/java/server/DAOMockTest.java](#)

6.7 db.DB Class Reference

Inheritance diagram for db.DB:



Public Member Functions

- [DB](#) ()
- [DB](#) ([IDAO](#) *udao*)
- List< [User](#) > [getAllUsers](#) ()
- List< [Review](#) > [getAllReviews](#) ()
- List< [Book](#) > [getAllBooks](#) ()
- boolean [buyBook](#) (String email, String book_title)
- boolean [addReview](#) ([Book](#) b, [Review](#) r, [User](#) u)
- boolean [registerUser](#) (String email, String password, boolean role)
- boolean [registerUser](#) ([User](#) u)
- boolean [addBookToDb](#) ([Book](#) b)
- [Book](#) [showBookByISBN](#) (int ISBN)
- [Book](#) [showBookByTitle](#) (String title)
- [Review](#) [showReview](#) (int id_review)
- [User](#) [showUser](#) (String email)
- List< [Review](#) > [getUserReviews](#) (String email)
- List< [Review](#) > [getBookReviews](#) (String title)
- double [averageRatingByBook](#) (String title)
- double [averageRatingByUser](#) (String email)
- boolean [deleteReview](#) (int id_review)
- boolean [deleteBook](#) (int ISBN)

6.7.1 Constructor & Destructor Documentation

6.7.1.1 [DB](#)() [1/2]

```
db.DB.DB ( )
```

6.7.1.2 [DB](#)() [2/2]

```
db.DB.DB (
    IDAO udao )
```

6.7.2 Member Function Documentation

6.7.2.1 [addBookToDb](#)()

```
boolean db.DB.addBookToDb (
    Book b )
```

Adds a new book to the database

Parameters

<i>b</i>	the book to add
----------	-----------------

Returns

true or false to tell if it worked

Implements [db.IDB](#).

6.7.2.2 addReview()

```
boolean db.DB.addReview (
    Book b,
    Review r,
    User user )
```

Adds a new Review to the database

Parameters

<i>b</i>	the book its from
<i>r</i>	the review
<i>user</i>	the user that wrote it

Returns

true or false to tell if it worked

Implements [db.IDB](#).

6.7.2.3 averageRatingByBook()

```
double db.DB.averageRatingByBook (
    String title )
```

Gets the average rating of a book

Parameters

<i>title</i>	of the book
--------------	-------------

Returns

a double the average rating

Implements [db.IDB](#).

6.7.2.4 averageRatingByUser()

```
double db.DB.averageRatingByUser (
    String email )
```

Gets the average rating given by a user

Parameters

<i>email</i>	of the user
--------------	-------------

Returns

a double the average rating

Implements [db.IDB](#).

6.7.2.5 buyBook()

```
boolean db.DB.buyBook (
    String email,
    String book_title )
```

Allow the User to buy a selected book

Parameters

<i>email</i>	Email of the buying user
<i>book_title</i>	the title of the book to buy

Returns

true or false to tell if it worked

Implements [db.IDB](#).

6.7.2.6 deleteBook()

```
boolean db.DB.deleteBook (
    int ISBN )
```

Eliminates a book from the database

Parameters

<i>ISBN</i>	
-------------	--

Returns

true or false to tell if it worked

Implements [db.IDB](#).

6.7.2.7 deleteReview()

```
boolean db.DB.deleteReview (
    int id_review )
```

Eliminates a Review from the database

Parameters

<i>id_review</i>	
------------------	--

Returns

true or false to tell if it worked

Implements [db.IDB](#).

6.7.2.8 getAllBooks()

```
List<Book> db.DB.getAllBooks ( )
```

get all the Books of the Database

Returns

A List<Book> of all users

Implements [db.IDB](#).

6.7.2.9 getAllReviews()

```
List<Review> db.DB.getAllReviews ( )
```

get all the Reviews of the Database

Returns

A List<Review> of all users

Implements [db.IDB](#).

6.7.2.10 getAllUsers()

```
List<User> db.DB.getAllUsers ( )
```

get all the Users of the Database

Returns

A List<User> of all users

Implements [db.IDB](#).

6.7.2.11 getBookReviews()

```
List<Review> db.DB.getBookReviews (
    String title )
```

Get all the reviews of a book

Parameters

<i>title</i>	of the book
--------------	-------------

Returns

A List<Review> of a given book

Implements [db.IDB](#).

6.7.2.12 `getUserReviews()`

```
List<Review> db.DB.getUserReviews (
    String email )
```

Get all the reviews a User wrote

Parameters

<i>email</i>	of the user
--------------	-------------

Returns

A List<Review> of a given user

Implements [db.IDB](#).

6.7.2.13 `registerUser()` [1/2]

```
boolean db.DB.registerUser (
    String email,
    String password,
    boolean role )
```

creates a new user

Parameters

<i>email</i>	
<i>password</i>	
<i>role</i>	(true admin; false user)

Returns

true or false to tell if it worked

Implements [db.IDB](#).

6.7.2.14 `registerUser()` [2/2]

```
boolean db.DB.registerUser (
    User u )
```

6.7.2.15 showBookByISBN()

```
Book db.DB.showBookByISBN (
    int ISBN )
```

Get a book from the database by its ISBN

Parameters

<i>ISBN</i>	
-------------	--

Returns

The Book

Implements [db.IDB](#).

6.7.2.16 showBookByTitle()

```
Book db.DB.showBookByTitle (
    String title )
```

Get a Book from the database by its title

Parameters

<i>title</i>	of the book
--------------	-------------

Returns

The book

Implements [db.IDB](#).

6.7.2.17 showReview()

```
Review db.DB.showReview (
    int id_review )
```

Get a Review from the database by its ID

Parameters

<i>id_review</i>	
------------------	--

Returns

The Review

Implements [db.IDB](#).

6.7.2.18 showUser()

```
User db.DB.showUser (
    String email )
```

Get a User form the database by its email

Parameters

<i>email</i>	Email of the user you want to get
--------------	-----------------------------------

Returns

The User

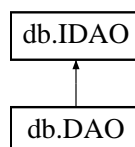
Implements [db.IDB](#).

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

- [src/main/java/db/DB.java](#)

6.8 db.IDAO Interface Reference

Inheritance diagram for db.IDAO:

**Public Member Functions**

- boolean [storeUser](#) ([User](#) u)
- [User](#) [retrieveUser](#) (String email)
- boolean [updateUser](#) ([User](#) u)
- [Review](#) [retrieveReview](#) (int id_review)
- boolean [updateReview](#) ([Review](#) r)
- boolean [storeBook](#) ([Book](#) b)
- [Book](#) [retrieveBook](#) (int ISBN)
- boolean [updateBook](#) ([Book](#) b)
- [Book](#) [retrieveBookByParameter](#) (String title)
- List< [User](#) > [getAllUsers](#) ()
- List< [Review](#) > [getAllReviews](#) ()
- List< [Book](#) > [getAllBooks](#) ()
- void [deleteReview](#) ([Review](#) r)
- void [deleteBook](#) ([Book](#) b)

6.8.1 Member Function Documentation

6.8.1.1 deleteBook()

```
void db.IDAO.deleteBook (
    Book b )
```

Eliminates a book from the database

Parameters

<i>b</i>	the book to eliminate
----------	-----------------------

Implemented in [db.DAO](#).

6.8.1.2 deleteReview()

```
void db.IDAO.deleteReview (
    Review r )
```

Eliminates a Review from the database

Parameters

<i>r</i>	the review to eliminate
----------	-------------------------

Implemented in [db.DAO](#).

6.8.1.3 getAllBooks()

```
List<Book> db.IDAO.getAllBooks ( )
```

get all the Books of the Database

Returns

A List<Book> of all users

Implemented in [db.DAO](#).

6.8.1.4 getAllReviews()

```
List<Review> db.IDAO.getAllReviews ( )
```

get all the Reviews of the Database

Returns

A List<Review> of all users

Implemented in [db.DAO](#).

6.8.1.5 getAllUsers()

```
List<User> db.IDAO.getAllUsers ( )
```

get all the Users of the Database

Returns

A List<User> of all users

Implemented in [db.DAO](#).

