

# **COBrA DAPPs**

**by Aldo D'Aquino**

Copyright 2018

---

**Package**

**com.aldodaquino.cobra.authorserver**

## com.aldodaquino.cobra.authorserver Class Main

```
java.lang.Object
|
+--com.aldodaquino.cobra.authorserver.Main
```

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

The Main class of the author server. The author server remains always online to listen access request. The author can use its server to deploy his content on the blockchain and publish it on the catalog. This server serves content also when the author has closed his client. Includes two functions that handle the two urls /deploy (to deploy a content) and /access to access a content.

**Author:**

Aldo D'Aquino.

**Version:**

1.0.

### Constructors

#### Main

```
public Main()
```

### Methods

#### main

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

Main method.

**Parameters:**

args - a String[] passed through command line.

**Throws:**

IOException - if it is not possible to create the server.

---

**Package**

**com.aldodaquino.cobra.connections**

## com.aldodaquino.cobra.connections Interface API

---

public interface **API**  
extends

Interface for the author server API. Specify the urls of the handlers. Check Main class in [com.aldodaquino.cobra.authorserver](#)

**Author:**

Aldo D'Aquino.

**Version:**

1.0.

---

### Fields

#### **DEPLOY\_API\_PATH**

public static final java.lang.String **DEPLOY\_API\_PATH**

Constant value: **/deploy**

---

#### **ACCESS\_API\_PATH**

public static final java.lang.String **ACCESS\_API\_PATH**

Constant value: **/access**

## com.aldodaquino.cobra.connections Class CobraHttpHelper

```
java.lang.Object
├── com.aldodaquino.javautils.HttpHelper
│   └── com.aldodaquino.cobra.connections.CobraHttpHelper
```

---

```
public class CobraHttpHelper
    extends HttpHelper
```

Contains method that help to make http request. Works with JSON body for POST request and query-style GET parameters.

**Author:**

Aldo D'Aquino.

**Version:**

1.0.

---

### Constructors

#### CobraHttpHelper

```
public CobraHttpHelper()
```

### Methods

#### newHandler

```
public static com.sun.net.httpserver.HttpHandler
newHandler(java.util.function.BiConsumer consumer,
            Status status)
```

Return new HttpHandler

**Parameters:**

consumer - a function to be called when a new request arrive. The consumer has to accept the HttpExchange request and the Status.

status - to be passed to the consumer.

**Returns:**

HttpHandler, the handler.

## com.aldodaquino.cobra.connections Class Status

```
java.lang.Object
  |
  +--com.aldodaquino.cobra.connections.Status
```

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

Defines the status for the author-server.

**Author:**

Aldo D'Aquino.

**Version:**

1.0.

### Fields

#### privateKey

```
public java.lang.String privateKey
```

#### credentials

```
public org.web3j.crypto.Credentials credentials
```

#### catalogManager

```
public com.aldodaquino.cobra.main.CatalogManager catalogManager
```

#### hostname

```
public java.lang.String hostname
```

#### port

```
public int port
```

### Constructors

#### Status

```
public Status()
```

(continued from last page)



---

**Package**

**com.aldodaquino.cobra.gui**

## com.aldodaquino.cobra.gui

### Class Main

```
java.lang.Object
|
+-com.aldodaquino.cobra.gui.Main
```

---

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

The GUI Main. Starts the GUI with the [StarterPanel](#).

**Author:**

Aldo D'Aquino.

**Version:**

1.0.

---

## Constructors

### Main

```
public Main()
```

## Methods

### main

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

Main method.

**Parameters:**

args - an empty array, no parameters are required.

## com.aldodaquino.cobra.gui

### Class Status

```
java.lang.Object
|
+--com.aldodaquino.cobra.gui.Status
```

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

The Status class. A Status object is generated in the [StarterPanel](#). Contains all the information about the user and the catalog and is passed through the classes.

**Author:**  
Aldo D'Aquino.

**Version:**  
1.0.

## Fields

### ROLE\_CUSTOMER

```
public static final int ROLE_CUSTOMER
```

Constant value: 0

### ROLE\_AUTHOR

```
public static final int ROLE_AUTHOR
```

Constant value: 1

### credentials

```
public org.web3j.crypto.Credentials credentials
```

## Constructors

### Status

```
public Status()
```

## Methods

### login

```
public void login(java.lang.String privateKey)
    throws javax.naming.OperationNotSupportedException
```

Given a private key generates and stores the credentials for this user.

