

# Table Of Content

---

<a href="#">Client</a> .....	3
<a href="#">MainClient</a> .....	3
<a href="#">Client.Cart</a> .....	6
<a href="#">CartController</a> .....	6
<a href="#">CartInfoController</a> .....	8
<a href="#">CheckoutController</a> .....	9
<a href="#">CheckoutController.TableInfo</a> .....	10
<a href="#">DefaultCartController</a> .....	11
<a href="#">DefaultCartController.TableInfo</a> .....	12
<a href="#">Client.Catalogue</a> .....	15
<a href="#">CatalogueController</a> .....	15
<a href="#">ProductInfoController</a> .....	16
<a href="#">Client.Login</a> .....	18
<a href="#">LoginFormController</a> .....	18
<a href="#">RegistrationFormController</a> .....	19
<a href="#">Client.Main</a> .....	21
<a href="#">MainFormController</a> .....	21
<a href="#">Client.Orders</a> .....	23
<a href="#">OrderInfoController</a> .....	23
<a href="#">OrderInfoController.TableInfo</a> .....	24
<a href="#">OrdersController</a> .....	27
<a href="#">Client.Reviews</a> .....	29
<a href="#">AddReviewController</a> .....	29
<a href="#">DefaultReviewsController</a> .....	30
<a href="#">ReviewInfoController</a> .....	32
<a href="#">ReviewsController</a> .....	33
<a href="#">Core</a> .....	35
<a href="#">Core</a> .....	35
<a href="#">Core.Cart</a> .....	40
<a href="#">Cart</a> .....	40
<a href="#">ProductInCart</a> .....	41
<a href="#">Core.Catalogue</a> .....	43
<a href="#">Catalogue</a> .....	43
<a href="#">Category</a> .....	44
<a href="#">Product</a> .....	46
<a href="#">Section</a> .....	49
<a href="#">Server</a> .....	52
<a href="#">Database</a> .....	52

<a href="#">MainServer</a> .....	55
<a href="#">Utils</a> .....	57
<a href="#">MessageBoxOKController</a> .....	57
<a href="#">Utils</a> .....	58
<a href="#">Utils.StreamsJSON</a> .....	64
<a href="#">JSONReader</a> .....	64
<a href="#">JSONWriter</a> .....	65
<a href="#">Index</a> .....	67

# Package Client

## Class Summary

### [MainClient](#)

Classe principale del Client.

#### Client

## Class MainClient

```
java.lang.Object
|
+--javafx.application.Application
|
+--Client.MainClient
```

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

```
public class MainClient
extends javafx.application.Application
```

Classe principale del Client. Contiene informazioni riguardo la socket, i manager per gli Stream in lettura e scrittura, L'utente che si e' collegato, ed il catalogo prodotti.

## Fields

### ConfigFile

```
public static org.w3c.dom.Document ConfigFile
```

### catalogue

```
public static Catalogue catalogue
```

### clientSocket

```
public static java.net.Socket clientSocket
```

### reader

```
public static JSONReader reader
```

## siteAddress

```
public static java.lang.String siteAddress
```

---

## user

```
public static Core.User.User user
```

---

## writer

```
public static JSONWriter writer
```

## Constructors

### MainClient

```
public MainClient()
```

## Methods

### UpdateOrders

```
public static void UpdateOrders()
```

Aggiorna gli ordini dell'utente, controllandone prima la corrispondenza nel numero.

---

## close

```
public static void close()
```

Chiude la connessione col server.

---

## main

```
public static void main(java.lang.String[] args)
```

Metodo Main del Client. Avvia l'applicazione.

#### Parameters:

args - Parametri passati all'esecuzione dell'applicazione.

---

## start

```
public void start(javafx.stage.Stage stage)
```

Metodo start, derivato da Application. Esegue un codice alla creazione della finestra.

**Parameters:**

stage - Finestra su cui disegnare la mia GUI.

**Overrides:**

start in class javafx.application.Application

# Package Client.Cart

## Class Summary

### [CartController](#)

Classe che implementa Initializable.

### [CartInfoController](#)

Classe che implementa Initializable.

### [CheckoutController](#)

Classe che implementa Initializable.

### [CheckoutController.TableInfo](#)

InnerClass: Utilizzata per definire un tipo da associare alle colonne della Tabella.

### [DefaultCartController](#)

Classe che implementa Initializable.

### [DefaultCartController.TableInfo](#)

InnerClass: Utilizzata per definire un tipo da associare alle colonne della Tabella.

---

## Client.Cart

# Class CartController

```
java.lang.Object
|
+--Client.Cart.CartController
```

### All Implemented Interfaces:

javafx.fxml.Initializable

---

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

---

```
public class CartController
extends java.lang.Object
implements javafx.fxml.Initializable
```

Classe che implementa Initializable. Rappresenta il carrello dell'utente.

## Constructors

# CartController

```
public CartController()
```

## Methods

### initialize

```
public void initialize(java.net.URL location,  
                      java.util.ResourceBundle resources)
```

Metodo derivato da Initializable. Inizializza l'albero.

**Parameters:**

location - Path relativa all'oggetto iniziale. Nullo se non esiste.

resources - Risorse utilizzare per risalire all'oggetto iniziale. Nullo se non esiste.

---

### initializeDefaultCartPage

```
public void initializeDefaultCartPage()
```

Inizializza la pagina di default del carrello, con un riepilogo dell'ordine, ed il bottone di checkout.

---

### loadDefaultCart

```
public void loadDefaultCart(CartController cartController)
```

Carica la pagina predefinita del carrello.

**Parameters:**

cartController - Il controller di questa pagina, per poter effettuare modifiche su di un suo panel.

---

### loadProductInCart

```
public void loadProductInCart(ProductInCart product,  
                             CartController cartController)
```

Carica i dettagli di un prodotto all'interno del carrello su di un Panel.

**Parameters:**

product - Prodotto del carrello.

cartController - Il controller di questa pagina, per poter effettuare modifiche su di un suo panel.

---

Client.Cart

# Class CartInfoController

```
java.lang.Object
|
+--Client.Cart.CartInfoController
```

**All Implemented Interfaces:**  
javafx.fxml.Initializable

---

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

---

```
public class CartInfoController
extends java.lang.Object
implements javafx.fxml.Initializable
```

Classe che implementa Initializable. Rappresenta i dettagli per un prodotto nel carrello.

## Constructors

### CartInfoController

```
public CartInfoController()
```

## Methods

### initialize

```
public void initialize(java.net.URL location,
                       java.util.ResourceBundle resources)
```

Metodo derivato da Initializable.

**Parameters:**

location - Path relativa all'oggetto iniziale. Nullo se non esiste.  
resources - Risorse utilizzare per risalire all'oggetto iniziale. Nullo se non esiste.

---

### initializePage

```
public void initializePage(ProductInCart productInfo,
                           CartController cartController)
```

Inizializza i dettagli riguardanti il prodotto acquistato.

**Parameters:**

productInfo - Le informazioni sul prodotto acquistato.  
cartController - Rappresenta il pannello corrente.



Client.Cart

# Class CheckoutController

```
java.lang.Object
|
+--Client.Cart.CheckoutController
```

**All Implemented Interfaces:**  
javafx.fxml.Initializable

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

```
public class CheckoutController
extends java.lang.Object
implements javafx.fxml.Initializable
```

Classe che implementa Initializable. Rappresenta la pagina di conferma del checkout del carrello.