6.8.1.6 retrieveBook()

```
Book db.IDAO.retrieveBook (
    int ISBN )
```

Get a Book from the database by its ISBN

Parameters

ISBN	
------	--

Returns

The Book

Implemented in [db.DAO](#).

6.8.1.7 retrieveBookByParameter()

```
Book db.IDAO.retrieveBookByParameter (
    String title )
```

Get a Book from the database by its title

Parameters

<i>title</i>	of the book
--------------	-------------

Returns

The book

Implemented in [db.DAO](#).

6.8.1.8 retrieveReview()

```
Review db.IDAO.retrieveReview (
    int id_review )
```

Get a Review from the database by its ID

Parameters

<i>id_review</i>	
------------------	--

Returns

The Review

Implemented in [db.DAO](#).

6.8.1.9 retrieveUser()

```
User db.IDAO.retrieveUser (
    String email )
```

Get a User form the database by its email

Parameters

<i>email</i>	Email of the user you want to get
--------------	-----------------------------------

Returns

The User

Implemented in [db.DAO](#).

6.8.1.10 storeBook()

```
boolean db.IDAO.storeBook (
    Book b )
```

Store a Book in the database

Parameters

<i>b</i>	the Book you want to store
----------	----------------------------

Returns

true of false to tell if it worked

Implemented in [db.DAO](#).

6.8.1.11 storeUser()

```
boolean db.IDAO.storeUser (
    User u )
```

Store a user in the dataBase

Parameters

<i>u</i>	User you want to store
----------	------------------------

Returns

true of false to tell if it worked

Implemented in [db.DAO](#).

6.8.1.12 updateBook()

```
boolean db.IDAO.updateBook (
    Book b )
```

Updates the row in the database of a given Book

Parameters

<i>b</i>	the book
----------	----------

Returns

true of false to tell if it worked

Implemented in [db.DAO](#).

6.8.1.13 updateReview()

```
boolean db.IDAO.updateReview (
    Review r )
```

Updates the row in the database of a given Review

Parameters

<i>r</i>	the review to update
----------	----------------------

Returns

true of false to tell if it worked

Implemented in [db.DAO](#).

6.8.1.14 updateUser()

```
boolean db.IDAO.updateUser (
    User u )
```

Updates the row in the database of a given User

Parameters

<i>u</i>	User you are updating
----------	-----------------------

Returns

true of false to tell if it worked

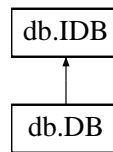
Implemented in [db.DAO](#).

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

- [src/main/java/db/IDAO.java](#)

6.9 db.IDB Interface Reference

Inheritance diagram for db.IDB:



Public Member Functions

- List< [User](#) > [getAllUsers](#) ()
- List< [Review](#) > [getAllReviews](#) ()
- List< [Book](#) > [getAllBooks](#) ()
- boolean [buyBook](#) (String email, String book_title)
- boolean [registerUser](#) (String email, String password, boolean role)
- boolean [addBookToDb](#) ([Book](#) b)
- boolean [addReview](#) ([Book](#) b, [Review](#) r, [User](#) user)
- [Book](#) [showBookByISBN](#) (int ISBN)
- [Book](#) [showBookByTitle](#) (String title)
- [Review](#) [showReview](#) (int id_review)
- [User](#) [showUser](#) (String email)
- List< [Review](#) > [getUserReviews](#) (String email)
- List< [Review](#) > [getBookReviews](#) (String title)
- double [averageRatingByBook](#) (String title)
- double [averageRatingByUser](#) (String email)
- boolean [deleteReview](#) (int id_review)
- boolean [deleteBook](#) (int ISBN)

6.9.1 Member Function Documentation

6.9.1.1 addBookToDb()

```
boolean db.IDB.addBookToDb (
    Book b )
```

Adds a new book to the database

Parameters

<i>b</i>	the book to add
----------	-----------------

Returns

true or false to tell if it worked

Implemented in [db.DB](#).

6.9.1.2 `addReview()`

```
boolean db.IDB.addReview (
    Book b,
    Review r,
    User user )
```

Adds a new Review to the database

Parameters

<i>b</i>	the book its from
<i>r</i>	the review
<i>user</i>	the user that wrote it

Returns

true or false to tell if it worked

Implemented in [db.DB](#).

6.9.1.3 `averageRatingByBook()`

```
double db.IDB.averageRatingByBook (
    String title )
```

Gets the average rating of a book

Parameters

<i>title</i>	of the book
--------------	-------------

Returns

a double the average rating

Implemented in [db.DB](#).

6.9.1.4 `averageRatingByUser()`

```
double db.IDB.averageRatingByUser (
    String email )
```

Gets the average rating given by a user

Parameters

<i>email</i>	of the user
--------------	-------------

Returns

a double the average rating

Implemented in [db.DB](#).

6.9.1.5 buyBook()

```
boolean db.IDB.buyBook (
    String email,
    String book_title )
```

Allow the User to buy a selected book

Parameters

<i>email</i>	Email of the buying user
<i>book_title</i>	the title of the book to buy

Returns

true or false to tell if it worked

Implemented in [db.DB](#).

6.9.1.6 deleteBook()

```
boolean db.IDB.deleteBook (
    int ISBN )
```

Eliminates a book from the database

Parameters

<i>ISBN</i>	
-------------	--

Returns

true or false to tell if it worked

Implemented in [db.DB](#).

6.9.1.7 deleteReview()

```
boolean db.IDB.deleteReview (
    int id_review )
```

Eliminates a Review from the database

Parameters

<i>id_review</i>	
------------------	--

Returns

true or false to tell if it worked

Implemented in [db.DB](#).

6.9.1.8 getAllBooks()

```
List<Book> db.IDB.getAllBooks ( )
```

get all the Books of the Database

Returns

A List<[Book](#)> of all users

Implemented in [db.DB](#).

6.9.1.9 getAllReviews()

```
List<Review> db.IDB.getAllReviews ( )
```

get all the Reviews of the Database

Returns

A List<[Review](#)> of all users

Implemented in [db.DB](#).

6.9.1.10 getAllUsers()

```
List<User> db.IDB.getAllUsers ( )
```

get all the Users of the Database

Returns

A List<User> of all users

Implemented in [db.DB](#).

6.9.1.11 getBookReviews()

```
List<Review> db.IDB.getBookReviews (
    String title )
```

Get all the reviews of a book

Parameters

<i>title</i>	of the book
--------------	-------------

Returns

A List<Review> of a given book

Implemented in [db.DB](#).

6.9.1.12 getUserReviews()

```
List<Review> db.IDB.getUserReviews (
    String email )
```

Get all the reviews a User wrote

Parameters

<i>email</i>	of the user
--------------	-------------

Returns

A List<Review> of a given user

Implemented in [db.DB](#).

6.9.1.13 registerUser()

```
boolean db.IDB.registerUser (
    String email,
    String password,
    boolean role )
```

creates a new user

Parameters

<i>email</i>	
<i>password</i>	
<i>role</i>	(true admin; false user)

Returns

true or false to tell if it worked

Implemented in [db.DB](#).

6.9.1.14 showBookByISBN()

```
Book db.IDB.showBookByISBN (
    int ISBN )
```

Get a book from the database by its ISBN

Parameters

<i>ISBN</i>	
-------------	--

Returns

The Book

Implemented in [db.DB](#).

6.9.1.15 showBookByTitle()

```
Book db.IDB.showBookByTitle (
    String title )
```

Get a Book from the database by its title

Parameters

<i>title</i>	of the book
--------------	-------------

Returns

The book

Implemented in [db.DB](#).

6.9.1.16 showReview()

```
Review db.IDB.showReview (
    int id_review )
```

Get a Review from the database by its ID

Parameters

<i>id_review</i>	
------------------	--

Returns

The Review

Implemented in [db.DB](#).

6.9.1.17 showUser()

```
User db.IDB.showUser (
    String email )
```

Get a User form the database by its email

Parameters

<i>email</i>	Email of the user you want to get
--------------	-----------------------------------

Returns

The User

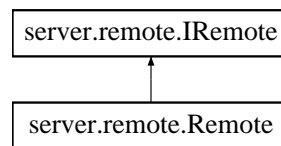
Implemented in [db.DB](#).

