

Table Of Content

database	2
Configuration	2
Connector	4
mail	7
MailTransfer	7
object	9
Bill	9
History	11
ObjectHandler	13
Product	20
User	21
server	25
Account	25
Authentication	28
Booking	29
Content	31
Search	33
Website	36
util	43
TimerThread	43
Util	44
Index	48

Package database

Class Summary

[Configuration](#)

A simple class to read the database connection information from.

[Connector](#)

Simple Class to connect to a database and use the read/write/update functions of the database.

database

Class Configuration

```
java.lang.Object
|
+--database.Configuration
```

< [Constructors](#) > < [Methods](#) >

```
public class Configuration
extends java.lang.Object
```

A simple class to read the database connection information from.

Author:

Chris

Constructors

Configuration

```
public Configuration()
```

Creates a Configuration Object with standard values.

Methods

getDatabase

```
public java.lang.String getDatabase()
```

Returns:

The databases name

getMailAddress

```
public java.lang.String getMailAddress()
```

Returns:

The mail address to send mails from

getMailPassword

```
public java.lang.String getMailPassword()
```

Returns:

The password to log in to the mail account

getPassword

```
public java.lang.String getPassword()
```

Returns:

The users password

getPort

```
public java.lang.String getPort()
```

Returns:

The database server port

getServer

```
public java.lang.String getServer()
```

Returns:

The database server url

getUser

```
public java.lang.String getUser()
```

Returns:

The database user

getUserTimeout

```
public int getUserTimeout()
```

Returns:

the time in milliseconds it takes to automatically logout users

database

Class Connector

```
java.lang.Object
|
+--database.Connector
```

< [Constructors](#) > < [Methods](#) >

```
public class Connector
extends java.lang.Object
```

Simple Class to connect to a database and use the read/write/update functions of the database. The database connection information is read from a Configuration object.

Author:

Chris

Constructors

Connector

```
public Connector(Configuration c)
```

Creates a Connector object with the connection information from the given Configuration object. If errors occur they will get logged in the log files.

Parameters:

c - The Configuration to read the connection information from

Methods

deleteFromDatabase

```
public boolean deleteFromDatabase(java.lang.String table,  
                                 java.lang.String[][] condition)
```

Deletes all values which fit the pattern given by columnValue in the given table.

Parameters:

table - The table to delete from

condition - The pattern to find the values to be deleted.

Returns:

True if the operation went fine, false if not.

insertToDatabase

```
public boolean insertToDatabase(java.lang.String table,  
                                java.lang.String[][] columnValue)
```

Inserts one row in the given table.

Parameters:

table - The table to insert in

columnValue - First column of the array is the column name. Second column is the value being inserted.

Returns:

True if the operation went fine, false if not.

readFromDatabase

```
public java.sql.ResultSet readFromDatabase(java.lang.String table,  
                                           java.lang.String[] columns,  
                                           java.lang.String[][] condition)
```

Reads the columns from the given table with the given conditions

Parameters:

table - The table to read from

columns - The columns

condition - The conditions

Returns:

A ResultSet containing the read data

readFromDatabase

```
public java.sql.ResultSet readFromDatabase(java.lang.String table,  
                                           java.lang.String[] columns,  
                                           java.lang.String[][] condition,  
                                           int limit,  
                                           java.lang.String order)
```

Reads the columns from the given table with the given conditions and the given limit

Parameters:

table - The table to read from
columns - The columns
condition - The conditions
limit - The number of read lines

Returns:

A ResultSet containing the read data

updateDatabase

```
public boolean updateDatabase(java.lang.String table,  
                              java.lang.String[][] columnValueNew,  
                              java.lang.String[][] columnValueOld)
```

Updates the old value given by columnValueOld by the new value given by columnValueNew in the given table.

Parameters:

table - The table to be updated.
columnValueNew - The new value in the given column.
columnValueOld - The old value in the given column.

Returns:

True if the operation went fine, false if not.

Package mail

Class Summary

[MailTransfer](#)

Simple class to send mails.

mail

Class MailTransfer

```
java.lang.Object
|
+--mail.MailTransfer
```

< [Constructors](#) > < [Methods](#) >

```
public class MailTransfer
extends java.lang.Object
```

Simple class to send mails. Uses a given google mail account to send the mails from.

Author:

Chris

Constructors

MailTransfer

```
public MailTransfer()
```

Methods

sendMail

```
public boolean sendMail(java.lang.String recipient,  
                        java.lang.String msg,  
                        java.lang.String subject)
```

Sends a mail from the mail account to the given recipient.