## Constructors

### CheckoutController

```
public CheckoutController()
```

## Methods

### initialize

```
public void initialize(java.net.URL location,
                      java.util.ResourceBundle resources)
```

Metodo derivato da Initializable.

**Parameters:**

location - Path relativa all'oggetto iniziale. Nullo se non esiste.  
resources - Risorse utilizzare per risalire all'oggetto iniziale. Nullo se non esiste.

### initializePage

```
public void initializePage(CartController cartController)
```

Inizializza la pagina del checkout.

**Parameters:**

cartController - Rappresenta il pannello corrente, per eventuali modifiche.

Client.Cart

# Class CheckoutController.TableInfo

```
java.lang.Object
|
+--Client.Cart.CheckoutController.TableInfo
```

---

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

---

```
public class CheckoutController.TableInfo
extends java.lang.Object
```

InnerClass: Utilizzata per definire un tipo da associare alle colonne della Tabella. Contiene Getter & Setter.

## Constructors

### TableInfo

```
public TableInfo(java.lang.String courierName,
                  double cost,
                  java.lang.String expressTime)
```

## Methods

### getCost

```
public double getCost()
```

---

### getCourierName

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

---

### getExpressTime

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

---

## setCost

```
public void setCost(double cost)
```

---

## setCourierName

```
public void setCourierName(java.lang.String cCourierName)
```

---

## setExpressTime

```
public void setExpressTime(java.lang.String cExpressTime)
```

---

Client.Cart

# Class DefaultCartController

```
java.lang.Object
|
+--Client.Cart.DefaultCartController
```

### All Implemented Interfaces:

javafx.fxml.Initializable

---

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

---

```
public class DefaultCartController
extends java.lang.Object
implements javafx.fxml.Initializable
```

Classe che implementa Initializable. Rappresenta il riepilogo del carrello prima del checkout.

## Constructors

### DefaultCartController

```
public DefaultCartController()
```

## Methods

## initialize

```
public void initialize(java.net.URL location,  
                      java.util.ResourceBundle resources)
```

Metodo derivato da Initializable.

### Parameters:

location - Path relativa all'oggetto iniziale. Nullo se non esiste.

resources - Risorse utilizzare per risalire all'oggetto iniziale. Nullo se non esiste.

---

## initializePage

```
public void initializePage(javafx.scene.layout.Pane cart,  
                          CartController cartController)
```

Inizializza la pagina del riepilogo con i dati.

### Parameters:

cart - Panel su cui inserire gli elementi vari.

cartController - Controller per avere informazioni sul pannello corrente.

---

Client.Cart

## Class DefaultCartController.TableInfo

```
java.lang.Object  
|  
+--Client.Cart.DefaultCartController.TableInfo
```

---

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

---

```
public class DefaultCartController.TableInfo  
extends java.lang.Object
```

InnerClass: Utilizzata per definire un tipo da associare alle colonne della Tabella. Contiene Getter & Setter.

## Constructors

# TableInfo

```
public TableInfo(java.lang.String productName,  
                 int quantity,  
                 double productPrice,  
                 double discount,  
                 double total)
```

Costruttore TableInfo

## Parameters:

productName - Nome Prodotto  
quantity - Quantita'  
productPrice - Prezzo Prodotto  
discount - Percentuale Sconto  
total - Totale Prodotto

## Methods

### getDiscount

```
public double getDiscount()
```

---

### getProductName

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

---

### getProductPrice

```
public double getProductPrice()
```

---

### getQuantity

```
public int getQuantity()
```

---

### getTotal

```
public double getTotal()
```

---

## setDiscount

```
public void setDiscount(double discount)
```

---

## setProductName

```
public void setProductName(java.lang.String pName)
```

---

## setProductPrice

```
public void setProductPrice(double pPrice)
```

---

## setQuantity

```
public void setQuantity(int quantity)
```

---

## setTotal

```
public void setTotal(double total)
```

# Package Client.Catalogue

## Class Summary

### [CatalogueController](#)

Metodo che implementa Initializable.

### [ProductInfoController](#)

Classe che implementa Initializable.

## Client.Catalogue

# Class CatalogueController

```
java.lang.Object
|
+--Client.Catalogue.CatalogueController
```

### All Implemented Interfaces:

javafx.fxml.Initializable

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

```
public class CatalogueController
extends java.lang.Object
implements javafx.fxml.Initializable
```

Metodo che implementa Initializable. Rappresenta il catalogo prodotti dell'azienda.

## Constructors

## CatalogueController

```
public CatalogueController()
```

## Methods

## initialize

```
public void initialize(java.net.URL location,
                       java.util.ResourceBundle resources)
```

Metodo derivato da Initializable.

### Parameters:

location - Path relativa all'oggetto iniziale. Nulla se non esiste.  
resources - Risorse utilizzare per risalire all'oggetto iniziale. Nulla se non esiste.

---

## initializePage

```
public void initializePage()
```

Inizializza la pagina, caricando il catalogo.

---

Client.Catalogue

## Class ProductInfoController

```
java.lang.Object
|
+--Client.Catalogue.ProductInfoController
```

### All Implemented Interfaces:

javafx.fxml.Initializable

---

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

---

```
public class ProductInfoController
extends java.lang.Object
implements javafx.fxml.Initializable
```

Classe che implementa Initializable. Rappresenta le informazioni di un singolo prodotto nel catalogo.

## Constructors

### ProductInfoController

```
public ProductInfoController()
```

## Methods

### initialize

```
public void initialize(java.net.URL url,
                       java.util.ResourceBundle rb)
```

Metodo derivato da Initializable. Imposta l'evento alla TextField per la quantita'.

#### Parameters:

url - Path relativa all'oggetto iniziale. Nullo se non esiste.

rb - Risorse utilizzare per risalire all'oggetto iniziale. Nullo se non esiste.

---



## setProduct

```
public void setProduct(Product productInfo)
```

Inizializza il prodotto selezionato.

### Parameters:

productInfo -

# Package Client.Login

## Class Summary

### [LoginFormController](#)

Classe che implementa Initiaizable.

### [RegistrationFormController](#)

Classe che implementa Initializable.

## Client.Login

# Class LoginFormController

```
java.lang.Object
|
+--Client.Login.LoginFormController
```

### All Implemented Interfaces:

javafx.fxml.Initializable

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

```
public class LoginFormController
extends java.lang.Object
implements javafx.fxml.Initializable
```

Classe che implementa Initiaizable. Rappresenta la finestra di Login.

## Constructors

## LoginFormController

```
public LoginFormController()
```

## Methods

## initialize

```
public void initialize(java.net.URL url,
                       java.util.ResourceBundle rb)
```

Metodo derivato da Initializable.

### Parameters:

url - Path relativa all'oggetto iniziale. Nulla se non esiste.

rb - Risorse utilizzare per risalire all'oggetto iniziale. Nulla se non esiste.

---

## setXML

```
public void setXML(org.w3c.dom.Document document)
```

Imposta il file di configurazione XML da cui leggere i dati.

**Parameters:**

document - Dati XML.

---

### Client.Login

## Class RegistrationFormController

```
java.lang.Object  
|  
+--Client.Login.RegistrationFormController
```

**All Implemented Interfaces:**

javafx.fxml.Initializable

---

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

---

```
public class RegistrationFormController  
extends java.lang.Object  
implements javafx.fxml.Initializable
```

Classe che implementa Initializable. Rappresenta la finestra di registrazione.