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

- [src/main/java/db/IDB.java](#)

6.10 server.remote.IRemote Interface Reference

Inheritance diagram for server.remote.IRemote:



Public Member Functions

- boolean [registerUser](#) (String email, String password, boolean role) throws RemoteException
- boolean [addBook](#) (Book book) throws RemoteException
- List< Book > [showBooksInStore](#) () throws RemoteException
- List< User > [getAllUsers](#) () throws RemoteException
- List< Review > [getAllReviews](#) () throws RemoteException
- Book [bookTest](#) () throws RemoteException
- Book [getBookByISBN](#) (int ISBN) throws RemoteException
- Book [getBookByTitle](#) (String title) throws RemoteException
- Review [getReview](#) (int id_review) throws RemoteException
- User [getUser](#) (String email) throws RemoteException
- boolean [buyBook](#) (String email, String book_title) throws RemoteException
- boolean [addReview](#) (Book b, Review r, User u) throws RemoteException
- List< Review > [getUserReviews](#) (String email) throws RemoteException
- List< Review > [getBookReviews](#) (String title) throws RemoteException
- double [averageRatingByBook](#) (String title) throws RemoteException
- double [averageRatingByUser](#) (String email) throws RemoteException
- void [deleteReview](#) (int id_review) throws RemoteException
- void [deleteBook](#) (int ISBN) throws RemoteException
- boolean [addBook](#) (int ISBN, String title, String category, String edition, String author, double price, String description, String img) throws RemoteException

6.10.1 Member Function Documentation

6.10.1.1 [addBook\(\)](#) [1/2]

```
boolean server.remote.IRemote.addBook (
    Book book ) throws RemoteException
```

adds a book to the database

Parameters

<i>book</i>	
-------------	--

Returns

true or false to tell if it worked

Exceptions

<i>RemoteException</i>	
------------------------	--

Implemented in [server.remote.Remote](#).

6.10.1.2 addBook() [2/2]

```
boolean server.remote.IRemote.addBook (
    int ISBN,
    String title,
    String category,
    String edition,
    String author,
    double price,
    String description,
    String img ) throws RemoteException
```

Creates new book in the database

Parameters

<i>ISBN</i>	unique identifier
<i>title</i>	String of the book
<i>category</i>	genre of book
<i>edition</i>	
<i>author</i>	
<i>price</i>	
<i>description</i>	
<i>img</i>	

Returns

true of false to tell if it worked

Exceptions

<i>RemoteException</i>	
------------------------	--

Implemented in [server.remote.Remote](#).

6.10.1.3 addReview()

```
boolean server.remote.IRemote.addReview (
    Book b,
    Review r,
    User u ) throws RemoteException
```

add a new review to a book

Parameters

<i>b</i>	the book to add a review
<i>r</i>	the Review
<i>u</i>	the User who made the review

Returns

true or false to tell if it worked

Exceptions

<i>RemoteException</i>	
------------------------	--

Implemented in [server.remote.Remote](#).

6.10.1.4 averageRatingByBook()

```
double server.remote.IRemote.averageRatingByBook (
    String title ) throws RemoteException
```

Get the average Rating of a book

Parameters

<i>title</i>	of the book
--------------	-------------

Returns

a double

Exceptions

<i>RemoteException</i>	
------------------------	--

Implemented in [server.remote.Remote](#).

6.10.1.5 averageRatingByUser()

```
double server.remote.IRemote.averageRatingByUser (
    String email ) throws RemoteException
```

Get the average Rating given by a User

Parameters

<i>email</i>	
--------------	--

Returns

a double

Exceptions

<i>RemoteException</i>	
------------------------	--

Implemented in [server.remote.Remote](#).

6.10.1.6 bookTest()

```
Book server.remote.IRemote.bookTest ( ) throws RemoteException
```

Test

Returns

the test book

Exceptions

<i>RemoteException</i>	
------------------------	--

Implemented in [server.remote.Remote](#).

6.10.1.7 buyBook()

```
boolean server.remote.IRemote.buyBook (
    String email,
    String book_title ) throws RemoteException
```

Buy a book

Parameters

<i>email</i>	of the user
<i>book_title</i>	title of the book

Returns

true of false to tell if it worked

Exceptions

<i>RemoteException</i>	
------------------------	--

Implemented in [server.remote.Remote](#).

6.10.1.8 deleteBook()

```
void server.remote.IRemote.deleteBook (
    int ISBN ) throws RemoteException
```

Eliminates a Book

Parameters

<i>ISBN</i>	
-------------	--

Exceptions

<i>RemoteException</i>	
------------------------	--

Implemented in [server.remote.Remote](#).

6.10.1.9 deleteReview()

```
void server.remote.IRemote.deleteReview (
    int id_review ) throws RemoteException
```

Eliminates a Review

Parameters

<i>id_review</i>	
------------------	--

Exceptions

<i>RemoteException</i>	
------------------------	--

Implemented in [server.remote.Remote](#).

6.10.1.10 getAllReviews()

```
List<Review> server.remote.IRemote.getAllReviews ( ) throws RemoteException
```

gives you a all the Review in the database

Returns

a list<Review> with all the books in the database

Exceptions

<i>RemoteException</i>	
------------------------	--

Implemented in [server.remote.Remote](#).

6.10.1.11 getAllUsers()

```
List<User> server.remote.IRemote.getAllUsers ( ) throws RemoteException
```

gives you a all the users in the database

Returns

a list<User> with all the users in the database

Exceptions

<i>RemoteException</i>	
------------------------	--

Implemented in [server.remote.Remote](#).

6.10.1.12 getBookByISBN()

```
Book server.remote.IRemote.getBookByISBN (
    int ISBN ) throws RemoteException
```

Get a book by its ISBN

Parameters

<i>ISBN</i>	unique book ID
-------------	----------------

Returns

a Book with the selected ISBN

Exceptions

<i>RemoteException</i>	
------------------------	--

Implemented in [server.remote.Remote](#).

6.10.1.13 `getBookByTitle()`

```
Book server.remote.IRemote.getBookByTitle (
    String title ) throws RemoteException
```

Get a Book by its title

Parameters

<i>title</i>	name of the book
--------------	------------------

Returns

a Book with the selected Title

Exceptions

<i>RemoteException</i>	
------------------------	--

Implemented in [server.remote.Remote](#).

6.10.1.14 `getBookReviews()`

```
List<Review> server.remote.IRemote.getBookReviews (
    String title ) throws RemoteException
```

Get all the Reviews of a Book

Parameters

<i>title</i>	of the book
--------------	-------------

Returns

List<Review>

Exceptions

<i>RemoteException</i>	
------------------------	--

Implemented in [server.remote.Remote](#).

6.10.1.15 `getReview()`

```
Review server.remote.IRemote.getReview (
    int id_review ) throws RemoteException
```

Get a Review by its ID

Parameters

<i>id_review</i>	unique number
------------------	---------------

Returns

a Review

Exceptions

<i>RemoteException</i>	
------------------------	--

Implemented in [server.remote.Remote](#).

6.10.1.16 `getUser()`

```
User server.remote.IRemote.getUser (
    String email ) throws RemoteException
```

Get a User by its Email

Parameters

<i>email</i>	
--------------	--

Returns

a User

Exceptions

<i>RemoteException</i>	
------------------------	--

Implemented in [server.remote.Remote](#).

6.10.1.17 getUserReviews()

```
List<Review> server.remote.IRemote.getUserReviews (
    String email ) throws RemoteException
```

Get all the Reviews a User Made

Parameters

<i>email</i>	of the user
--------------	-------------

Returns

a List<Review>

Exceptions

<i>RemoteException</i>	
------------------------	--

Implemented in [server.remote.Remote](#).

6.10.1.18 registerUser()

```
boolean server.remote.IRemote.registerUser (
    String email,
    String password,
    boolean role ) throws RemoteException
```

creates a new user

Parameters

<i>email</i>	
<i>password</i>	
<i>role</i>	(true admin; false user)

Returns

true or false to tell if it worked

Exceptions

<i>RemoteException</i>	
------------------------	--

Implemented in [server.remote.Remote](#).