Parameters:

recipient - The recipient
msg - The message
subject - The subject

Returns:

True if mail was sent, false if not.

Package object

Class Summary

[Bill](#)

[History](#)

A simple class representing a buy history of a user.

[ObjectHandler](#)

A simple ObjectHandler that is able to interact with the database

[Product](#)

Represents a line in the Product table of the database

[User](#)

A simple User class, representing a line in the User Table of the database

object

Class Bill

```
java.lang.Object
|
+--object.Bill
```

< [Constructors](#) > < [Methods](#) >

```
public class Bill
extends java.lang.Object
```

Author:

Chris Simple class representing a bill in the database

Constructors

Bill

```
public Bill(int id,  
            java.lang.String date,  
            java.lang.String user,  
            double amount)
```

Creates a bill with given values

Parameters:

id - the bill id
date - the date of creation
user - the users mail
amount - the amount to pay

Bill

```
public Bill(int id,  
            java.lang.String date,  
            java.lang.String user,  
            double amount,  
            java.util.ArrayList history)
```

Creates a bill with given values

Parameters:

id - the bill id
date - the date of creation
user - the users mail
amount - the amount to pay

Methods

getAmount

```
public double getAmount()
```

Returns:

the Amount

getDate

```
public java.lang.String getDate()
```

Returns:

the Date

getHistory

```
public java.lang.String getHistory()
```

Returns:

the History in String representation

getID

```
public int getID()
```

Returns:

the ID

getUser

```
public java.lang.String getUser()
```

Returns:

the User

object

Class History

```
java.lang.Object  
|  
+--object.History
```

< [Constructors](#) > < [Methods](#) >

```
public class History  
extends java.lang.Object
```

A simple class representing a buy history of a user.

Author:

Chris

Constructors

History

```
public History(int id,  
               java.lang.String date,  
               double price,  
               java.lang.String name,  
               int user,  
               boolean payed)
```

Creates a History with the given values

Parameters:

id - The id
date - The date
price - The price of the product
name - The product name
user - The user
payed - The payment status

Methods

getDate

```
public java.lang.String getDate()
```

Returns:

the Date

getID

```
public int getID()
```

Returns:

the ID

getName

```
public java.lang.String getName()
```

Returns:

The booked products Name

getPrice

```
public double getPrice()
```

Returns:

The booked Products Price

getUser

```
public int getUser()
```

Returns:

the User

isPayed

```
public boolean isPayed()
```

Returns:

True if the product was already payed, false if not

object

Class ObjectHandler

```
java.lang.Object  
|  
+--object.ObjectHandler
```

< [Constructors](#) > < [Methods](#) >

```
public class ObjectHandler  
extends java.lang.Object
```

A simple ObjectHandler that is able to interact with the database

Author:

Chris

Constructors

ObjectHandler

```
public ObjectHandler()
```

Methods

changePW

```
public static boolean changePW(int id,  
                                java.lang.String password)
```

Changes the password of the user specified by the id.

Parameters:

id - The users id
password - The new password

Returns:

True if success, false if not.

checkLogin

```
public static int checkLogin(java.lang.String sessionId)
```

Checks if the given session id is linked to a logged in account.

Parameters:

sessionId - The session id

Returns:

True if logged in, false if not.

createUser

```
public static boolean createUser(User u)
```

Writes the given User to the database

Parameters:

u - The user

Returns:

True if action was successful, false if not.

deleteBill

```
public static boolean deleteBill(java.lang.String id)
```

Deletes a bill but leaves all associated History entries on the unpaid status

Parameters:

id - The bills id

Returns:

True if success, false if not

deleteBillAndHistory

```
public static boolean deleteBillAndHistory(java.lang.String id)
```

Deletes a bill and sets all associated History entries on the paid status

Parameters:

id - The bills id

Returns:

True if success, false if not

deleteUser

```
public static boolean deleteUser(java.lang.String mail)
```

Deletes a given user and all information about it.

Parameters:

mail - The users mail

Returns:

True if success, false if not

getAllUsers

```
public static java.util.ArrayList getAllUsers()
```

Returns a list of all users with set values for mail, name, prename and id

Returns:

The list

getBills

```
public static java.util.ArrayList getBills()
```

A list of all bills.

Returns:

The list

getBills

```
public static java.util.ArrayList getBills(java.lang.String mail)
```

A list of all bills of the user specified by the mail

Parameters:

mail - The users mail

Returns:

The list

getUInfo

```
public static User getUInfo(java.lang.String value,  
                             java.lang.String key)
```

Returns a user with set values for id, mail, name, prename. Can search with three different keys: mail, name, prename.