## Constructors

### RegistrationFormController

```
public RegistrationFormController()
```

## Methods

### initialize

```
public void initialize(java.net.URL url,  
                      java.util.ResourceBundle rb)
```

Metodo derivato da Initializable. Inizializza le combobox con i valori delle date.

**Parameters:**

url - Path relativa all'oggetto iniziale. Nulla se non esiste.

rb - Risorse utilizzare per risalire all'oggetto iniziale. Nulla se non esiste.

---

## setXML

```
public void setXML(org.w3c.dom.Document document)
```

Imposta il file di configurazione XML da cui leggere i dati.

**Parameters:**

document - Dati XML.

# Package Client.Main

## Class Summary

### [MainFormController](#)

Classe che implementa Initializable.

Client.Main

## Class MainFormController

```
java.lang.Object
|
+--Client.Main.MainFormController
```

### All Implemented Interfaces:

javafx.fxml.Initializable

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

```
public class MainFormController
extends java.lang.Object
implements javafx.fxml.Initializable
```

Classe che implementa Initializable. Rappresenta la finestra principale.

## Constructors

### MainFormController

```
public MainFormController()
```

## Methods

### initialize

```
public void initialize(java.net.URL url,
                       java.util.ResourceBundle rb)
```

Metodo derivato da Initializable. Inizializza le diverse funzionalità disponibili con l'applicazione.

#### Parameters:

url - Path relativa all'oggetto iniziale. Nullo se non esiste.

rb - Risorse utilizzare per risalire all'oggetto iniziale. Nullo se non esiste.

## **initializeCartPage**

```
public void initializeCartPage()
```

Inizializza il Panel del carrello.

---

## **initializeCataloguePage**

```
public void initializeCataloguePage()
```

Inizializza il Panel del catalogo.

---

## **initializeOrdersPage**

```
public void initializeOrdersPage()
```

Inizializza il Panel degli ordini.

# Package Client.Orders

## Class Summary

### [OrderInfoController](#)

Classe che implementa Initializable.

### [OrderInfoController.TableInfo](#)

InnerClass: Utilizzata per definire un tipo da associare alle colonne della Tabella.

### [OrdersController](#)

Classe che implementa Initializable.

## Client.Orders

# Class OrderInfoController

```
java.lang.Object
|
+--Client.Orders.OrderInfoController
```

### All Implemented Interfaces:

javafx.fxml.Initializable

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

```
public class OrderInfoController
extends java.lang.Object
implements javafx.fxml.Initializable
```

Classe che implementa Initializable. Rappresenta le informazioni di un ordine.

## Constructors

## OrderInfoController

```
public OrderInfoController()
```

## Methods

## initialize

```
public void initialize(java.net.URL location,  
                      java.util.ResourceBundle resources)
```

Metodo derivato da Initializable.

### Parameters:

location - Path relativa all'oggetto iniziale. Nullo se non esiste.

resources - Risorse utilizzare per risalire all'oggetto iniziale. Nullo se non esiste.

---

## initializePage

```
public void initializePage(Core.Order.Order order)
```

Metodo per inizializzare la pagina.

### Parameters:

order - Ordine da rappresentare.

---

### Client.Orders

## Class OrderInfoController.TableInfo

```
java.lang.Object  
|  
+--Client.Orders.OrderInfoController.TableInfo
```

---

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

---

```
public class OrderInfoController.TableInfo  
extends java.lang.Object
```

InnerClass: Utilizzata per definire un tipo da associare alle colonne della Tabella. Contiene Getter & Setter.

---

## Constructors



# TableInfo

```
public TableInfo(java.lang.String productName,  
                 int quantity,  
                 double total,  
                 java.lang.String status,  
                 java.lang.String startDate,  
                 java.lang.String goalDate)
```

Costruttore TableInfo.

## Parameters:

productName - Nome Prodotto.  
quantity - Quantita'.  
total - Totale.  
status - Stato.  
startDate - Data di partenza.  
goalDate - Data di arrivo.

## Methods

### getGoalDate

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

---

### getProductName

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

---

### getQuantity

```
public int getQuantity()
```

---

### getStartDate

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

---

## getStatus

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

---

## getTotal

```
public double getTotal()
```

---

## setGoalDate

```
public void setGoalDate(java.lang.String oGoalDate)
```

---

## setProductName

```
public void setProductName(java.lang.String oProductName)
```

---

## setQuantity

```
public void setQuantity(int oQuantity)
```

---

## setStartDate

```
public void setStartDate(java.lang.String oStartDate)
```

---

## setStatus

```
public void setStatus(java.lang.String oStatus)
```

---

## setTotal

```
public void setTotal(double oTotal)
```

---

Client.Orders

# Class OrdersController

```
java.lang.Object
|
+--Client.Orders.OrdersController
```

## All Implemented Interfaces:

javafx.fxml.Initializable

---

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

---

```
public class OrdersController
extends java.lang.Object
implements javafx.fxml.Initializable
```

Classe che implementa Initializable. Rappresenta le informazioni riguardo agli ordini che sono stati effettuati dall'utente in passato.

## Constructors

### OrdersController

```
public OrdersController()
```

## Methods

### initialize

```
public void initialize(java.net.URL location,
                       java.util.ResourceBundle resources)
```

Metodo derivato da Initializable. Inizializza l'evento della ListView per la selezione di un file.

#### Parameters:

location - Path relativa all'oggetto iniziale. Nullo se non esiste.

resources - Risorse utilizzare per risalire all'oggetto iniziale. Nullo se non esiste.

---

### initializePage

```
public void initializePage()
```

Inizializza la pagina, ricaricando gli ordini, e caricandoli nella lista.

---

## loadOrderInfo

```
public void loadOrderInfo(Core.Order.Order order)
```

Viene richiamato se viene selezionato un elemento della ListView. Carica le informazioni su di un singolo ordine nel Panel.

**Parameters:**

order - Informazioni sull'ordine interessato.

# Package Client.Reviews

## Class Summary

### [AddReviewController](#)

Classe che implementa Initializable.

### [DefaultReviewsController](#)

Classe che implementa Initializable.

### [ReviewInfoController](#)

Classe che implementa Initializable.

### [ReviewsController](#)

Classe che implementa Initializable.

---

## Client.Reviews

# Class AddReviewController

```
java.lang.Object
|
+--Client.Reviews.AddReviewController
```

### All Implemented Interfaces:

javafx.fxml.Initializable

---

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

---

```
public class AddReviewController
extends java.lang.Object
implements javafx.fxml.Initializable
```

Classe che implementa Initializable. Rappresenta la finestra per aggiungere una recensione.

## Constructors

### AddReviewController

```
public AddReviewController()
```

## Methods

## initialize

```
public void initialize(java.net.URL location,  
                      java.util.ResourceBundle resources)
```

Metodo derivato da Initializable. Inizializza la combobox coi voti.

### Parameters:

location - Path relativa all'oggetto iniziale. Nullo se non esiste.

resources - Risorse utilizzare per risalire all'oggetto iniziale. Nullo se non esiste.

---

## initializePage

```
public void initializePage(int purchaseID,  
                          Product product)
```

Inizializza la pagina.