6.10.1.19 showBooksInStore()

```
List<Book> server.remote.IRemote.showBooksInStore ( ) throws RemoteException
```

gives you a all the books in the database

Returns

a list<Book> with all the books in the database

Exceptions

<i>RemoteException</i>	
------------------------	--

Implemented in [server.remote.Remote](#).

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

- [src/main/java/server/remote/IRemote.java](#)

6.11 client.gui.Login Class Reference

Public Member Functions

- [Login](#) ()

Static Public Member Functions

- static void [main](#) (String[] args)

6.11.1 Constructor & Destructor Documentation

6.11.1.1 Login()

```
client.gui.Login.Login ( )
```

Create the application.

6.11.2 Member Function Documentation

6.11.2.1 main()

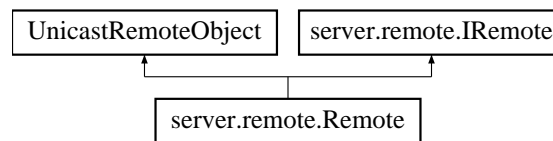
```
static void client.gui.Login.main (
    String [] args ) [static]
```

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

- [src/main/java/client/gui/Login.java](#)

6.12 server.remote.Remote Class Reference

Inheritance diagram for server.remote.Remote:



Public Member Functions

- [Remote](#) () throws RemoteException
- boolean [registerUser](#) (String email, String password, boolean role) throws RemoteException
- List< [Book](#) > [showBooksInStore](#) () throws RemoteException
- List< [User](#) > [getAllUsers](#) () throws RemoteException
- boolean [addBook](#) ([Book](#) book) throws RemoteException
- List< [Review](#) > [getAllReviews](#) () throws RemoteException
- [Book](#) [bookTest](#) ()
- [Book](#) [getBookByISBN](#) (int ISBN) throws RemoteException
- [Book](#) [getBookByTitle](#) (String title) throws RemoteException
- [Review](#) [getReview](#) (int id_review) throws RemoteException
- [User](#) [getUser](#) (String email) throws RemoteException
- boolean [buyBook](#) (String email, String book_title) throws RemoteException
- boolean [addReview](#) ([Book](#) b, [Review](#) r, [User](#) u)
- List< [Review](#) > [getUserReviews](#) (String email)
- List< [Review](#) > [getBookReviews](#) (String title)
- double [averageRatingByBook](#) (String title)
- double [averageRatingByUser](#) (String email)
- void [deleteReview](#) (int id_review)
- void [deleteBook](#) (int ISBN)
- boolean [addBook](#) (int ISBN, String title, String category, String edition, String author, double price, String description, String img)

6.12.1 Constructor & Destructor Documentation

6.12.1.1 Remote()

`server.remote.Remote.Remote ()` throws `RemoteException`

6.12.2 Member Function Documentation

6.12.2.1 addBook() [1/2]

`boolean server.remote.Remote.addBook (
 Book book)` throws `RemoteException`

adds a book to the database

Parameters

<i>book</i>	
-------------	--

Returns

true or false to tell if it worked

Exceptions

<i>RemoteException</i>	
------------------------	--

Implements [server.remote.IRemote](#).

6.12.2.2 addBook() [2/2]

```
boolean server.remote.Remote.addBook (   
    int ISBN,   
    String title,   
    String category,   
    String edition,   
    String author,   
    double price,   
    String description,   
    String img )
```

Returns

boolean true if the book is created and stored in db

Implements [server.remote.IRemote](#).

6.12.2.3 addReview()

```
boolean server.remote.Remote.addReview (
    Book b,
    Review r,
    User u )
```

add a new review to a book

Parameters

<i>b</i>	the book to add a review
<i>r</i>	the Review
<i>u</i>	the User who made the review

Returns

true or false to tell if it worked

Exceptions

<i>RemoteException</i>	
------------------------	--

Implements [server.remote.IRemote](#).

6.12.2.4 averageRatingByBook()

```
double server.remote.Remote.averageRatingByBook (
    String title )
```

Get the average Rating of a book

Parameters

<i>title</i>	of the book
--------------	-------------

Returns

a double

Exceptions

<i>RemoteException</i>	
------------------------	--

Implements [server.remote.IRemote](#).

6.12.2.5 averageRatingByUser()

```
double server.remote.Remote.averageRatingByUser (
    String email )
```

Get the average Rating given by a User

Parameters

<i>email</i>	
--------------	--

Returns

a double

Exceptions

<i>RemoteException</i>	
------------------------	--

Implements [server.remote.IRemote](#).

6.12.2.6 bookTest()

```
Book server.remote.Remote.bookTest ( )
```

Test

Returns

the test book

Exceptions

<i>RemoteException</i>	
------------------------	--

Implements [server.remote.IRemote](#).

6.12.2.7 buyBook()

```
boolean server.remote.Remote.buyBook (
    String email,
    String book_title ) throws RemoteException
```

Buy a book

Parameters

<i>email</i>	of the user
<i>book_title</i>	title of the book

Returns

true of false to tell if it worked

Exceptions

<i>RemoteException</i>	
------------------------	--

Implements [server.remote.IRemote](#).

6.12.2.8 deleteBook()

```
void server.remote.Remote.deleteBook (
    int ISBN )
```

Eliminates a Book

Parameters

<i>ISBN</i>	
-------------	--

Exceptions

<i>RemoteException</i>	
------------------------	--

Implements [server.remote.IRemote](#).

6.12.2.9 deleteReview()

```
void server.remote.Remote.deleteReview (
    int id_review )
```

Eliminates a Review

Parameters

<i>id_review</i>	
------------------	--

Exceptions

<i>RemoteException</i>	
------------------------	--

Implements [server.remote.IRemote](#).

6.12.2.10 getAllReviews()

```
List<Review> server.remote.Remote.getAllReviews ( ) throws RemoteException
```

gives you a all the Review in the database

Returns

a list<[Review](#)> with all the books in the database

Exceptions

<i>RemoteException</i>	
------------------------	--

Implements [server.remote.IRemote](#).

6.12.2.11 getAllUsers()

```
List<User> server.remote.Remote.getAllUsers ( ) throws RemoteException
```

gives you a all the users in the database

Returns

a list<[User](#)> with all the users in the database

Exceptions

<i>RemoteException</i>	
------------------------	--

Implements [server.remote.IRemote](#).

6.12.2.12 `getBookByISBN()`

```
Book server.remote.Remote.getBookByISBN (
    int ISBN ) throws RemoteException
```

Get a book by its ISBN

Parameters

<i>ISBN</i>	unique book ID
-------------	----------------

Returns

a Book with the selected ISBN

Exceptions

<i>RemoteException</i>	
------------------------	--

Implements [server.remote.IRemote](#).

6.12.2.13 `getBookByTitle()`

```
Book server.remote.Remote.getBookByTitle (
    String title ) throws RemoteException
```

Get a Book by its title

Parameters

<i>title</i>	name of the book
--------------	------------------

Returns

a Book with the selected Title

Exceptions

<i>RemoteException</i>	
------------------------	--

Implements [server.remote.IRemote](#).

6.12.2.14 getBookReviews()

```
List<Review> server.remote.Remote.getBookReviews (
    String title )
```

Get all the Reviews of a Book

Parameters

<i>title</i>	of the book
--------------	-------------

Returns

List<Review>

Exceptions

<i>RemoteException</i>	
------------------------	--

Implements [server.remote.IRemote](#).

6.12.2.15 getReview()

```
Review server.remote.Remote.getReview (
    int id_review ) throws RemoteException
```

Get a Review by its ID

Parameters

<i>id_review</i>	unique number
------------------	---------------

Returns

a Review

Exceptions

<i>RemoteException</i>	
------------------------	--

Implements [server.remote.IRemote](#).

6.12.2.16 getUser()

```
User server.remote.Remote.getUser (
    String email ) throws RemoteException
```

Get a User by its Email

Parameters

<i>email</i>	
--------------	--

Returns

a User

Exceptions

<i>RemoteException</i>	
------------------------	--

Implements [server.remote.IRemote](#).