---

(continued from last page)

**Parameters:**

`privateKey` - the user's private key.

**Throws:**

`OperationNotSupportedException` - if the user is already logged in.

---

## **connectCatalog**

```
public void connectCatalog(java.lang.String catalogAddress)  
    throws javax.naming.OperationNotSupportedException
```

Connects to an existent catalog and stores it.

**Parameters:**

`catalogAddress` - the catalog address.

**Throws:**

`OperationNotSupportedException` - if the user is not logged in.

---

## **disconnectCatalog**

```
public void disconnectCatalog()
```

Disconnects from the catalog.

---

## **deployCatalog**

```
public void deployCatalog()  
    throws javax.naming.OperationNotSupportedException
```

Deploys a new catalog with the user credentials.

**Throws:**

`OperationNotSupportedException` - if the user is not logged in.

---

## **getUserAddress**

```
public java.lang.String getUserAddress()  
    throws javax.naming.OperationNotSupportedException
```

Returns the user address.

**Returns:**

a String with the user address.

**Throws:**

`OperationNotSupportedException` - if the user is not logged in.

---

## **isCatalogOwner**

```
public boolean isCatalogOwner()  
    throws javax.naming.OperationNotSupportedException
```

Returns true if the user that deployed the catalog is the current user, false otherwise.

**Returns:**

true if the current user is the catalog owner, false otherwise.

---

(continued from last page)

**Throws:**

`OperationNotSupportedException` - if the user is not connected to a catalog.

---

**getCatalogManager**

```
public CatalogManager getCatalogManager()
```

Returns the catalog manager.

**Returns:**

the `CatalogManager`.

---

**getPrivateKey**

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

Returns the user's private key.

**Returns:**

a `String` with the user's private key.

---

**setRole**

```
public void setRole(int role)
```

Set the role for the current user.

**Parameters:**

`role` - an `int` specifying the role.

---

## com.aldodaquino.cobra.gui Class Utils

```
java.lang.Object
|
+--com.aldodaquino.cobra.gui.Utils
```

---

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

Utilities for the GUI.

**Author:**  
Aldo D'Aquino.

**Version:**  
1.0.

---

### Constructors

#### Utils

```
public Utils()
```

### Methods

#### newWindow

```
public static javax.swing.JFrame newWindow(java.lang.String title,
      javax.swing.JComponent panel,
      boolean exitOnClose)
```

Show a new centered Window with fixed dimensions and not resizable.

**Parameters:**

title - of the Window.  
panel - to show in the Window body.  
exitOnClose - if true exit the Client when the Window is close.

**Returns:**

the generated JFrame.

---

#### newMessageDialog

```
public static void newMessageDialog(java.lang.String msg)
```

Shows a dialog, running on another thread.

**Parameters:**

msg - the dialog message.

---

#### newErrorDialog

```
public static void newErrorDialog(java.lang.String msg)
```

---

(continued from last page)

Shows an error dialog, running on another thread.

**Parameters:**

msg - the error message.

---

## newExitDialog

```
public static void newExitDialog(java.lang.String errorMessage)
```

Shows an error dialog that advice that the program will close, then exit the program.

**Parameters:**

errorMessage - the error message.

---

## newConfirmDialog

```
public static boolean newConfirmDialog(java.lang.String msg)
```

Shows a confirmation dialog (yes/no).

**Parameters:**

msg - the question.

**Returns:**

true for yes, false for no.

---

## setFontSize

```
public static void setFontSize(javax.swing.JLabel label,  
                                int fontSize)
```

Set the font size of a label.

**Parameters:**

label - of which set the font size.  
fontSize - the size to set.

---

## openFileDialog

```
public static java.io.File openFileDialog()
```

Show the file selection dialog for file choosing and saving.

**Returns:**

a File.

---

## saveFileDialog

```
public static java.io.File saveFileDialog(java.lang.String defaultName)
```

Show the file selection dialog for file choosing and saving.

**Parameters:**

defaultName - specify the original filename of the incoming file.

**Returns:**

a File.

---

---

**Package**

**com.aldodaquino.cobra.gui.components**



## com.aldodaquino.cobra.gui.components

### Class AsyncPanel

```

java.lang.Object
  |
  +- java.awt.Component
        |
        +- java.awt.Container
              |
              +- javax.swing.JComponent
                    |
                    +- javax.swing.JPanel
                          |
                          +- com.aldodaquino.cobra.gui.components.AsyncPanel
  
```

#### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

#### Direct Known Subclasses:

[AuthorPanel](#), [ContentInfoPanel](#), [UpgradablePanel](#)

```

public class AsyncPanel
extends javax.swing.JPanel
  
```

A JPanel that listen for ancestor changes. If it is added to an ancestor it saves it a protected variable window. Include a doAsync method that runs the Runnable in another Thread and set the glass panel of the window in a loading state to prevent actions from the user.