### Parameters:

purchaseID -

product -

---

## Client.Reviews

# Class DefaultReviewsController

```
java.lang.Object  
|  
+--Client.Reviews.DefaultReviewsController
```

### All Implemented Interfaces:

javafx.fxml.Initializable

---

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

---

```
public class DefaultReviewsController  
extends java.lang.Object  
implements javafx.fxml.Initializable
```

Classe che implementa Initializable. Rappresenta la pagina di riassunto di tutte le recensioni di un prodotto.

## Constructors

## DefaultReviewsController

```
public DefaultReviewsController()
```

## Methods

### initialize

```
public void initialize(java.net.URL location,  
                      java.util.ResourceBundle resources)
```

Metodo derivato da Initializable. Inizializza l'albero.

**Parameters:**

location - Path relativa all'oggetto iniziale. Nullo se non esiste.

resources - Risorse utilizzare per risalire all'oggetto iniziale. Nullo se non esiste.

---

### initializePanel

```
public void initializePanel()
```

Inizializza il pannello con i dati generali sulle recensioni.

---

### setProductInfo

```
public void setProductInfo(Product product)
```

Imposta il prodotto di cui si devono visualizzare le recensioni.

**Parameters:**

product - Prodotto selezionato.

---

### setReviews

```
public void setReviews(java.util.ArrayList reviews)
```

Imposta le recensioni da visualizzare.

**Parameters:**

reviews - Tutte le recensioni su quel prodotto.

---

Client.Reviews

# Class ReviewInfoController

```
java.lang.Object
|
+--Client.Reviews.ReviewInfoController
```

**All Implemented Interfaces:**  
javafx.fxml.Initializable

---

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

---

```
public class ReviewInfoController
extends java.lang.Object
implements javafx.fxml.Initializable
```

Classe che implementa Initializable. Rappresenta le informazioni di una singola recensione.

## Constructors

### ReviewInfoController

```
public ReviewInfoController()
```

## Methods

### initialize

```
public void initialize(java.net.URL location,
                      java.util.ResourceBundle resources)
```

Metodo derivato da Initializable.

**Parameters:**

location - Path relativa all'oggetto iniziale. Nullo se non esiste.  
resources - Risorse utilizzare per risalire all'oggetto iniziale. Nullo se non esiste.

---

### setReview

```
public void setReview(Core.Review.Review review)
```

Imposta la pagina con i dati della recensione.

**Parameters:**

review - Recensione da visualizzare.

---



Client.Reviews

# Class ReviewsController

```
java.lang.Object
|
+--Client.Reviews.ReviewsController
```

## All Implemented Interfaces:

javafx.fxml.Initializable

---

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

---

```
public class ReviewsController
extends java.lang.Object
implements javafx.fxml.Initializable
```

Classe che implementa Initializable. Rappresenta le recensioni di un prodotto.

## Constructors

### ReviewsController

```
public ReviewsController()
```

## Methods

### initialize

```
public void initialize(java.net.URL location,
                      java.util.ResourceBundle resources)
```

Metodo derivato da Initializable. Inizializza l'albero.

#### Parameters:

location - Path relativa all'oggetto iniziale. Nullo se non esiste.

resources - Risorse utilizzare per risalire all'oggetto iniziale. Nullo se non esiste.

---

### loadDefaultForm

```
public void loadDefaultForm()
```

Carica la pagina di riassunto su tutte le recensioni, e per aggiungerne / rimuoverne una.

---

## loadReview

```
public void loadReview(Core.Review.Review review)
```

Carica la recensione scelta sul pannello.

**Parameters:**

review - Recensione da voler caricare.

---

## loadReviews

```
public void loadReviews()
```

Carica le recensioni facendo la richiesta al server.

---

## setProductInfo

```
public void setProductInfo(Product product)
```

Imposta il prodotto di cui si vogliono visualizzare le recensioni.

**Parameters:**

product - Prodotto da visualizzare.

# Package Core

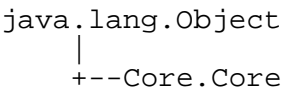
## Class Summary

### [Core](#)

Classe principale.

### Core

## Class Core



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

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

Classe principale. Contiene le varie funzioni che comunicano con il database.

## Constructors

### Core

```
public Core()
```

## Methods

## AddReview

```
public static JsonObject AddReview(java.lang.String title,  
                                     java.lang.String text,  
                                     int vote,  
                                     int purchaseID,  
                                     Database dbConnection)
```

Richiesta per aggiungere una nuova recensione.

### Parameters:

title - Titolo della recensione  
text - Testo della recensione.  
vote - Voto della recensione.  
purchaseID - ID dell'acquisto.  
dbConnection - Riferimento all'oggetto del database creato dal server.

### Returns:

---

## CheckNumberOfOrders

```
public static JsonObject CheckNumberOfOrders(int userID,  
                                              Database dbConnection)
```

Controlla il numero di ordini di un dato utente.

### Parameters:

userID - ID Utente.  
dbConnection - Riferimento all'oggetto del database creato dal server.

### Returns:

Risultato della richiesta ed il numero di ordini-

---

## ConfirmCheckout

```
public static JsonObject ConfirmCheckout(Cart cart,  
                                           Core.Courier.Courier courier,  
                                           Core.User.User user,  
                                           Database dbConnection)
```

Effettua il checkout del carrello.

### Parameters:

cart - I prodotti nel carrello.  
courier - Corriere scelto per la spedizione.  
user - L'utente che ha effettuato l'acquisto.  
dbConnection - Riferimento all'oggetto del database creato dal server.

### Returns:

Ritorna il risultato della richiesta.

---

## LoadCatalogue

```
public static JsonObject LoadCatalogue(Database dbConnection)
```

Carica il catalogo prodotti del sito di ECommerce.

**Parameters:**

dbConnection - Riferimento all'oggetto del database creato dal server.

**Returns:**

Oggetto contenente il risultato, ed il catalogo da visualizzare.

---

## LoadCouriers

```
public static JsonObject LoadCouriers(Database dbConnection)
```

Richiesta per caricare i corrieri disponibili per la spedizione degli acquisti.

**Parameters:**

dbConnection - Riferimento all'oggetto del database creato dal server.

**Returns:**

Il risultato della richiesta, ed i corrieri trovati.

---

## LoadOrders

```
public static JsonObject LoadOrders(int userID,  
                                     Database dbConnection)
```

Carica gli ordini di un dato utente.

**Parameters:**

userID - ID dell'utente.

dbConnection - Riferimento all'oggetto del database creato dal server.

**Returns:**

Risultato della richiesta, con tutti gli ordini trovati.

---

## LoadReviews

```
public static JsonObject LoadReviews(Database dbConnection,  
                                     int productID)
```

Carica le recensioni di un dato prodotto.

**Parameters:**

dbConnection - Riferimento all'oggetto del database creato dal server.

productID - ID del prodotto di cui si vogliono le recensioni

**Returns:**

Recensioni trovate su di quel prodotto.

---

## RemoveReview

```
public static JsonObject RemoveReview(int purchaseID,  
                                         Database dbConnection)
```

Richiesta per rimuovere una recensione dal database.

**Parameters:**

purchaseID - ID dell'acquisto.

dbConnection - Riferimento all'oggetto del database creato dal server.

**Returns:**

Risultato sulla rimozione, se e' avvenuta o meno.

---

## checkCredentials

```
public static JsonObject checkCredentials(java.lang.String username,  
                                             java.lang.String password,  
                                             Database dbConnection)
```

Controlla le credenziali inserite dall'utente nel form di login.