6.12.2.17 getUserReviews()

```
List<Review> server.remote.Remote.getUserReviews (
    String email )
```

Get all the Reviews a User Made

Parameters

<i>email</i>	of the user
--------------	-------------

Returns

a List<[Review](#)>

Exceptions

<i>RemoteException</i>	
------------------------	--

Implements [server.remote.IRemote](#).

6.12.2.18 registerUser()

```
boolean server.remote.Remote.registerUser (
    String email,
    String password,
    boolean role ) throws RemoteException
```

creates a new user

Parameters

<i>email</i>	
<i>password</i>	
<i>role</i>	(true admin; false user)

Returns

true or false to tell if it worked

Exceptions

<i>RemoteException</i>	
------------------------	--

Implements [server.remote.IRemote](#).

6.12.2.19 showBooksInStore()

```
List<Book> server.remote.Remote.showBooksInStore ( ) throws RemoteException
```

gives you a all the books in the database

Returns

a list<Book> with all the books in the database

Exceptions

<i>RemoteException</i>	
------------------------	--

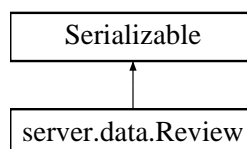
Implements [server.remote.IRemote](#).

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

- `src/main/java/server/remote/Remote.java`

6.13 server.data.Review Class Reference

Inheritance diagram for server.data.Review:



Public Member Functions

- [Review](#) ([User](#) user, [Book](#) book, String comment, double rating)
- [Review](#) ()
- [Review](#) (String comment, double rating)
- [User](#) [getUser](#) ()
- void [setUser](#) ([User](#) user)
- [Book](#) [getBook](#) ()
- void [setBook](#) ([Book](#) book)
- String [getComment](#) ()
- void [setComment](#) (String comment)
- double [getRating](#) ()
- void [setRating](#) (double rating)
- int [getId_review](#) ()
- String [toString](#) ()
- void [setId_review](#) (int id_review)

6.13.1 Constructor & Destructor Documentation

6.13.1.1 [Review\(\)](#) [1/3]

```
server.data.Review.Review (
    User user,
    Book book,
    String comment,
    double rating )
```

6.13.1.2 [Review\(\)](#) [2/3]

```
server.data.Review.Review ( )
```

6.13.1.3 [Review\(\)](#) [3/3]

```
server.data.Review.Review (
    String comment,
    double rating )
```

6.13.2 Member Function Documentation

6.13.2.1 getBook()

```
Book server.data.Review.getBook ( )
```

6.13.2.2 getComment()

```
String server.data.Review.getComment ( )
```

6.13.2.3 getId_review()

```
int server.data.Review.getId_review ( )
```

6.13.2.4 getRating()

```
double server.data.Review.getRating ( )
```

6.13.2.5 getUser()

```
User server.data.Review.getUser ( )
```

6.13.2.6 setBook()

```
void server.data.Review.setBook (
    Book book )
```

6.13.2.7 setComment()

```
void server.data.Review.setComment (
    String comment )
```

6.13.2.8 setId_review()

```
void server.data.Review.setId_review (
    int id_review )
```

6.13.2.9 setRating()

```
void server.data.Review.setRating (
    double rating )
```

6.13.2.10 setUser()

```
void server.data.Review.setUser (
    User user )
```

6.13.2.11 toString()

```
String server.data.Review.toString ( )
```

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

- src/main/java/server/data/[Review.java](#)

6.14 server.RMTest Class Reference

Public Member Functions

- void [setUpClient](#) ()
- void [registerNewUserTest](#) ()
- void [registerExistingUserTest](#) ()
- void [bookTestValidation](#) ()
- void [showBooksInStoreTest](#) ()
- void [showUsersTest](#) ()
- void [getThings](#) ()
- void [showUserFailTest](#) () throws RemoteException

Static Public Member Functions

- static junit.framework.Test [suite](#) ()
- static void [setUp](#) ()
- static void [tearDown](#) ()

6.14.1 Detailed Description

RMI Unit test for Simple Client / [Server](#) RMI Testing. Testing the only the Remoteness

6.14.2 Member Function Documentation

6.14.2.1 bookTestValidation()

```
void server.RMITest.bookTestValidation ( )
```

6.14.2.2 getThings()

```
void server.RMITest.getThings ( )
```

6.14.2.3 registerExistingUserTest()

```
void server.RMITest.registerExistingUserTest ( )
```

6.14.2.4 registerNewUserTest()

```
void server.RMITest.registerNewUserTest ( )
```

6.14.2.5 setUp()

```
static void server.RMITest.setUp ( ) [static]
```

6.14.2.6 setUpClient()

```
void server.RMITest.setUpClient ( )
```

6.14.2.7 showBooksInStoreTest()

```
void server.RMITest.showBooksInStoreTest ( )
```

6.14.2.8 showUserFailTest()

```
void server.RMITest.showUserFailTest ( ) throws RemoteException
```

6.14.2.9 showUsersTest()

```
void server.RMITest.showUsersTest ( )
```

6.14.2.10 suite()

```
static junit.framework.Test server.RMITest.suite ( ) [static]
```

6.14.2.11 tearDown()

```
static void server.RMITest.tearDown ( ) [static]
```

```
public void deleteDatabase() { PersistenceManagerFactory pmf = JDOHelper.getPersistenceManagerFactory("datanucleus.properties"); PersistenceManager pm = pmf.getPersistenceManager(); Transaction tx = pm.currentTransaction(); try { tx.begin();
```

```
logger.info("Deleting test users from persistence. Cleaning up."); Query<User> q1 = pm.newQuery(User.class); Query<Book> q2 = pm.newQuery(Book.class); Query<Review> q3 = pm.newQuery(Review.class); long numberInstancesDeleted1 = q1.deletePersistentAll(); long numberInstancesDeleted2 = q2.deletePersistentAll(); long numberInstancesDeleted3 = q3.deletePersistentAll();
```

```
        tx.commit();
    }
    finally
    {
        if (tx.isActive())
        {
            tx.rollback();
        }
        pm.close();
    }
}
```

```
}
```

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

- [src/test/java/server/RMITest.java](#)

6.15 server.Server Class Reference

Static Public Member Functions

- static void [main](#) (String[] args)

6.15.1 Member Function Documentation

6.15.1.1 main()

```
static void server.Server.main (  
    String [] args ) [static]
```

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

- src/main/java/server/[Server.java](#)

6.16 client.gui.ShowBooks Class Reference

Public Member Functions

- [ShowBooks](#) (String email)

Static Public Member Functions

- static void [main](#) (String[] args)

6.16.1 Constructor & Destructor Documentation

6.16.1.1 ShowBooks()

```
client.gui.ShowBooks.ShowBooks (  
    String email )
```

Create the application.

6.16.2 Member Function Documentation

6.16.2.1 main()

```
static void client.gui.ShowBooks.main (
    String [] args ) [static]
```

Launch the application.

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

- src/main/java/client/gui/[ShowBooks.java](#)

6.17 client.gui.ShowBooksAdmin Class Reference

Public Member Functions

- [ShowBooksAdmin](#) (String email)
- [ShowBooksAdmin](#) ()

Static Public Member Functions

- static void [main](#) (String[] args)

6.17.1 Constructor & Destructor Documentation

6.17.1.1 ShowBooksAdmin() [1/2]

```
client.gui.ShowBooksAdmin.ShowBooksAdmin (
    String email )
```

Create the application.

6.17.1.2 ShowBooksAdmin() [2/2]

```
client.gui.ShowBooksAdmin.ShowBooksAdmin ( )
```

6.17.2 Member Function Documentation

6.17.2.1 main()

```
static void client.gui.ShowBooksAdmin.main (
    String [] args ) [static]
```

Launch the application.

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

- src/main/java/client/gui/[ShowBooksAdmin.java](#)

6.18 client.gui.ShowDescription Class Reference

Public Member Functions

- [ShowDescription](#) (String title, String email) throws RemoteException

6.18.1 Constructor & Destructor Documentation

6.18.1.1 ShowDescription()

```
client.gui.ShowDescription.ShowDescription (
    String title,
    String email ) throws RemoteException
```

Launch the application. Create the application.