Parameters:

value - The value of the key to search for.

key - The key

Returns:

The user object

getUser

```
public static java.util.ArrayList getUser(java.lang.String mail)
```

Searches for the user with the given mail in the database.

Parameters:

mail - The users mail

Returns:

A list of users with the given mail

getUserHistory

```
public static java.util.ArrayList getUserHistory(int user,  
int limit)
```

Reads the last x entries in the specified users History

Parameters:

user - The user
limit - The limit x

Returns:

The entries

getUserHistoryUnpaid

```
public static java.util.ArrayList getUserHistoryUnpaid(int user)
```

Reads the last x entries in the specified users History

Parameters:

user - The user

Returns:

The entries

getUserInfo

```
public static User getUserInfo(java.lang.String sessionId)
```

Returns a simple user with set values for mail, name and prename.

Parameters:

sessionId - The session id of the user

Returns:

The user object

isAdmin

```
public static boolean isAdmin(int id)
```

Checks if the user specified by the id is an admin

Parameters:

id - The users id

Returns:

True if it is an admin, false if not

removeProduct

```
public static boolean removeProduct(Product p)
```

Removes a product from the database

Parameters:

p - The product to delete

Returns:

True if success, false if failure

searchForProduct

```
public static java.util.ArrayList searchForProduct(java.lang.String[][] condition)
```

Searches for specific products which fit the conditions

Parameters:

condition - The search conditions

Returns:

A list of the found products

searchUser

```
public static java.util.ArrayList searchUser(java.lang.String[][] condition)
```

Searches for users that fit the given condition

Parameters:

condition - The condition in a key-value array.

Returns:

The list of users

updateProduct

```
public static boolean updateProduct(Product pOld,  
                                     Product pNew)
```

Updates the information about a product in the database

Parameters:

pOld - The old product information

pNew - The new product information

Returns:

The success of the operation

userLogin

```
public static boolean userLogin(java.lang.String sessionId,  
                                 java.lang.String mail)
```

Logs a user in.

Parameters:

sessionId - The session id of the new session
mail - The users mail

Returns:

True if success, false if not

userLogout

```
public static boolean userLogout(java.lang.String sessionId)
```

Logs a user out.

Parameters:

sessionId - The session of the user

Returns:

True if success, false if not.

writeBill

```
public static boolean writeBill(Bill b)
```

Writes a bill to the database

Parameters:

b - The bill

Returns:

True if success, false if not.

writeHistory

```
public static boolean writeHistory(History history)
```

Writes a history item into the database

Parameters:

history - The item

Returns:

The success

writeProductToDB

```
public static boolean writeProductToDB(Product product)
```

Writes a new product to the Database

Parameters:

product - the new product information

Returns:

The success of the operation

object

Class Product

```
java.lang.Object  
|  
+--object.Product
```

< [Constructors](#) > < [Methods](#) >

```
public class Product  
extends java.lang.Object
```

Represents a line in the Product table of the database

Author:

Chris

Constructors

Product

```
public Product(java.lang.String barcode,  
               java.lang.String name,  
               java.lang.String price)
```

Creates a Product with the given values

Parameters:

barcode - The barcode

name - The name

price - The price

Methods

getBarcode

```
public long getBarcode()
```

Returns:

the Barcode

getName

```
public java.lang.String getName()
```

Returns:

the Name

getPrice

```
public double getPrice()
```

Returns:

the Price

object

Class User

```
java.lang.Object  
|  
+--object.User
```

< [Constructors](#) > < [Methods](#) >

```
public class User  
extends java.lang.Object
```

A simple User class, representing a line in the User Table of the database

Author:

Chris

Constructors

User

```
public User(java.lang.String mail,  
            java.lang.String password)
```

Creates a user with the given values

Parameters:

mail - The users email address
password - The users password

User

```
public User(java.lang.String mail,  
            java.lang.String prename,  
            java.lang.String name,  
            double money)
```

Creates a user with all possible information

Parameters:

mail - The users Email-address
prename - The users Prenom
name - The users name
money - The users current money to pay

User

```
public User(java.lang.String mail,  
            java.lang.String prename,  
            java.lang.String name,  
            int ID)
```

Creates a user with the given values.

Parameters:

mail - The mail
prename - The prename
name - The name
ID - The id

User

```
public User(java.lang.String mail,  
            java.lang.String password,  
            java.lang.String prename,  
            java.lang.String name)
```

Creates a User with the given values, all other variables have default value.