**Parameters:**

username - Username

password - Password

dbConnection - Riferimento all'oggetto del database creato dal server.

**Returns:**

Un oggetto contenente il risultato della richiesta, e le informazioni dell'utente.

---

## loadDiscount

```
public static double loadDiscount(Database dbConnection,  
                                   int productID)
```

Controlla lo sconto sul prodotto richiesto.

**Parameters:**

dbConnection - Riferimento all'oggetto del database creato dal server.

productID - ID del prodotto su cui controllare se esiste uno sconto.

**Returns:**

Ritorna la percentuale di sconto del prodotto.

---

# registerUser

```
public static JsonObject registerUser(java.lang.String username,  
                                       java.lang.String password,  
                                       java.lang.String email,  
                                       java.lang.String name,  
                                       java.lang.String surname,  
                                       java.lang.String address,  
                                       java.lang.String city,  
                                       java.lang.String cap,  
                                       java.lang.String telephone,  
                                       java.lang.String identitycode,  
                                       java.lang.String borndate,  
                                       Database dbConnection)
```

Effettua la registrazione di un nuovo utente.

## Parameters:

username - Username  
password - Password  
email - Email  
name - Nome  
surname - Cognome  
address - Indirizzo  
city - Citta'  
cap - CAP  
telephone - Numero di Telefono  
identitycode - Codice Fiscale  
borndate - Data di Nascita  
dbConnection - Riferimento all'oggetto del database creato dal server.

## Returns:

Un oggetto contenente il risultato della richiesta.

# Package Core.Cart

## Class Summary

### [Cart](#)

Classe contenente le informaioni sul carrello.

### [ProductInCart](#)

Created by david on 26/10/2015.

---

Core.Cart

## Class Cart

```
java.lang.Object
|
+--Core.Cart.Cart
```

---

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

---

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

Classe contenente le informaioni sul carrello.

## Fields

### productInCarts

```
public java.util.ArrayList productInCarts
```

## Constructors

### Cart

```
public Cart()
```

Costruttore Cart. Inizializza i prodotti nel carrello ad una lista vuota.

## Methods



## addToCart

```
public boolean addToCart(Product prod,  
                        int quantity)
```

Aggiunge un prodotto al carrello.

**Parameters:**

prod - Prodotto da inserire nel carrello.

quantity - Quantita' da voler acquistare.

**Returns:**

Risultato dell'inserimento nel carrello, potrebbe essere gia' presente.

---

## removeFromCart

```
public boolean removeFromCart(int productID)
```

Rimuove un prodotto dal carrello.

**Parameters:**

productID - ID del prodotto da voler rimuovere.

**Returns:**

Risultato della rimozione.

---

## updateQuantity

```
public boolean updateQuantity(int productID,  
                             int newValue)
```

Aggiorna la quantita' di un prodotto nel carrello.

**Parameters:**

productID - ID del prodotto di cui si vuole aggiornare la quantita'.

newValue - Nuovo valore del prodotto nel carrello.

**Returns:**

Se e' stato aggiornato con successo, ritorna True, altrimenti False.

---

Core.Cart

## Class ProductInCart

```
java.lang.Object  
|  
+--Core.Cart.ProductInCart
```

---

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

---

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

Created by david on 26/10/2015.

## Fields

### product

```
public Product product
```

### quantity

```
public int quantity
```

## Constructors

### ProductInCart

```
public ProductInCart(Product product,
                      int quantity)
```

## Methods

### CheckIfAvailable

```
public boolean CheckIfAvailable(int newQuantity)
```

### toString

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

Effettuo l'override del metodo toString della classe Object.

**Returns:**

Nome del prodotto.

**Overrides:**

toString in class java.lang.Object

# Package Core.Catalogue

## Class Summary

### [Catalogue](#)

Classe contenente le informazioni sul catalogo.

### [Category](#)

Classe contenente le informazioni su una categoria.

### [Product](#)

Classe contenente le informazioni di un prodotto.

### [Section](#)

Classe contenente le informazioni su una sezione del catalogo.

---

## Core.Catalogue

# Class Catalogue

```
java.lang.Object
|
+--Core.Catalogue.Catalogue
```

---

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

---

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

Classe contenente le informazioni sul catalogo. Contiene Getter & Setter.

## Constructors

### Catalogue

```
public Catalogue()
```

Costruttore Catalogue.

## Methods

## addSection

```
public void addSection(Section section)
```

Aggiunge una sezione al catalogo.

**Parameters:**

section - Sezione da aggiungere.

---

## getCatalogue

```
public java.util.ArrayList getCatalogue()
```

---

## hasSections

```
public boolean hasSections()
```

Controlla se esistono sezioni.

**Returns:**

Se ci sono sezioni, ritorna True, False altrimenti.

---

## setCatalogue

```
public void setCatalogue(java.util.ArrayList sections)
```

---

### Core.Catalogue

## Class Category

```
java.lang.Object  
|  
+--Core.Catalogue.Category
```

---

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

---

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

Classe contenente le informazioni su una categoria. Contiene Getter & Setter.

## Fields

## products

```
public java.util.ArrayList products
```

## Constructors

### Category

```
public Category(int id,  
                java.lang.String name,  
                java.lang.String desc)
```

Costruttore Category.

**Parameters:**

id - ID Categoria.

name - Nome Categoria.

desc - Descrizione Categoria.

## Methods

### getCategoryDesc

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

---

### getCategoryID

```
public int getCategoryID()
```

---

### getCategoryName

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

---

### hasProducts

```
public java.lang.Boolean hasProducts()
```

Controlla se ha prodotti.

**Returns:**

True se ci sono prodotti, False altrimenti.

---

## setProduct

```
public void setProduct(int idProd,  
                        java.lang.String nameProd,  
                        java.lang.String descProd,  
                        double priceProd,  
                        double discountProd,  
                        int quantityProd,  
                        java.lang.String imageProd,  
                        java.lang.String categoryName)
```

Aggiunge un prodotto alla categoria.

### Parameters:

idProd - ID Prodotto.  
nameProd - Nome Prodotto.  
descProd - Descrizione Prodotto.  
priceProd - Prezzo Prodotto.  
discountProd - Sconto Prodotto.  
quantityProd - Quantita' disponibile Prodotto.  
imageProd - Immagine del prodotto.  
categoryName - Nome della categoria che lo contiene.

---

## toString

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

Effettuo l'override del metodo toString della classe Object.

### Returns:

Nome della categoria.

### Overrides:

toString in class java.lang.Object

---

## Core.Catalogue

# Class Product

```
java.lang.Object  
|  
+--Core.Catalogue.Product
```

---

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

---

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

Classe contenente le informazioni di un prodotto. Contiene Getter & Setter.

# Constructors

## Product

```
public Product(int id,  
               java.lang.String name,  
               java.lang.String desc,  
               double price,  
               double discountPercentage,  
               int quantity,  
               java.lang.String imageProduct,  
               java.lang.String categoryName)
```

Costruttore Product.

### Parameters:

id - ID Prodotto.  
name - Nome Prodotto.  
desc - Descrizione Prodotto.  
price - Prezzo Prodotto.  
discountPercentage - Percentuale di Sconto.  
quantity - Quantità Disponibile del Prodotto.  
imageProduct - Immagine Prodotto.  
categoryName - Nome della categoria in cui è contenuto.