Parameters

<i>user</i>	
<i>user</i>	

Exceptions

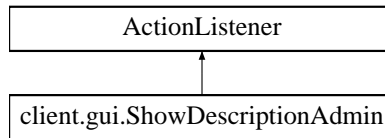
<i>RemoteException</i>	
------------------------	--

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

- src/main/java/client/gui/[ShowDescription.java](#)

6.19 client.gui.ShowDescriptionAdmin Class Reference

Inheritance diagram for client.gui.ShowDescriptionAdmin:



Public Member Functions

- [ShowDescriptionAdmin](#) (String title, String email)
- void [actionPerformed](#) (ActionEvent arg0)

Static Public Member Functions

- static void [main](#) (String[] args)

6.19.1 Constructor & Destructor Documentation

6.19.1.1 ShowDescriptionAdmin()

```

client.gui.ShowDescriptionAdmin.ShowDescriptionAdmin (
    String title,
    String email )
  
```

Create the application.

6.19.2 Member Function Documentation

6.19.2.1 actionPerformed()

```

void client.gui.ShowDescriptionAdmin.actionPerformed (
   (ActionEvent arg0 )
  
```

6.19.2.2 main()

```

static void client.gui.ShowDescriptionAdmin.main (
    String [] args ) [static]
  
```

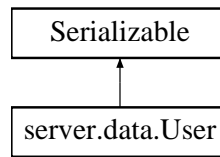
Launch the application.

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

- src/main/java/client/gui/[ShowDescriptionAdmin.java](#)

6.20 server.data.User Class Reference

Inheritance diagram for server.data.User:



Public Member Functions

- double [getMoney](#) ()
- void [setMoney](#) (double money)
- [User](#) (String email, String password, String name, String address, boolean role)
- [User](#) (String email, String password, boolean role)
- void [addReview](#) ([Review](#) review)
- void [removeReview](#) ([Review](#) review)
- List< [Review](#) > [getReviews](#) ()
- void [addBook](#) ([Book](#) book)
- void [removeBook](#) ([Book](#) book)
- List< [Book](#) > [getBooks](#) ()
- String [getEmail](#) ()
- void [setEmail](#) (String email)
- String [getPassword](#) ()
- void [setPassword](#) (String password)
- String [getName](#) ()
- void [setName](#) (String name)
- String [getAddress](#) ()
- void [setAddress](#) (String address)
- boolean [getRole](#) ()
- void [setRole](#) (boolean role)
- String [toString](#) ()

6.20.1 Constructor & Destructor Documentation

6.20.1.1 [User\(\)](#) [1/2]

```
server.data.User.User (
    String email,
    String password,
    String name,
    String address,
    boolean role )
```

6.20.1.2 User() [2/2]

```
server.data.User.User (
    String email,
    String password,
    boolean role )
```

6.20.2 Member Function Documentation

6.20.2.1 addBook()

```
void server.data.User.addBook (
    Book book )
```

6.20.2.2 addReview()

```
void server.data.User.addReview (
    Review review )
```

6.20.2.3 getAddress()

```
String server.data.User.getAddress ( )
```

6.20.2.4 getBooks()

```
List<Book> server.data.User.getBooks ( )
```

6.20.2.5 getEmail()

```
String server.data.User.getEmail ( )
```

6.20.2.6 getMoney()

```
double server.data.User.getMoney ( )
```

6.20.2.7 getName()

```
String server.data.User.getName ( )
```

6.20.2.8 getPassword()

```
String server.data.User.getPassword ( )
```

6.20.2.9 getReviews()

```
List<Review> server.data.User.getReviews ( )
```

6.20.2.10 getRole()

```
boolean server.data.User.getRole ( )
```

6.20.2.11 removeBook()

```
void server.data.User.removeBook (
    Book book )
```

6.20.2.12 removeReview()

```
void server.data.User.removeReview (
    Review review )
```

6.20.2.13 setAddress()

```
void server.data.User.setAddress (
    String address )
```

6.20.2.14 setEmail()

```
void server.data.User.setEmail (
    String email )
```

6.20.2.15 setMoney()

```
void server.data.User.setMoney (
    double money )
```

6.20.2.16 setName()

```
void server.data.User.setName (
    String name )
```

6.20.2.17 setPassword()

```
void server.data.User.setPassword (
    String password )
```

6.20.2.18 setRole()

```
void server.data.User.setRole (
    boolean role )
```

6.20.2.19 toString()

```
String server.data.User.toString ( )
```

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

- [src/main/java/server/data/User.java](#)

Chapter 7

File Documentation

7.1 src/main/java/client/Client.java File Reference

Classes

- class [client.Client](#)

Packages

- package [client](#)

7.2 src/main/java/client/gui/AddBooks.java File Reference

Classes

- class [client.gui.AddBooks](#)

Packages

- package [client.gui](#)

7.3 src/main/java/client/gui/Login.java File Reference

Classes

- class [client.gui.Login](#)

Packages

- package [client.gui](#)

7.4 `src/main/java/client/gui/ShowBooks.java` File Reference

Classes

- class [client.gui.ShowBooks](#)
- class **client.gui.BookTableModel**

Packages

- package [client.gui](#)

7.5 `src/main/java/client/gui/ShowBooksAdmin.java` File Reference

Classes

- class [client.gui.ShowBooksAdmin](#)

Packages

- package [client.gui](#)

7.6 `src/main/java/client/gui/ShowDescription.java` File Reference

Classes

- class [client.gui.ShowDescription](#)
- class **client.gui.ReviewTableModel**

Packages

- package [client.gui](#)

7.7 `src/main/java/client/gui/ShowDescriptionAdmin.java` File Reference

Classes

- class [client.gui.ShowDescriptionAdmin](#)

Packages

- package [client.gui](#)

7.8 src/main/java/db/DAO.java File Reference

Classes

- class [db.DAO](#)

Packages

- package [db](#)

7.9 src/main/java/db/DB.java File Reference

Classes

- class [db.DB](#)

Packages

- package [db](#)

7.10 src/main/java/db/IDAO.java File Reference

Classes

- interface [db.IDAO](#)

Packages

- package [db](#)

7.11 src/main/java/db/IDB.java File Reference

Classes

- interface [db.IDB](#)

Packages

- package [db](#)

7.12 src/main/java/server/data/Book.java File Reference

Classes

- class [server.data.Book](#)

Packages

- package [server.data](#)

7.13 src/main/java/server/data/Review.java File Reference

Classes

- class [server.data.Review](#)

Packages

- package [server.data](#)

7.14 src/main/java/server/data/User.java File Reference

Classes

- class [server.data.User](#)

Packages

- package [server.data](#)

7.15 src/main/java/server/remote/IRemote.java File Reference

Classes

- interface [server.remote.IRemote](#)

Packages

- package [server.remote](#)

7.16 src/main/java/server/remote/Remote.java File Reference

Classes

- class [server.remote.Remote](#)

Packages

- package [server.remote](#)

7.17 src/main/java/server/Server.java File Reference

Classes

- class [server.Server](#)

Packages

- package [server](#)

7.18 src/test/java/server/ContiPerfTest.java File Reference

Classes

- class [server.ContiPerfTest](#)

Packages

- package [server](#)

7.19 src/test/java/server/DAOMockTest.java File Reference

Classes

- class [server.DAOMockTest](#)

Packages

- package [server](#)

7.20 src/test/java/server/RMITest.java File Reference

Classes

- class [server.RMITest](#)

Packages

- package [server](#)