Parameters:

mail - The users e-mail address
password - The users password
prename - The users prename
name - The users name

Methods

getID

```
public int getID()
```

Returns:

The users unique id

getMail

```
public java.lang.String getMail()
```

Returns:

the Mail

getMoney

```
public double getMoney()
```

Returns:

The current money to be paid

getName

```
public java.lang.String getName()
```

Returns:

The name of the user

getPassword

```
public java.lang.String getPassword()
```

Returns:

the Password

getPreName

```
public java.lang.String getPreName()
```

Returns:

The Prenom of the User

setID

```
public void setID(int id)
```

Sets the users unique ID

Parameters:

id - The users id

setMoney

```
public void setMoney(double price)
```

Sets the money the user has to pay

Parameters:

price - The new debt

Package server

Class Summary

[Account](#)

Class which provides methods to get information about all relevant Account information/options

[Authentication](#)

Class which provides methods of authentication such as login/logout and registration

[Booking](#)

Class that provides methods relevant for booking or product changes.

[Content](#)

Simple class to provide content such as images or css files

[Search](#)

The class to search for specified Objects in the database

[Website](#)

Class that provides Access to the website of the kitchen app and a small administration

server

Class Account

```
java.lang.Object
|
+--server.Account
```

< [Constructors](#) > < [Methods](#) >

```
public class Account
extends java.lang.Object
```

Class which provides methods to get information about all relevant Account information/options

Author:

Chris

Constructors

Account

```
public Account()
```

Methods

account

```
public java.lang.String account(java.lang.String session)
```

Information about the current logged in Account

Returns:

The information in String representation

adminList

```
public java.lang.String adminList(java.lang.String session)
```

Parameters:

session - The current users session id. It must be an Admin account to use this function.

Returns:

The List of all users in String representation, Failed authentication or Access denied

changePW

```
public java.lang.String changePW(java.lang.String session,  
                                   java.lang.String password)
```

Method to change the password of the current user. Only the current user can change his password!

Parameters:

session - The current users session id
password - The new password

Returns:

Failed authentication, check or fail

confirmBill

```
public java.lang.String confirmBill(java.lang.String session,  
                                     java.lang.String id)
```

Method to confirm the bill with the given id.

Parameters:

session - The current users session id. Must be admin to use this method.

id - The id of the bill

Returns:

Failed authentication, Access denied, check or fail

getBills

```
public java.lang.String getBills(java.lang.String session)
```

A list of all bills

Parameters:

session - The current users session id. Must be an Admin to use this method.

Returns:

Failed authentication, Access denied or the list of all Bills in String representation.

isAdmin

```
public java.lang.String isAdmin(java.lang.String session)
```

Returns:

Returns true if the user linked to the sessionID is a Admin, false if not

sendMail

```
public java.lang.String sendMail(javax.servlet.ServletContext context,  
                                   java.lang.String session,  
                                   java.lang.String mail)
```

Sends a reminder mail to the given mail address with information about the linked users bill. If sending the mail fails, the bill will be deleted again

Parameters:

session - The current users session ID

mail - The recipients mail address

Returns:

Failed authentication or Access denied or Mail sent or Failed to send Mail

server

Class Authentication

```
java.lang.Object
|
+--server.Authentication
```

< [Constructors](#) > < [Methods](#) >

```
public class Authentication
extends java.lang.Object
```

Class which provides methods of authentication such as login/logout and registration

Author:

Chris

Constructors

Authentication

```
public Authentication()
```

Methods

login

```
public java.lang.String login(java.lang.String mail,
                               java.lang.String password)
```

Logs a user in. Also starts a timer which logs out the user after 10 minutes.

Parameters:

mail - The users mail

password - The users password

Returns:

no such user, wrong password, failed to login or at success the id of the current session.

logout

```
public java.lang.String logout(java.lang.String session)
```

Logs a user out

Parameters:

session - The session id linked to the user

Returns:

failed authentication, Check or fail.

register

```
public java.lang.String register(javax.servlet.ServletContext context,  
    java.lang.String mail,  
    java.lang.String password,  
    java.lang.String name,  
    java.lang.String prename)
```

Registers a new user to the system.

Parameters:

mail - The users mail

password - The password

name - The name

prename - The prename

Returns:

user already exists, check or failed insert into database

server

Class Booking

```
java.lang.Object  
|  
+--server.Booking
```

< [Constructors](#) > < [Methods](#) >

```
public class Booking  
extends java.lang.Object
```

Class that provides methods relevant for booking or product changes.

Author:

Chris

Constructors

Booking

```
public Booking()
```

Methods

addProduct

```
public java.lang.String addProduct(java.lang.String session,  
                                     java.lang.String code,  
                                     java.lang.String name,  
                                     java.lang.String price)
```

Adds a product to the database

Parameters:

session - The users session id (must be an admin account)
code - The product code
name - The products name
price - The products price

Returns:

failed authentication, Access denied, code already in use, check, fail

book

```
public java.lang.String book(java.lang.String session,  
                               java.lang.String code)
```

Books a product on the users account linked to the session id.