#### Author:

Aldo D'Aquino.

#### Version:

1.0.

## Fields

### window

protected javax.swing.JFrame **window**

## Constructors

### AsyncPanel

protected **AsyncPanel**()

Constructor.

## Methods

### doAsync

protected void **doAsync**(java.lang.Runnable runnable)

Run a runnable asynchronously. Shows a loading panel during the loading.

(continued from last page)

**Parameters:**

runnable - to be run.

## com.aldodaquino.cobra.gui.components

### Class AuthorContentTable

```

java.lang.Object
  |-- java.awt.Component
        |-- java.awt.Container
              |-- javax.swing.JComponent
                    |-- javax.swing.JTable
                          |-- com.aldodaquino.cobra.gui.components.AuthorContentTable
  
```

#### All Implemented Interfaces:

```

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.swing.event.RowSorterListener,
javax.accessibility.Accessible, javax.swing.event.CellEditorListener, javax.swing.event.ListSelectionListener,
javax.swing.event.TableColumnModelListener, javax.swing.Scrollable, javax.swing.event.TableModelListener
  
```

```

public class AuthorContentTable
extends javax.swing.JTable
  
```

A JTable for the [AuthorPanel](#).

#### Author:

Aldo D'Aquino.

#### Version:

1.0.

## Constructors

### AuthorContentTable

```

public AuthorContentTable(Status status,
                           java.util.List contents)
  
```

Constructor.

#### Parameters:

status - the Status object.

contents - a List of Content objects.

## Methods

### isCellEditable

```

public boolean isCellEditable(int row,
                               int column)
  
```

## com.aldodaquino.cobra.gui.components

### Class CatalogForm

```
java.lang.Object
├-- java.awt.Component
│   ├── java.awt.Container
│   │   ├── javax.swing.JComponent
│   │   │   ├── javax.swing.JPanel
│   │   │   └-- com.aldodaquino.cobra.gui.components.CatalogForm
```

#### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

---

```
public class CatalogForm
extends javax.swing.JPanel
```

A JPanel used in the [StarPanel](#). Ask the user which catalog to connect to.

#### Author:

Aldo D'Aquino.

#### Version:

1.0.

---

## Constructors

### CatalogForm

```
public CatalogForm(java.util.function.Consumer connectCallback,
                   java.lang.Runnable deployCallback)
```

Constructor.

#### Parameters:

connectCallback - a callback to be invoked when the deploy button is clicked or the form is submitted and the catalog address is correct.

deployCallback - a callback invoked when the deploy button is clicked.

## com.aldodaquino.cobra.gui.components

### Class ChartWidget

```
java.lang.Object
├-- java.awt.Component
│   ├── java.awt.Container
│   │   ├── javax.swing.JComponent
│   │   │   ├── javax.swing.JPanel
│   │   │   │   ├── com.aldodaquino.cobra.gui.components.InfoPanel
│   │   │   │   │   └-- com.aldodaquino.cobra.gui.components.ChartWidget
```

#### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

public class **ChartWidget**  
extends [InfoPanel](#)

An [InfoPanel](#) used in the [CustomerPanel](#). Shows the chart of the catalog, that includes: the latest content, the most popular content, and the highest rated content in absolute and for each category.

#### Author:

Aldo D'Aquino.

#### Version:

1.0.

## Constructors

### ChartWidget

```
public ChartWidget(Status status)
```

Constructor.

#### Parameters:

status - the Status object.

## com.aldodaquino.cobra.gui.components

# Class ComponentFactory

```
java.lang.Object
|
+--com.aldodaquino.cobra.gui.components.ComponentFactory
```

---

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

Components factory: makes components creation faster and easier.

**Author:**

Aldo D'Aquino.

**Version:**

1.0.

---

## Constructors

### ComponentFactory

```
public ComponentFactory()
```

## Methods

### newButton

```
public static javax.swing.JButton newButton(java.lang.String text,
                                             java.awt.event.ActionListener e)
```

Returns a new JButton.

**Parameters:**

text - the text of the button.

e - the ActionListener of the button.

**Returns:**

a JButton.