Index

- actionPerformed
 - client::gui::ShowDescriptionAdmin, 76
- addBook
 - server::data::User, 78
 - server::remote::IRemote, 48, 49
 - server::remote::Remote, 59
- addBookToDb
 - db::DB, 28
 - db::IDB, 41
- AddBooks
 - client::gui::AddBooks, 11
- addReview
 - db::DB, 29
 - db::IDB, 42
 - server::data::Book, 13
 - server::data::User, 78
 - server::remote::IRemote, 49
 - server::remote::Remote, 60
- addUser
 - server::data::Book, 13
- averageRatingByBook
 - db::DB, 29
 - db::IDB, 42
 - server::remote::IRemote, 50
 - server::remote::Remote, 60
- averageRatingByUser
 - db::DB, 30
 - db::IDB, 42
 - server::remote::IRemote, 50
 - server::remote::Remote, 61
- Book
 - server::data::Book, 12
- bookTest
 - server::remote::IRemote, 51
 - server::remote::Remote, 61
- bookTestValidation
 - server::RMITest, 71
- buyBook
 - client::Client, 16
 - db::DB, 30
 - db::IDB, 43
 - server::remote::IRemote, 51
 - server::remote::Remote, 61
- client, 9
- client.Client, 16
- client.gui, 9
- client.gui.AddBooks, 11
- client.gui.Login, 57
- client.gui.ShowBooks, 73
- client.gui.ShowBooksAdmin, 74
- client.gui.ShowDescription, 75
- client.gui.ShowDescriptionAdmin, 75
- client::Client
 - buyBook, 16
 - displayMenu, 17
 - displaySubMenu, 17
 - login, 17
 - main, 17
 - mainMenu, 17
 - menuBook, 17
 - menuReviews, 17
 - menuSearch, 18
 - menuShowBooks, 18
 - searchISBN, 18
 - searchTitle, 18
 - showBook, 18
 - showBooks, 18
 - signup, 18
- client::gui::AddBooks
 - AddBooks, 11
 - main, 11
- client::gui::Login
 - Login, 58
 - main, 58
- client::gui::ShowBooks
 - main, 73
 - ShowBooks, 73
- client::gui::ShowBooksAdmin
 - main, 74
 - ShowBooksAdmin, 74
- client::gui::ShowDescription
 - ShowDescription, 75
- client::gui::ShowDescriptionAdmin
 - actionPerformed, 76
 - main, 76
 - ShowDescriptionAdmin, 76
- DAO
 - db::DAO, 21
- DB
 - db::DB, 28
- db, 9
- db.DAO, 20
- db.DB, 27
- db.IDAO, 35
- db.IDB, 41
- db::DAO
 - DAO, 21

- deleteBook, 21
- deleteReview, 21
- getAllBooks, 22
- getAllReviews, 22
- getAllUsers, 22
- retrieveBook, 22
- retrieveBookByParameter, 23
- retrieveReview, 23
- retrieveUser, 23
- storeBook, 24
- storeUser, 24
- updateBook, 25
- updateReview, 25
- updateUser, 25
- db::DB
 - addBookToDb, 28
 - addReview, 29
 - averageRatingByBook, 29
 - averageRatingByUser, 30
 - buyBook, 30
 - DB, 28
 - deleteBook, 30
 - deleteReview, 31
 - getAllBooks, 31
 - getAllReviews, 31
 - getAllUsers, 32
 - getBookReviews, 32
 - getUserReviews, 32
 - registerUser, 33
 - showBookByISBN, 33
 - showBookByTitle, 34
 - showReview, 34
 - showUser, 35
- db::IDAO
 - deleteBook, 36
 - deleteReview, 36
 - getAllBooks, 36
 - getAllReviews, 36
 - getAllUsers, 37
 - retrieveBook, 37
 - retrieveBookByParameter, 37
 - retrieveReview, 38
 - retrieveUser, 38
 - storeBook, 39
 - storeUser, 39
 - updateBook, 39
 - updateReview, 40
 - updateUser, 40
- db::IDB
 - addBookToDb, 41
 - addReview, 42
 - averageRatingByBook, 42
 - averageRatingByUser, 42
 - buyBook, 43
 - deleteBook, 43
 - deleteReview, 44
 - getAllBooks, 44
 - getAllReviews, 44
 - getAllUsers, 44
 - getBookReviews, 45
 - getUserReviews, 45
 - registerUser, 45
 - showBookByISBN, 46
 - showBookByTitle, 46
 - showReview, 47
 - showUser, 47
- deleteBook
 - db::DAO, 21
 - db::DB, 30
 - db::IDAO, 36
 - db::IDB, 43
 - server::remote::IRemote, 52
 - server::remote::Remote, 62
- deleteReview
 - db::DAO, 21
 - db::DB, 31
 - db::IDAO, 36
 - db::IDB, 44
 - server::remote::IRemote, 52
 - server::remote::Remote, 62
- displayMenu
 - client::Client, 17
- displaySubMenu
 - client::Client, 17
- getAddress
 - server::data::User, 78
- getAllBooks
 - db::DAO, 22
 - db::DB, 31
 - db::IDAO, 36
 - db::IDB, 44
- getAllReviews
 - db::DAO, 22
 - db::DB, 31
 - db::IDAO, 36
 - db::IDB, 44
 - server::remote::IRemote, 53
 - server::remote::Remote, 63
- getAllUsers
 - db::DAO, 22
 - db::DB, 32
 - db::IDAO, 37
 - db::IDB, 44
 - server::remote::IRemote, 53
 - server::remote::Remote, 63
- getAuthor
 - server::data::Book, 13
- getBook
 - server::data::Review, 68
- getBookByISBN
 - server::remote::IRemote, 53
 - server::remote::Remote, 63
- getBookByTitle
 - server::remote::IRemote, 54
 - server::remote::Remote, 64
- getBookReviews

- db::DB, 32
- db::IDB, 45
- server::remote::IRemote, 54
- server::remote::Remote, 64
- getBooks
 - server::data::User, 78
- getCategory
 - server::data::Book, 13
- getComment
 - server::data::Review, 69
- getDescription
 - server::data::Book, 13
- getEdition
 - server::data::Book, 13
- getEmail
 - server::data::User, 78
- getISBN
 - server::data::Book, 14
- getId_review
 - server::data::Review, 69
- getImg
 - server::data::Book, 14
- getMoney
 - server::data::User, 78
- getName
 - server::data::User, 79
- getPassword
 - server::data::User, 79
- getPrice
 - server::data::Book, 14
- getRating
 - server::data::Review, 69
- getReview
 - server::remote::IRemote, 55
 - server::remote::Remote, 65
- getReviews
 - server::data::Book, 14
 - server::data::User, 79
- getRole
 - server::data::User, 79
- getThings
 - server::RMITest, 71
- getTitle
 - server::data::Book, 14
- getUser
 - server::data::Review, 69
 - server::remote::IRemote, 55
 - server::remote::Remote, 65
- getUserReviews
 - db::DB, 32
 - db::IDB, 45
 - server::remote::IRemote, 56
 - server::remote::Remote, 66
- getUsers
 - server::data::Book, 14
- Login
 - client::gui::Login, 58
- login
 - client::Client, 17
- main
 - client::Client, 17
 - client::gui::AddBooks, 11
 - client::gui::Login, 58
 - client::gui::ShowBooks, 73
 - client::gui::ShowBooksAdmin, 74
 - client::gui::ShowDescriptionAdmin, 76
 - server::Server, 73
- mainMenu
 - client::Client, 17
- menuBook
 - client::Client, 17
- menuReviews
 - client::Client, 17
- menuSearch
 - client::Client, 18
- menuShowBooks
 - client::Client, 18
- registerExistingUserTest
 - server::RMITest, 71
- registerNewUserTest
 - server::ContiPerfTest, 19
 - server::RMITest, 71
- registerUser
 - db::DB, 33
 - db::IDB, 45
 - server::remote::IRemote, 56
 - server::remote::Remote, 66
- Remote
 - server::remote::Remote, 59
- removeBook
 - server::data::User, 79
- removeReview
 - server::data::Book, 14
 - server::data::User, 79
- removeUser
 - server::data::Book, 14
- retrieveBook
 - db::DAO, 22
 - db::IDAO, 37
- retrieveBookByParameter
 - db::DAO, 23
 - db::IDAO, 37
- retrieveReview
 - db::DAO, 23
 - db::IDAO, 38
- retrieveUser
 - db::DAO, 23
 - db::IDAO, 38
- Review
 - server::data::Review, 68
- rule
 - server::ContiPerfTest, 20
- searchISBN
 - client::Client, 18