Parameters:

session - The session id
code - The product code

Returns:

failed authentication, fail or check

changeProduct

```
public java.lang.String changeProduct(java.lang.String session,
                                     java.lang.String nameOld,
                                     java.lang.String codeOld,
                                     java.lang.String priceOld,
                                     java.lang.String nameNew,
                                     java.lang.String codeNew,
                                     java.lang.String priceNew)
```

Changes a product

Parameters:

session - The session id (must be an admin account!)
nameOld - The old name
codeOld - The old code
priceOld - The old price
nameNew - The new name
codeNew - The new code
priceNew - the new price

Returns:

failed authentication, Access denied, new code already in use, check, fail

delete

```
public java.lang.String delete(java.lang.String session,
                               java.lang.String code)
```

Deletes a product from the database (already booked products will still be listed in the histories)

Parameters:

session - The users session (must be an admin!)
code - The code

Returns:

failed authentication, Access denied, check, fail

server

Class Content

```
java.lang.Object
|
+--server.Content
```

< [Constructors](#) > < [Methods](#) >

```
public class Content
extends java.lang.Object
```

Simple class to provide content such as images or css files

Author:

Chris

Constructors

Content

```
public Content()
```

Methods

getCertificat

```
public javax.ws.rs.core.Response getCertificat(javax.servlet.ServletContext context)
```

Returns the servers ssl certificate so clients can receive it and put it in their truststore

Parameters:

context - The ServletContext

Returns:

The kitchen.crt as attachment

getCss

```
public javax.ws.rs.core.Response getCss(javax.servlet.ServletContext context, java.lang.String name)
```

Returns a css file with the given name

Parameters:

context - The ServletContext

name - The filename and filetype

Returns:

The file as attachment

getFile

```
public javax.ws.rs.core.Response getFile(javax.servlet.ServletContext context,  
                                           java.lang.String name)
```

Returns the png image with the given name

Parameters:

context - The ServletContext
name - The images name

Returns:

The file as attachment

server

Class Search

```
java.lang.Object  
|  
+--server.Search
```

< [Constructors](#) > < [Methods](#) >

```
public class Search  
extends java.lang.Object
```

The class to search for specified Objects in the database

Author:

Chris

Constructors

Search

```
public Search()
```

Methods

deleteUser

```
public java.lang.String deleteUser(java.lang.String session,  
                                   java.lang.String mail)
```

Deletes a user from the database and all bookings associated with that account

Parameters:

session - The session (must be an Admin or the user himself!)

mail - The users mail

Returns:

Failed authentication, Access denied, open bills, check, fail

productByName

```
public java.lang.String productByName(java.lang.String key)
```

Returns the products specified by the given search key as a list

Parameters:

key - The name or code of the product

Returns:

The list in String representation

productExplicit

```
public java.lang.String productExplicit(java.lang.String name,  
                                        java.lang.String code)
```

Returns a list of Products specified by the given name and code

Parameters:

name - The name to search for

code - The code to search for

Returns:

The list in String representation

searchUserM

```
public java.lang.String searchUserM(java.lang.String session,  
                                       java.lang.String mail)
```

Searches for Users with the given mail

Parameters:

session - The session (must be an Admin!)

mail - The mail to search for

Returns:

Failed authentication, Access denied, no such user or the list of users.

searchUserN

```
public java.lang.String searchUserN(java.lang.String session,  
                                       java.lang.String name,  
                                       java.lang.String prename)
```

Searches all users with the given name and prename

Parameters:

session - The session (must be an Admin!)

name - The name to search for

prename - The prename to search for

Returns:

Failed authentication, Access denied, no such user, The list of users.

userHistory

```
public java.lang.String userHistory(java.lang.String limit,  
                                       java.lang.String session)
```

Returns a list of Histories from the current user with a given limit of size

Parameters:

limit - The amount of Histories

Returns:

The list in String representation

server

Class Website

```
java.lang.Object
|
+--server.Website
```

< [Constructors](#) > < [Methods](#) >

```
public class Website
extends java.lang.Object
```

Class that provides Access to the website of the kitchen app and a small administration

Author:

Chris

Constructors

Website

```
public Website()
```

Methods

addProduct

```
public java.lang.String addProduct(javax.servlet.ServletContext context,
                                     java.lang.String session,
                                     java.lang.String success)
```

The Product Management page to add products.