---

### newTextField

```
public static javax.swing.JTextField newTextField(java.awt.event.ActionListener e)
```

Returns a new JTextField.

**Parameters:**

e - the ActionListener called when enter is pressed.

**Returns:**

a JTextField.

---

(continued from last page)

## newBorder

```
public static javax.swing.border.Border newBorder(int width,  
int height)
```

Returns a new border with the specified dimensions.

**Parameters:**

width - the border width.  
height - the border height.

**Returns:**

a Border.

---

## newVSpacer

```
public static java.awt.Component newVSpacer(java.awt.Dimension dimension)
```

Returns a vertical spacer of the specified dimensions.

**Parameters:**

dimension - the dimensions.

**Returns:**

a Component.

## com.aldodaquino.cobra.gui.components

### Class ContentList

```
java.lang.Object
├-- java.awt.Component
│   ├── java.awt.Container
│   │   ├── javax.swing.JComponent
│   │   │   ├── javax.swing.JList
│   │   │   └-- com.aldodaquino.cobra.gui.components.ContentList
```

#### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible,  
javax.swing.Scrollable

---

public class **ContentList**  
extends javax.swing.JList

a JList of Content objects.

**Author:**

Aldo D'Aquino.

**Version:**

1.0.

---

## Constructors

### ContentList

```
public ContentList(Status status,  
                  java.util.List contents)
```

Constructor.

**Parameters:**

status - the Status object.

contents - a List of Content objects.



## com.aldodaquino.cobra.gui.components

### Class InfoPanel

```
java.lang.Object
├-- java.awt.Component
│   ├── java.awt.Container
│   │   ├── javax.swing.JComponent
│   │   │   ├── javax.swing.JPanel
│   │   │   └-- com.aldodaquino.cobra.gui.components.InfoPanel
```

#### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

#### Direct Known Subclasses:

[AuthorInfoPanel](#), [GenreInfoPanel](#), [ChartWidget](#)

---

```
public class InfoPanel
extends javax.swing.JPanel
```

Info panel, superclass of [AuthorInfoPanel](#), [GenreInfoPanel](#) and [ChartWidget](#).

#### Author:

Aldo D'Aquino.

#### Version:

1.0.

---

## Fields

### latestLabel

```
protected final com.aldodaquino.cobra.gui.components.LabelPanel latestLabel
```

---

### mostPopularLabel

```
protected final com.aldodaquino.cobra.gui.components.LabelPanel mostPopularLabel
```

---

### highestRatedLabel

```
protected final com.aldodaquino.cobra.gui.components.LabelPanel highestRatedLabel
```

---

### mostEnjoyedLabel

```
protected final com.aldodaquino.cobra.gui.components.LabelPanel mostEnjoyedLabel
```

---

## biggestPriceFairnessLabel

```
protected final com.aldodaquino.cobra.gui.components.LabelPanel  
biggestPriceFairnessLabel
```

---

## highestContentMeaningLabel

```
protected final com.aldodaquino.cobra.gui.components.LabelPanel  
highestContentMeaningLabel
```

---

## Constructors

### InfoPanel

```
protected InfoPanel(Status status,  
                    java.lang.String mainLabelString)
```

Constructor. Can be invoked only by its children.

#### Parameters:

status - the Status object.

mainLabelString - the String to be putted in the top of the panel, with a bigger font.

## com.aldodaquino.cobra.gui.components

### Class LabelPanel

```
java.lang.Object
├-- java.awt.Component
│   ├── java.awt.Container
│   │   ├── javax.swing.JComponent
│   │   │   ├── javax.swing.JPanel
│   │   │   │   ├── com.aldodaquino.cobra.gui.components.AsyncPanel
│   │   │   │   │   ├── com.aldodaquino.cobra.gui.components.UpgradablePanel
│   │   │   │   │   │   └-- com.aldodaquino.cobra.gui.components.LabelPanel
```

#### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

```
public class LabelPanel
extends UpgradablePanel
```

Label JPanel, contains a fixed label and an updatable value. The initial value is a loader spinner.

**Author:**

Aldo D'Aquino.

**Version:**

1.0.

## Methods

### update

```
public void update(Content content)
```

Set the content name as value of the panel in a link style.

**Parameters:**

content - the Content object.

## com.aldodaquino.cobra.gui.components

### Class LoginForm

```
java.lang.Object
├-- java.awt.Component
│   ├── java.awt.Container
│   │   ├── javax.swing.JComponent
│   │   │   ├── javax.swing.JPanel
│   │   │   └-- com.aldodaquino.cobra.gui.components.LoginForm
```

#### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

---

public class **LoginForm**  
extends javax.swing.JPanel

Login Form under the Logo in the Starter Panel.