- searchTitle
 - client::Client, 18
- server, 10
- server.ContiPerfTest, 19
- server.DAOMockTest, 26
- server.data, 10
- server.data.Book, 12
- server.data.Review, 67
- server.data.User, 77
- server.RMITest, 70
- server.remote, 10
- server.remote.IRemote, 48
- server.remote.Remote, 58
- server.Server, 73
- server::ContiPerfTest
 - registerNewUserTest, 19
 - rule, 20
 - showBookThreads, 19
 - showBookThreads2, 19
 - storeBook, 20
 - storeBookDuration, 20
 - suite, 20
- server::DAOMockTest
 - setUp, 26
 - suite, 26
 - testAddBookInvalidRemote, 26
 - testAddBookValid, 27
 - testRegisterUserAlreadyExists, 27
 - testRegisterUserCorrectly, 27
- server::RMITest
 - bookTestValidation, 71
 - getThings, 71
 - registerExistingUserTest, 71
 - registerNewUserTest, 71
 - setUp, 71
 - setUpClient, 71
 - showBooksInStoreTest, 71
 - showUserFailTest, 72
 - showUsersTest, 72
 - suite, 72
 - tearDown, 72
- server::Server
 - main, 73
- server::data::Book
 - addReview, 13
 - addUser, 13
 - Book, 12
 - getAuthor, 13
 - getCategory, 13
 - getDescription, 13
 - getEdition, 13
 - getISBN, 14
 - getImg, 14
 - getPrice, 14
 - getReviews, 14
 - getTitle, 14
 - getUsers, 14
 - removeReview, 14
 - removeUser, 14
 - setAuthor, 15
 - setCategory, 15
 - setDescription, 15
 - setEdition, 15
 - setISBN, 15
 - setImg, 15
 - setPrice, 15
 - setTitle, 16
 - toString, 16
- server::data::Review
 - getBook, 68
 - getComment, 69
 - getId_review, 69
 - getRating, 69
 - getUser, 69
 - Review, 68
 - setBook, 69
 - setComment, 69
 - setId_review, 69
 - setRating, 70
 - setUser, 70
 - toString, 70
- server::data::User
 - addBook, 78
 - addReview, 78
 - getAddress, 78
 - getBooks, 78
 - getEmail, 78
 - getMoney, 78
 - getName, 79
 - getPassword, 79
 - getReviews, 79
 - getRole, 79
 - removeBook, 79
 - removeReview, 79
 - setAddress, 79
 - setEmail, 80
 - setMoney, 80
 - setName, 80
 - setPassword, 80
 - setRole, 80
 - toString, 80
 - User, 77
- server::remote::IRemote
 - addBook, 48, 49
 - addReview, 49
 - averageRatingByBook, 50
 - averageRatingByUser, 50
 - bookTest, 51
 - buyBook, 51
 - deleteBook, 52
 - deleteReview, 52
 - getAllReviews, 53
 - getAllUsers, 53
 - getBookByISBN, 53
 - getBookByTitle, 54
 - getBookReviews, 54

- getReview, 55
- getUser, 55
- getUserReviews, 56
- registerUser, 56
- showBooksInStore, 57
- server::remote::Remote
 - addBook, 59
 - addReview, 60
 - averageRatingByBook, 60
 - averageRatingByUser, 61
 - bookTest, 61
 - buyBook, 61
 - deleteBook, 62
 - deleteReview, 62
 - getAllReviews, 63
 - getAllUsers, 63
 - getBookByISBN, 63
 - getBookByTitle, 64
 - getBookReviews, 64
 - getReview, 65
 - getUser, 65
 - getUserReviews, 66
 - registerUser, 66
 - Remote, 59
 - showBooksInStore, 67
- setAddress
 - server::data::User, 79
- setAuthor
 - server::data::Book, 15
- setBook
 - server::data::Review, 69
- setCategory
 - server::data::Book, 15
- setComment
 - server::data::Review, 69
- setDescription
 - server::data::Book, 15
- setEdition
 - server::data::Book, 15
- setEmail
 - server::data::User, 80
- setISBN
 - server::data::Book, 15
- setId_review
 - server::data::Review, 69
- setImg
 - server::data::Book, 15
- setMoney
 - server::data::User, 80
- setName
 - server::data::User, 80
- setPassword
 - server::data::User, 80
- setPrice
 - server::data::Book, 15
- setRating
 - server::data::Review, 70
- setRole
 - server::data::User, 80
- setTitle
 - server::data::Book, 16
- setUp
 - server::DAOMockTest, 26
 - server::RMITest, 71
- setUpClient
 - server::RMITest, 71
- setUser
 - server::data::Review, 70
- showBook
 - client::Client, 18
- showBookByISBN
 - db::DB, 33
 - db::IDB, 46
- showBookByTitle
 - db::DB, 34
 - db::IDB, 46
- showBookThreads
 - server::ContiPerfTest, 19
- showBookThreads2
 - server::ContiPerfTest, 19
- ShowBooks
 - client::gui::ShowBooks, 73
- showBooks
 - client::Client, 18
- ShowBooksAdmin
 - client::gui::ShowBooksAdmin, 74
- showBooksInStore
 - server::remote::IRemote, 57
 - server::remote::Remote, 67
- showBooksInStoreTest
 - server::RMITest, 71
- ShowDescription
 - client::gui::ShowDescription, 75
- ShowDescriptionAdmin
 - client::gui::ShowDescriptionAdmin, 76
- showReview
 - db::DB, 34
 - db::IDB, 47
- showUser
 - db::DB, 35
 - db::IDB, 47
- showUserFailTest
 - server::RMITest, 72
- showUsersTest
 - server::RMITest, 72
- signUp
 - client::Client, 18
- src/main/java/client/Client.java, 81
- src/main/java/client/gui/AddBooks.java, 81
- src/main/java/client/gui/LogIn.java, 81
- src/main/java/client/gui/ShowBooks.java, 82
- src/main/java/client/gui/ShowBooksAdmin.java, 82
- src/main/java/client/gui/ShowDescription.java, 82
- src/main/java/client/gui/ShowDescriptionAdmin.java, 82
- src/main/java/db/DAO.java, 83
- src/main/java/db/DB.java, 83

- src/main/java/db/IDAO.java, [83](#)
- src/main/java/db/IDB.java, [83](#)
- src/main/java/server/Server.java, [85](#)
- src/main/java/server/data/Book.java, [84](#)
- src/main/java/server/data/Review.java, [84](#)
- src/main/java/server/data/User.java, [84](#)
- src/main/java/server/remote/IRemote.java, [84](#)
- src/main/java/server/remote/Remote.java, [85](#)
- src/test/java/server/ContiPerfTest.java, [85](#)
- src/test/java/server/DAOMockTest.java, [85](#)
- src/test/java/server/RMITest.java, [85](#)
- storeBook
 - db::DAO, [24](#)
 - db::IDAO, [39](#)
 - server::ContiPerfTest, [20](#)
- storeBookDuration
 - server::ContiPerfTest, [20](#)
- storeUser
 - db::DAO, [24](#)
 - db::IDAO, [39](#)
- suite
 - server::ContiPerfTest, [20](#)
 - server::DAOMockTest, [26](#)
 - server::RMITest, [72](#)
- tearDown
 - server::RMITest, [72](#)
- testAddBookInvalidRemote
 - server::DAOMockTest, [26](#)
- testAddBookValid
 - server::DAOMockTest, [27](#)
- testRegisterUserAlreadyExists
 - server::DAOMockTest, [27](#)
- testRegisterUserCorrectly
 - server::DAOMockTest, [27](#)
- toString
 - server::data::Book, [16](#)
 - server::data::Review, [70](#)
 - server::data::User, [80](#)
- updateBook
 - db::DAO, [25](#)
 - db::IDAO, [39](#)
- updateReview
 - db::DAO, [25](#)
 - db::IDAO, [40](#)
- updateUser
 - db::DAO, [25](#)
 - db::IDAO, [40](#)
- User
 - server::data::User, [77](#)