## Methods

### getCategoryName

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

---

### getProductDesc

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

---

### getProductDiscount

```
public double getProductDiscount()
```

---

## getProductDiscountPercentage

```
public double getProductDiscountPercentage()
```

---

## getProductID

```
public int getProductID()
```

---

## getProductImage

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

---

## getProductName

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

---

## getProductPrice

```
public double getProductPrice()
```

---

## getProductQuantity

```
public int getProductQuantity()
```

---

## hasDiscount

```
public boolean hasDiscount()
```

---

## setQuantity

```
public void setQuantity(int newValue)
```



---

## toString

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

Effettuo l'override del metodo toString della classe Object.

### Returns:

Nome del prodotto.

### Overrides:

toString in class java.lang.Object

---

## Core.Catalogue

# Class Section

```
java.lang.Object
|
+--Core.Catalogue.Section
```

---

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

---

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

Classe contenente le informazioni su una sezione del catalogo. Contiene Getter & Setter

## Fields

## categories

```
public java.util.ArrayList categories
```

## Constructors

## Section

```
public Section(int id,
               java.lang.String name,
               java.lang.String desc)
```

Costruttore Section

### Parameters:

id - ID Sezione

name - Nome Sezione

desc - Descrizione Sezione

## Methods

### getSectionDesc

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

---

### getSectionID

```
public int getSectionID()
```

---

### getSectionName

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

---

### hasCategories

```
public java.lang.Boolean hasCategories()
```

Controlla se la sezione ha categorie.

**Returns:**

True se la sezione ha categorie, False altrimenti.

---

### setCategory

```
public void setCategory(int idCat,  
                        java.lang.String nameCat,  
                        java.lang.String descCat)
```

Aggiunge una nuova categoria alla sezione.

**Parameters:**

idCat - ID Categoria

nameCat - Nome Categoria

descCat - Descrizione Categoria

---

# toString

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

Effettuo l'override del metodo toString della classe Object.

**Returns:**

Nome della sezione.

**Overrides:**

toString in class java.lang.Object

# Package Server

## Class Summary

### [Database](#)

Classe per la comunicazione con la base di dati.

### [MainServer](#)

Classe col metodo Main per il server.

---

## Server

## Class Database

```
java.lang.Object
|
+--Server.Database
```

---

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

---

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

Classe per la comunicazione con la base di dati.

## Fields

### connection

```
public java.sql.Connection connection
```

## Constructors

### Database

```
public Database(java.lang.String IP,
                java.lang.String Port,
                java.lang.String dbName,
                java.lang.String dbUser,
                java.lang.String dbPaswd)
```

Costruttore Database. Viene inizializzato il driver JDBC al suo interno.

## Methods

## CheckIfExistQuery

```
public boolean CheckIfExistQuery(java.lang.String query)
```

Controlla se esiste un elemento.

**Parameters:**

query - Query di selezione.

**Returns:**

Se esiste ritornera' True, altrimenti False.

---

## Close

```
public void Close()
```

Chiude lo statement ed il ResultSet.

---

## CloseResultSet

```
public void CloseResultSet()
```

Chiude il ResultSet.

---

## CloseStatement

```
public void CloseStatement()
```

Chiude lo statement.

---

## Commit

```
public void Commit()
```

Effettua il commit dei dati al database.

---

## GetLastInsertId

```
public long GetLastInsertId()
```

Richiesta dell'ultimo ID che e' stato inserito nel database.

**Returns:**

L'ultimo ID inserito.

---

## Initialize

```
public void Initialize()
```

Inizializza lo statement ed il ResultSet.

---

## InitializeResultSet

```
public void InitializeResultSet()
```

Inizializza il ResultSet.

---

## InitializeStatement

```
public void InitializeStatement()
```

Inizializza uno statement.

---

## InsertQuery

```
public boolean InsertQuery(java.lang.String query)
```

Effettua un inserimento nel database.

**Parameters:**

query - Query di inserimento.

**Returns:**

Risultato dell'inserimento, se e' 0 vuol dire che è fallito.

---

## RemoveQuery

```
public boolean RemoveQuery(java.lang.String query)
```

Effettua una rimozione dal database.

**Parameters:**

query - Query di rimozione.

**Returns:**

Se la rimozione e' avvenuta correttamente, ritornera' un numero diverso da 0.

---

## Rollback

```
public void Rollback()
```

Disabilita le modifiche effettuate al database.

---

## SelectionQuery

```
public java.sql.ResultSet SelectionQuery(java.lang.String query)
```

Effettua una query di selezione nel database.

### Parameters:

query - Query di selezione.

### Returns:

ResultSet con le informazioni.

---

### Server

## Class MainServer

```
java.lang.Object  
|  
+--Server.MainServer
```

---

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

---

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

Classe col metodo Main per il server.

## Constructors

### MainServer

```
public MainServer()
```

## Methods

# main

```
public static void main(java.lang.String[] args)
```

Metodo Main per il server.

**Parameters:**

args - Contiene eventuali informazione passate come parametro all'avvio dell'applicazione.



# Package Utils

## Class Summary

### [MessageBoxOKController](#)

Implementa l'interfaccia di JavaFX.

### [Utils](#)

Classe con diverse funzioni utilizzate nel progetto.

---

#### Utils

## Class MessageBoxOKController

```
java.lang.Object
|
+--Utils.MessageBoxOKController
```

#### All Implemented Interfaces:

javafx.fxml.Initializable

---

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

---

```
public class MessageBoxOKController
extends java.lang.Object
implements javafx.fxml.Initializable
```

Implementa l'interfaccia di JavaFX. Crea una finestra dinamica, che si ridimensiona a seconda del testo contenuto.

## Constructors

### MessageBoxOKController

```
public MessageBoxOKController()
```

## Methods

## initialize

```
public void initialize(java.net.URL location,  
                      java.util.ResourceBundle resources)
```

Override dell'interfaccia Initializable. Viene eseguito alla creazione della finestra.

### Parameters:

location - Path relativa all'oggetto iniziale. Nullo se non esiste.

resources - Risorse utilizzare per risalire all'oggetto iniziale. Nullo se non esiste.

---

## setMessage

```
public double setMessage(java.lang.String text)
```

Imposta il messaggio che verra' mostrato nella finestra.

### Parameters:

text - Il messaggio da mostrare.

### Returns:

La nuova grandezza della finestra.

---

## setOldStage

```
public void setOldStage(javafx.stage.Stage oldStage)
```

Se passato uno stage, viene impostato sull'oggetto.

### Parameters:

oldStage - Lo stage di una finestra secondaria.

---

## Utils

# Class Utils

```
java.lang.Object  
|  
+--Utils.Utils
```

---

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

---

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

Classe con diverse funzioni utilizzate nel progetto.

# Constructors

## Utils

```
public Utils()
```

Costruttore Utils

## Methods

### AddElement

```
public static void AddElement(org.w3c.dom.Element parent,  
                                org.w3c.dom.Element children)
```

Inserisce un elemento figlio dentro il padre.

**Parameters:**

parent - Elemento padre.  
children - Elemento figlio.

---

### CheckAddressString

```
public boolean CheckAddressString(java.lang.String address)
```

Controlla i caratteri all'interno di un indirizzo.

**Parameters:**

address - Indirizzo da controllare.

**Returns:**

True se e' valido, False altrimenti.

---

### CheckBornDate

```
public boolean CheckBornDate(java.lang.String day,  
                               java.lang.String month,  
                               java.lang.String year)
```

Controlla la data di nascita.