#### Author:

Aldo D'Aquino.

#### Version:

1.0.

---

## Constructors

### LoginForm

```
public LoginForm(java.util.function.Consumer loginCallback)
```

Constructor.

#### Parameters:

loginCallback - a String Consumer called if the login button is clicked or enter is pressed and the private key is valid.

## com.aldodaquino.cobra.gui.components

### Class Logo

```
java.lang.Object
  |-- java.awt.Component
        |-- java.awt.Container
              |-- javax.swing.JComponent
                    |-- javax.swing.JPanel
                          |-- com.aldodaquino.cobra.gui.components.Logo
```

#### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

---

public class **Logo**  
extends javax.swing.JPanel

Application Logo for the [StarterPanel](#).

#### Author:

Aldo D'Aquino.

#### Version:

1.0.

---

## Constructors

### Logo

```
public Logo()
```

Constructor.

## com.aldodaquino.cobra.gui.components

### Class NewContentsWidget

```
java.lang.Object
├-- java.awt.Component
│   ├── java.awt.Container
│   │   ├── javax.swing.JComponent
│   │   │   ├── javax.swing.JPanel
│   │   │   └-- com.aldodaquino.cobra.gui.components.NewContentsWidget
```

#### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

---

```
public class NewContentsWidget
extends javax.swing.JPanel
```

A widget inserted in the [CustomerPanel](#) under the [ChartWidget](#). Allows to select how many content the user want in the new content list and shows a window with this content.

**Author:**

Aldo D'Aquino.

**Version:**

1.0.

---

## Constructors

### NewContentsWidget

```
public NewContentsWidget(Status status)
```

Constructor.

**Parameters:**

status - the Status object.

## com.aldodaquino.cobra.gui.components

### Class RoleForm

```
java.lang.Object
├-- java.awt.Component
│   ├── java.awt.Container
│   │   ├── javax.swing.JComponent
│   │   │   ├── javax.swing.JPanel
│   │   │   └-- com.aldodaquino.cobra.gui.components.RoleForm
```

#### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

```
public class RoleForm
extends javax.swing.JPanel
```

The last of the Panels shows in the [StarterPanel](#). Asks the user if want to see the [CustomerPanel](#) or the [AuthorPanel](#).

#### Author:

Aldo D'Aquino.

#### Version:

1.0.

## Constructors

### RoleForm

```
public RoleForm( java.lang.Runnable browseCallback,
                 java.lang.Runnable manageCallback,
                 java.lang.Runnable disconnectCallback,
                 java.lang.Runnable deleteCallback)
```

Constructor.

#### Parameters:

browseCallback - callback for the "Browser contents" button.

manageCallback - callback for the "Manage my contents" button.

disconnectCallback - callback for the "Disconnect" button.

deleteCallback - callback for the "Delete catalogManager" contents button. If null the button is not shown.

## com.aldodaquino.cobra.gui.components

### Class StarPanel

```

java.lang.Object
  |-- java.awt.Component
    |-- java.awt.Container
      |-- javax.swing.JComponent
        |-- javax.swing.JPanel
          |-- com.aldodaquino.cobra.gui.components.AsyncPanel
            |-- com.aldodaquino.cobra.gui.components.UpgradablePanel
              |-- com.aldodaquino.cobra.gui.components.StarPanel

```

#### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

public class **StarPanel**  
 extends [UpgradablePanel](#)

Shows rating or ask the user to vote.

#### Author:

Aldo D'Aquino.

#### Version:

1.0.

## Constructors

### StarPanel

```
public StarPanel()
```

Constructor. Initialize an empty StarPanel, enabled.

### StarPanel

```
public StarPanel(int rating)
```

Constructor. Initialize a StarPanel with the specified rating, disabled.

#### Parameters:

rating - in the interval [0, 5]. 0 means no rating.

## Methods

### getRating

```
public int getRating()
```

Returns the rating.

#### Returns:



(continued from last page)

int in the interval  $[0, 5]$ . 0 means no rating.