Parameters:

context - The ServletContext

session - The current users session id (must be an admin!)

success - The success flag for the last add action

Returns:

The html code for the add page

addProductPost

```
public java.lang.String addProductPost(java.lang.String session,  
                                       java.lang.String name,  
                                       java.lang.String code,  
                                       java.lang.String price)
```

The post method to add a product. Redirects to the product add page with a success flag

Parameters:

session - The current users session id (must be an admin!)

name - The name of the product to add

code - The code of the product to add

price - The price of the product to add

Returns:

A javascript redirection to the add page

administration

```
public java.lang.String administration(javax.servlet.ServletContext context,  
                                       java.lang.String session)
```

The administration overview

Parameters:

context - The ServletContext

session - The session of the current logged in user.

Returns:

The Administration overview

authentication

```
public java.lang.String authentication(java.lang.String mail,  
                                       java.lang.String password)
```

Logs a user in and redirects him to the administration overview. If the login failed He will be redirected to the login page with the error that occurred.

Parameters:

mail - The users mail

password - The users password

Returns:

A redirection java script.

billAdministration

```
public java.lang.String billAdministration(javax.servlet.ServletContext
context,
                                         java.lang.String session,
                                         java.lang.String success)
```

The bill management page

Parameters:

context - The ServletContext
session - The current users session id (must be an admin!)
success - The success flag of the last action

Returns:

The html code

changeProduct

```
public java.lang.String changeProduct(javax.servlet.ServletContext context,
                                       java.lang.String session,
                                       java.lang.String name,
                                       java.lang.String code,
                                       java.lang.String price,
                                       java.lang.String success)
```

The html code of the page to change a products data

Parameters:

context - The ServletContext
session - The current users session id (must be an admin!)
name - The products name
code - The products code
price - The products price
success - The success flag of the last change action

Returns:

The html code

changeProductPost

```
public java.lang.String changeProductPost(java.lang.String session,  
                                             java.lang.String name,  
                                             java.lang.String code,  
                                             java.lang.String price,  
                                             java.lang.String nameOld,  
                                             java.lang.String codeOld,  
                                             java.lang.String priceOld)
```

The post method to change a products data. Redirects to the change page with the success flag of this action

Parameters:

session - The current users session id (must be an admin!)
name - The products new name
code - The products new code
price - The products new price
nameOld - The products old name
codeOld - The products old code
priceOld - The products old price

Returns:

A javascript redirection

confirmBill

```
public java.lang.String confirmBill(java.lang.String session,  
                                       java.lang.String id)
```

The post method to confirm a given bill. Redirects to the bill management page

Parameters:

session - The session id of the current user (must be an admin!)
id - The id of the bill to delete

Returns:

The javascript redirection

deleteProduct

```
public java.lang.String deleteProduct(java.lang.String session,  
                                     java.lang.String code,  
                                     java.lang.String name,  
                                     java.lang.String price)
```

Deletes the specified product and redirects to the product management page with the success flag of this delete action

Parameters:

session - The current users session id (must be an admin!)
code - The code fo the product to delete
name - The name of the product to delete
price - The price of the product to delete

Returns:

A javascript to redirect to the product management page

deleteUser

```
public java.lang.String deleteUser(java.lang.String session,  
                                   java.lang.String mail)
```

Deletes a user and all information associated with it

Parameters:

session - The session
mail - The users mail

Returns:

A redirection to the User Management page with the status of the action

documentation

```
public java.lang.String documentation(javax.servlet.ServletContext context)
```

The documentation of this project

Returns:

The documentation page

documentationClient

```
public java.lang.String documentationClient(javax.servlet.ServletContext  
context)
```

The Client documentation of this project

Returns:

The documentation page

documentationServer

```
public java.lang.String documentationServer(javax.servlet.ServletContext context)
```

The server documentation of this project

Returns:

The documentation page

login

```
public java.lang.String login(javax.servlet.ServletContext context, java.lang.String error)
```

The login page

Parameters:

context - The ServletContext
error - The error of a previous action

Returns:

The login page

logout

```
public java.lang.String logout(java.lang.String session)
```

Logs the current user out.

Parameters:

session - The users session id

Returns:

A java redirection

main

```
public java.lang.String main(javax.servlet.ServletContext context)
```

The main page of the website

Parameters:

context - The ServletContext

Returns:

The Mainpage

productAdministration

```
public java.lang.String productAdministration(javax.servlet.ServletContext
context,
                                             java.lang.String session,
                                             java.lang.String success)
```

The Product Administration page.