**Parameters:**

day - Giorno  
month - Mese  
year - Anno

**Returns:**

Vero se e' una data valida, False altrimenti.

---

## CheckCity

```
public boolean CheckCity(java.lang.String city)
```

Controlla se i caratteri della citta' sono validi.

**Parameters:**

city - Citta' da controllare.

**Returns:**

True se e' valida, False altrimenti.

---

## CheckIdentityCode

```
public boolean CheckIdentityCode(java.lang.String cf)
```

Controlla il codice fiscale, che non abbia caratteri vietati e che sia di lunghezza 16 caratteri.

**Parameters:**

cf - Il codice fiscale da controllare.

**Returns:**

True se e' valido. False altrimenti.

---

## CheckInteger

```
public boolean CheckInteger(java.lang.String number)
```

Controlla se e' una stringa.

**Parameters:**

number - Numero da controllare.

**Returns:**

True se e' numero, False altrimenti.

---

## CheckString

```
public boolean CheckString(java.lang.String text)
```

Controlla se la stringa contiene certi caratteri.

**Parameters:**

text - Messaggio da controllare

**Returns:**

True se la stringa e' corretta, False se errata.

---

## CheckTelephoneNumber

```
public boolean CheckTelephoneNumber(java.lang.String telephone)
```

Controlla il numero di telefono

**Parameters:**

telephone - Il numero da controllare.

**Returns:**

True se e' valido. False altrimenti.

---

## CreateFolder

```
public static void CreateFolder(java.lang.String folderName)
```

Crea la cartella se non esiste.

**Parameters:**

folderName - Nome della cartella.

---

## CreateXML

```
public static void CreateXML(org.w3c.dom.Document document,  
                               java.lang.String fileName)
```

Crea un file XML.

**Parameters:**

document - Informazioni XML da scrivere.

fileName - Nome del file.

---

## InitializeXML

```
public static org.w3c.dom.Document InitializeXML(java.lang.String rootText)
```

Inizializza un nuovo documento XML

**Parameters:**

rootText - Testo all'interno del nodo di Root

**Returns:**

Il documento XML su cui inserire gli elementi

---

## MessageBox

```
public void MessageBox(java.lang.String text)
```

Metodo per creare una MessageBox.

**Parameters:**

text - Messaggio da visualizzare.

---

## MessageBox

```
public void MessageBox(java.lang.String text,  
                        javafx.stage.Stage oldStage)
```

Metodo per creare una MessageBox.

**Parameters:**

text - Messaggio da visualizzare.

oldStage - Finestra precedente da chiudere.

---

## ReadXML

```
public static org.w3c.dom.Document ReadXML(java.lang.String path)
```

Legge il file XML.

**Parameters:**

path - Path al file da leggere.

**Returns:**

Il documento caricato.

---

## ValidateEmail

```
public boolean ValidateEmail(java.lang.String email)
```

Controlla se l'email e' corretta.

**Parameters:**

email - Email da controllare.

**Returns:**

True se e' valida, False altrimenti.

---

## convertDate

```
public static java.lang.String convertDate(java.util.Date date)
```

Converte la data dal tipo Date a Stringa

**Parameters:**

date - Data da convertire

**Returns:**

Data convertita

---

## fromMySQLDate

```
public static java.lang.String fromMySQLDate(java.lang.String date)
```

Converte la data da quella di MySQL, con layout YYYY-MM-GG, al layout GG-MM-YYYY.

**Parameters:**

date - Data con layout errato.

**Returns:**

Data con layout corretto.

---

## toMySQLDate

```
public static java.lang.String toMySQLDate(java.lang.String date)
```

Converte la data da quella con layout GG-MM-YYYY a quella con layout YYYY-MM-GG per MySQL.

**Parameters:**

date - Data con layout errato.

**Returns:**

Data con layout corretto.

# Package Utils.StreamsJSON

## Class Summary

### [JSONReader](#)

Estende da Buffered Reader, viene utilizzato per leggere dal flusso dati i dati JSON, e convertiti in JsonObject.

### [JSONWriter](#)

Estende da OutputStreamWriter, permette di scrivere un oggetto JSON sul flusso dati della socket.

## Utils.StreamsJSON

## Class JSONReader

```
java.lang.Object
|
+-- java.io.Reader
|   |
|   +-- java.io.BufferedReader
|       |
|       +-- Utils.StreamsJSON.JSONReader
```

### All Implemented Interfaces:

java.io.Closeable, java.lang.Readable

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

```
public class JSONReader
extends java.io.BufferedReader
```

Estende da Buffered Reader, viene utilizzato per leggere dal flusso dati i dati JSON, e convertiti in JsonObject.

## Constructors

### JSONReader

```
public JSONReader(java.io.InputStream in)
```

Cosruttore JSONReader.

#### Parameters:

in - Flusso dati in ingresso.

## Methods



## readJson

```
public JsonObject readJson()
```

Legge il flusso dati come stringa, e lo converte in JSON.

**Returns:**

Un oggetto con le informazioni ricevute.

---

Utils.StreamsJSON

## Class JSONWriter

```
java.lang.Object
|
+-- java.io.Writer
|   |
|   +-- java.io.OutputStreamWriter
|       |
|       +-- Utils.StreamsJSON.JSONWriter
```

**All Implemented Interfaces:**

java.io.Closeable, java.io.Flushable, java.lang.Appendable

---

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

---

```
public class JSONWriter
extends java.io.OutputStreamWriter
```

Estende da OutputStreamWriter, permette di scrivere un oggetto JSON sul flusso dati della socket.

## Constructors

### JSONWriter

```
public JSONWriter(java.io.OutputStream out)
```

Costruttore JSONWriter.

**Parameters:**

out - Flusso dati in uscita.

## Methods

# write

```
public void write(JsonObject jsonObj)
```

Converte un oggetto JsonObject in stringa, e la scrive sul flusso dati.

**Parameters:**

jsonObj - Contiene le informazioni da scrivere sul flusso.

**Overrides:**

write in class java.io.OutputStreamWriter

# INDEX

## A

[addSection](#) ... 44  
[addToCart](#) ... 41  
[AddElement](#) ... 59  
[AddReview](#) ... 36  
[AddReviewController](#) ... 29  
[AddReviewController](#) ... 29

## C

[catalogue](#) ... 3  
[categories](#) ... 49  
[checkCredentials](#) ... 38  
[clientSocket](#) ... 3  
[close](#) ... 4  
[connection](#) ... 52  
[convertDate](#) ... 63  
[Cart](#) ... 40  
[Cart](#) ... 40  
[CartController](#) ... 6  
[CartController](#) ... 6  
[CartInfoController](#) ... 8  
[CartInfoController](#) ... 8  
[Catalogue](#) ... 43  
[Catalogue](#) ... 43  
[CatalogueController](#) ... 15  
[CatalogueController](#) ... 15  
[Category](#) ... 44  
[Category](#) ... 45  
[CheckAddressString](#) ... 59  
[CheckBornDate](#) ... 59  
[CheckCity](#) ... 60  
[CheckIdentityCode](#) ... 60  
[CheckIfAvailable](#) ... 42  
[CheckIfExistQuery](#) ... 53  
[CheckInteger](#) ... 60  
[CheckNumberOfOrders](#) ... 36  
[CheckoutController](#) ... 9  
[CheckoutController](#) ... 9  
[CheckoutController.TableInfo](#) ... 10  
[CheckString](#) ... 60  
[CheckTelephoneNumber](#) ... 61  
[Close](#) ... 53  
[CloseResultSet](#) ... 53  
[CloseStatement](#) ... 53  
[Commit](#) ... 53  
[ConfigFile](#) ... 3  
[ConfirmCheckout](#) ... 36  
[Core](#) ... 35  
[Core](#) ... 35  
[CreateFolder](#) ... 61  
[CreateXML](#) ... 61