Parameters:

context - The ServletContext
session - The current users session id (must be an admin!)
success - The success flag for any action on this page

Returns:

The html code for the Administration page

sendBill

```
public java.lang.String sendBill(javax.servlet.ServletContext context,
                                   java.lang.String session,
                                   java.lang.String mail)
```

Post method to send a bill mail to the given user. Will redirect to the User Management page with the flag if it was a success or not.

Parameters:

context - The ServletContext
session - The current users session id (must be an admin!)
mail - The users mail to send the bill to

Returns:

A redirect java script

userAdministration

```
public java.lang.String userAdministration(javax.servlet.ServletContext
context,
                                             java.lang.String session,
                                             java.lang.String success)
```

The user management page

Parameters:

context - The ServletContext
session - The current users session
success - The status of the previous action

Returns:

The management page

Package util

Class Summary

[TimerThread](#)

Timer class which logs out a given user after the time given in the Configuration

[Util](#)

Simple Util class specified for the kitchen-app project

util

Class TimerThread

```
java.lang.Object
|
+--util.TimerThread
```

All Implemented Interfaces:

java.lang.Runnable

< [Constructors](#) > < [Methods](#) >

```
public class TimerThread
extends java.lang.Object
implements java.lang.Runnable
```

Timer class which logs out a given user after the time given in the Configuration

Author:

Chris

Constructors

TimerThread

```
public TimerThread(java.lang.String sessionId,
                    javax.servlet.http.HttpSession session)
```

Creates a new Timer

Parameters:

sessionId - The users Session id

session - The HttpSession Object which will be invalidated after time is over

Methods

run

```
public void run()  
    Starts the timer
```

start

```
public synchronized void start()  
    Sets the thread and starts it
```

util

Class Util

```
java.lang.Object  
|  
+--util.Util
```

< [Constructors](#) > < [Methods](#) >

```
public class Util  
    extends java.lang.Object
```

Simple Util class specified for the kitchen-app project

Author:
Chris

Constructors

Util

```
public Util()
```

Methods

ATSBill

```
public static java.lang.String ATSBill(java.util.ArrayList list)
```

Converts a list of bills into its string representation

Parameters:

list - The list to convert

Returns:

The string representation

ATSHistory

```
public static java.lang.String ATSHistory(java.util.ArrayList list)
```

Converts a list of histories to a String which represents the History list as a list of Strings.

Parameters:

list - The History list

Returns:

The String

ATSProduct

```
public static java.lang.String ATSProduct(java.util.ArrayList list)
```

Converts a list of products to a String which represents the Product list as a list of Strings.

Parameters:

list - The Product list

Returns:

The String

ATSUser

```
public static java.lang.String ATSUser(java.util.ArrayList list)
```

Converts a list of Users to a String which represents the User list as a list of Strings.

Parameters:

list - The User list

Returns:

The String

STABill

```
public static java.util.ArrayList STABill(java.lang.String s)
```

Converts a string that represents a list of bills into this list

Parameters:

s - The string to convert

Returns:

The list of bills

STAHistory

```
public static java.util.ArrayList STAHistory(java.lang.String s)
```

Converts a String which represents a List of Histories to a List of Histories.

Parameters:

s - The String

Returns:

The History List

STAProduct

```
public static java.util.ArrayList STAProduct(java.lang.String s)
```

Converts a String which represents a List of Products to a List of Products.

Parameters:

s - The String

Returns:

The Product List

STAUser

```
public static java.util.ArrayList STAUser(java.lang.String s)
```

Converts a String which represents a List of Users to a List of Users.

Parameters:

s - The String

Returns:

The User List

appendList

```
public static java.util.ArrayList appendList(java.util.ArrayList list1,  
                                              java.util.ArrayList list2)
```

Appends list1 with elements from list2 that are not in list1 Note: Changes list1 during process!

Parameters:

list1 - the first list
list2 - the second list

Returns:

the changed list1

getFile

```
public static java.lang.String getFile(javax.servlet.ServletContext context,  
                                       java.lang.String name)
```

Reads the given file and returns its content in String representation

Parameters:

name - The name + type of the file

Returns:

The file content

getHash

```
public static java.lang.String getHash(java.lang.String text)
```

Creates the MD5 Hash of the given text

Parameters:

text - the text

Returns:

the hash

INDEX

A

[account](#) ... 26
[addProduct](#) ... 30
[addProduct](#) ... 36
[addProductPost](#) ... 37
[administration](#) ... 37
[adminList](#) ... 26
[appendList](#) ... 47
[authentication](#) ... 37
[Account](#) ... 25
[Account](#) ... 25
[ATSBill](#) ... 45
[ATSHistory](#) ... 45
[ATSPProduct](#) ... 45
[ATSUser](#) ... 45
[Authentication](#) ... 28
[Authentication](#) ... 28

B

[billAdministration](#) ... 38
[book](#) ... 30
[Bill](#) ... 9
[Bill](#) ... 10
[Bill](#) ... 10
[Booking](#) ... 29
[Booking](#) ... 30

C

[changeProduct](#) ... 31
[changeProduct](#) ... 38
[changeProductPost](#) ... 39
[changePW](#) ... 14
[changePW](#) ... 26
[checkLogin](#) ... 14
[confirmBill](#) ... 27
[confirmBill](#) ... 39
[createUser](#) ... 14
[Configuration](#) ... 2
[Configuration](#) ... 2
[Connector](#) ... 4
[Connector](#) ... 4
[Content](#) ... 31
[Content](#) ... 32

D

[delete](#) ... 31
[deleteBill](#) ... 15
[deleteBillAndHistory](#) ... 15
[deleteFromDatabase](#) ... 5
[deleteProduct](#) ... 40
[deleteUser](#) ... 15
[deleteUser](#) ... 34
[deleteUser](#) ... 40
[documentation](#) ... 40
[documentationClient](#) ... 40
[documentationServer](#) ... 41

G

[getAllUsers](#) ... 15
[getAmount](#) ... 10
[getBarcode](#) ... 21
[getBills](#) ... 16
[getBills](#) ... 16
[getBills](#) ... 27
[getCertificat](#) ... 32
[getCss](#) ... 32
[getDatabase](#) ... 2
[getDate](#) ... 10
[getDate](#) ... 12
[getFile](#) ... 33
[getFile](#) ... 47
[getHash](#) ... 47
[getHistory](#) ... 11
[getID](#) ... 11
[getID](#) ... 12
[getID](#) ... 23
[getMail](#) ... 23
[getMailAddress](#) ... 3
[getMailPassword](#) ... 3
[getMoney](#) ... 23
[getName](#) ... 12
[getName](#) ... 21
[getName](#) ... 23
[getPassword](#) ... 3
[getPassword](#) ... 24
[getPort](#) ... 3
[getPreName](#) ... 24
[getPrice](#) ... 13
[getPrice](#) ... 21
[getServer](#) ... 3
[getUInfo](#) ... 16
[getUser](#) ... 4
[getUser](#) ... 11
[getUser](#) ... 13
[getUser](#) ... 16
[getUserHistory](#) ... 17
[getUserHistoryUnpaid](#) ... 17
[getUserInfo](#) ... 17
[getUserTimeout](#) ... 4

H

[History](#) ... 11
[History](#) ... 12

I

[insertToDatabase](#) ... 5
[isAdmin](#) ... 17
[isAdmin](#) ... 27
[isPaid](#) ... 13

L

[login](#) ... 28
[login](#) ... 41
[logout](#) ... 29
[logout](#) ... 41

M

[main](#) ... 41
[MailTransfer](#) ... 7
[MailTransfer](#) ... 7

O

[ObjectHandler](#) ... 13
[ObjectHandler](#) ... 14

P

[productAdministration](#) ... 42
[productByName](#) ... 34
[productExplicit](#) ... 34
[Product](#) ... 20
[Product](#) ... 20

R

[readFromDatabase](#) ... 5
[readFromDatabase](#) ... 6
[register](#) ... 29
[removeProduct](#) ... 18
[run](#) ... 44

S

[searchForProduct](#) ... 18
[searchUser](#) ... 18
[searchUserM](#) ... 35
[searchUserN](#) ... 35
[sendBill](#) ... 42
[sendMail](#) ... 8
[sendMail](#) ... 27
[setID](#) ... 24
[setMoney](#) ... 24
[start](#) ... 44
[Search](#) ... 33
[Search](#) ... 33
[STABill](#) ... 46
[STAHistory](#) ... 46
[STAProduct](#) ... 46
[STAUser](#) ... 46

T

[TimerThread](#) ... 43
[TimerThread](#) ... 43

U

[updateDatabase](#) ... 6
[updateProduct](#) ... 18
[userAdministration](#) ... 42
[userHistory](#) ... 35
[userLogin](#) ... 19
[userLogout](#) ... 19
[User](#) ... 21
[User](#) ... 22
[User](#) ... 22
[User](#) ... 22
[User](#) ... 23
[Util](#) ... 44
[Util](#) ... 44

W

[writeBill](#) ... 19
[writeHistory](#) ... 19
[writeProductToDB](#) ... 20
[Website](#) ... 36
[Website](#) ... 36