## D

[Database](#) ... 52  
[Database](#) ... 52  
[DefaultCartController](#) ... 11  
[DefaultCartController](#) ... 11  
[DefaultCartController.TableInfo](#) ... 12  
[DefaultReviewsController](#) ... 30  
[DefaultReviewsController](#) ... 30

## F

[fromMySQLDate](#) ... 63

## G

[getCatalogue](#) ... 44  
[getCategoryDesc](#) ... 45  
[getCategoryID](#) ... 45  
[getCategoryName](#) ... 45  
[getCategoryName](#) ... 47  
[getCost](#) ... 10  
[getCourierName](#) ... 10  
[getDiscount](#) ... 13  
[getExpressTime](#) ... 10  
[getGoalDate](#) ... 25  
[getProductDesc](#) ... 47  
[getProductDiscount](#) ... 47  
[getProductDiscountPercentage](#) ... 48  
[getProductID](#) ... 48  
[getProductImage](#) ... 48  
[getProductName](#) ... 13  
[getProductName](#) ... 25  
[getProductName](#) ... 48  
[getProductPrice](#) ... 13  
[getProductPrice](#) ... 48  
[getProductQuantity](#) ... 48  
[getQuantity](#) ... 13  
[getQuantity](#) ... 25  
[getSectionDesc](#) ... 50  
[getSectionID](#) ... 50  
[getSectionName](#) ... 50  
[getStartDate](#) ... 25  
[getStatus](#) ... 26  
[getTotal](#) ... 13  
[getTotal](#) ... 26  
[GetLastInsertId](#) ... 53

## H

[hasCategories](#) ... 50  
[hasDiscount](#) ... 48  
[hasProducts](#) ... 45  
[hasSections](#) ... 44

## I

[initialize](#) ... 7  
[initialize](#) ... 8  
[initialize](#) ... 9  
[initialize](#) ... 12  
[initialize](#) ... 15  
[initialize](#) ... 16  
[initialize](#) ... 18  
[initialize](#) ... 19  
[initialize](#) ... 21  
[initialize](#) ... 24  
[initialize](#) ... 27  
[initialize](#) ... 30  
[initialize](#) ... 31  
[initialize](#) ... 32  
[initialize](#) ... 33  
[initialize](#) ... 58  
[initializeCartPage](#) ... 22  
[initializeCataloguePage](#) ... 22  
[initializeDefaultCartPage](#) ... 7  
[initializeOrdersPage](#) ... 22  
[initializePage](#) ... 8  
[initializePage](#) ... 9  
[initializePage](#) ... 12  
[initializePage](#) ... 16  
[initializePage](#) ... 24  
[initializePage](#) ... 27  
[initializePage](#) ... 30  
[initializePanel](#) ... 31  
[Initialize](#) ... 54  
[InitializeResultSet](#) ... 54  
[InitializeStatement](#) ... 54  
[InitializeXML](#) ... 61  
[InsertQuery](#) ... 54

## J

[JSONReader](#) ... 64  
[JSONReader](#) ... 64  
[JSONWriter](#) ... 65  
[JSONWriter](#) ... 65

## L

[loadDefaultCart](#) ... 7  
[loadDefaultForm](#) ... 33  
[loadDiscount](#) ... 38  
[loadOrderInfo](#) ... 28  
[loadProductInCart](#) ... 7  
[loadReview](#) ... 34  
[loadReviews](#) ... 34  
[LoadCatalogue](#) ... 37  
[LoadCouriers](#) ... 37  
[LoadOrders](#) ... 37  
[LoadReviews](#) ... 37  
[LoginFormController](#) ... 18  
[LoginFormController](#) ... 18

## M

[main](#) ... 4  
[main](#) ... 56  
[MainClient](#) ... 3  
[MainClient](#) ... 4  
[MainFormController](#) ... 21  
[MainFormController](#) ... 21  
[MainServer](#) ... 55  
[MainServer](#) ... 55  
[MessageBox](#) ... 62  
[MessageBox](#) ... 62  
[MessageBoxOKController](#) ... 57  
[MessageBoxOKController](#) ... 57

## O

[OrderInfoController](#) ... 23  
[OrderInfoController](#) ... 23  
[OrderInfoController.TableInfo](#) ... 24  
[OrdersController](#) ... 27  
[OrdersController](#) ... 27

## P

[product](#) ... 42  
[productInCarts](#) ... 40  
[products](#) ... 45  
[Product](#) ... 46  
[Product](#) ... 47  
[ProductInCart](#) ... 41  
[ProductInCart](#) ... 42  
[ProductInfoController](#) ... 16  
[ProductInfoController](#) ... 16

## Q

[quantity](#) ... 42

## R

[reader](#) ... 3  
[readJson](#) ... 65  
[registerUser](#) ... 39  
[removeFromCart](#) ... 41  
[ReadXML](#) ... 62  
[RegistrationFormController](#) ... 19  
[RegistrationFormController](#) ... 19  
[RemoveQuery](#) ... 54  
[RemoveReview](#) ... 38  
[ReviewInfoController](#) ... 32  
[ReviewInfoController](#) ... 32  
[ReviewsController](#) ... 33  
[ReviewsController](#) ... 33  
[Rollback](#) ... 55

## S

[setCatalogue](#) ... 44  
[setCategory](#) ... 50  
[setCost](#) ... 11  
[setCourierName](#) ... 11  
[setDiscount](#) ... 14  
[setExpressTime](#) ... 11  
[setGoalDate](#) ... 26  
[setMessage](#) ... 58  
[setOldStage](#) ... 58  
[setProduct](#) ... 17  
[setProduct](#) ... 46  
[setProductInfo](#) ... 31  
[setProductInfo](#) ... 34  
[setProductName](#) ... 14  
[setProductName](#) ... 26  
[setProductPrice](#) ... 14  
[setQuantity](#) ... 14  
[setQuantity](#) ... 26  
[setQuantity](#) ... 48  
[setReview](#) ... 32  
[setReviews](#) ... 31  
[setStartDate](#) ... 26  
[setStatus](#) ... 26  
[setTotal](#) ... 14  
[setTotal](#) ... 26  
[setXML](#) ... 19  
[setXML](#) ... 20  
[siteAddress](#) ... 4  
[start](#) ... 5  
[Section](#) ... 49  
[Section](#) ... 49  
[SelectionQuery](#) ... 55

## T

[toMySQLDate](#) ... 63  
[toString](#) ... 42  
[toString](#) ... 46  
[toString](#) ... 49  
[toString](#) ... 51  
[TableInfo](#) ... 10  
[TableInfo](#) ... 13  
[TableInfo](#) ... 25

## U

[updateQuantity](#) ... 41  
[user](#) ... 4  
[UpdateOrders](#) ... 4  
[Utils](#) ... 58  
[Utils](#) ... 59

## V

[ValidateEmail](#) ... 62

## W

[write](#) ... 66  
[writer](#) ... 4