## com.aldodaquino.cobra.gui.components

### Class UpgradablePanel

```

java.lang.Object
  |
  +- java.awt.Component
        |
        +- java.awt.Container
              |
              +- javax.swing.JComponent
                    |
                    +- javax.swing.JPanel
                          |
                          +- com.aldodaquino.cobra.gui.components.AsyncPanel
                                |
                                +- com.aldodaquino.cobra.gui.components.UpgradablePanel
  
```

#### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
 javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

#### Direct Known Subclasses:

[CustomerPanel](#), [StarterPanel](#), [LabelPanel](#), [StarPanel](#), [UserInfo](#)

```

public class UpgradablePanel
extends AsyncPanel
  
```

An [AsyncPanel](#) with java.awt.GridBagLayout. Implements method to easy replace panels.

#### Author:

Aldo D'Aquino.

#### Version:

1.0.

## Constructors

### UpgradablePanel

```
protected UpgradablePanel()
```

Constructor. Set the GridBagLayout. Can be called only by its children.

## Methods

### replaceComponent

```
protected void replaceComponent( java.awt.Component toBeReplaced,
                                java.awt.Component replacement,
                                java.awt.GridBagConstraints position)
```

Replace a component with another one. Can be called only by its children.

#### Parameters:

toBeReplaced - the old component.  
 replacement - the new component.  
 position - the position of the component to be replaced.

(continued from last page)

## **newGBC**

```
public static java.awt.GridBagConstraints newGBC(int x,  
int y)
```

A public method that help to create GridBagConstraints in less time.

### **Parameters:**

- x - the gridx property of the GridBagConstraints object.
- y - the gridy property of the GridBagConstraints object.

### **Returns:**

a GridBagConstraints object.

## com.aldodaquino.cobra.gui.components

### Class UserInfo

```
java.lang.Object
├-- java.awt.Component
│   ├── java.awt.Container
│   │   ├── javax.swing.JComponent
│   │   │   ├── javax.swing.JPanel
│   │   │   │   ├── com.aldodaquino.cobra.gui.components.AsyncPanel
│   │   │   │   │   ├── com.aldodaquino.cobra.gui.components.UpgradablePanel
│   │   │   │   │   │   └-- com.aldodaquino.cobra.gui.components.UserInfo
```

#### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

---

```
public class UserInfo
extends UpgradablePanel
```

Shows the catalog to which the user is connected and the user address and premium status.

#### Author:

Aldo D'Aquino.

#### Version:

1.0.

---

## Constructors

### UserInfo

```
public UserInfo(Status status)
```

Constructor.

#### Parameters:

status - the Status object.

---

## Methods

### updateStatus

```
public void updateStatus()
```

Update its fields according with the latest information available in status.

## com.aldodaquino.cobra.gui.components Class ViewsContentTable

```

java.lang.Object
  |-- java.awt.Component
    |-- java.awt.Container
      |-- javax.swing.JComponent
        |-- javax.swing.JTable
          |-- com.aldodaquino.cobra.gui.components.ViewsContentTable

```

### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
 javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.swing.event.RowSorterListener,  
 javax.accessibility.Accessible, javax.swing.event.CellEditorListener, javax.swing.event.ListSelectionListener,  
 javax.swing.event.TableColumnModelListener, javax.swing.Scrollable, javax.swing.event.TableModelListener

```

public class ViewsContentTable
extends javax.swing.JTable

```

A content list that includes also the number of views. Accessible in the [CustomerPanel](#).

#### Author:

Aldo D'Aquino.

#### Version:

1.0.

## Constructors

### ViewsContentTable

```

public ViewsContentTable(Status status,
                        java.util.List contents)

```

Constructor.

#### Parameters:

status - the Status object.

contents - a List of Content objects to be shown in this table.

## Methods

### isCellEditable

```

public boolean isCellEditable(int row,
                             int column)

```

---

**Package**

**com.aldodaquino.cobra.gui.constants**

## com.aldodaquino.cobra.gui.constants

### Class Colors

```
java.lang.Object
  |
  +--com.aldodaquino.cobra.gui.constants.Colors
```

---

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

Colors constants for the Graphic Interface.

**Author:**  
Aldo D'Aquino.

**Version:**  
1.0.

---

### Fields

#### **GREEN**

```
public static final java.awt.Color GREEN
```

---

#### **RED**

```
public static final java.awt.Color RED
```

---

### Constructors

#### **Colors**

```
public Colors()
```

## com.aldodaquino.cobra.gui.constants

### Class Dimensions

```
java.lang.Object
  |
  +--com.aldodaquino.cobra.gui.constants.Dimensions
```

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

Dimensions constant for the Graphic Interface.

**Author:**

Aldo D'Aquino.

**Version:**

1.0.

## Fields

### LOGO\_SIZE

```
public static final int LOGO_SIZE
```

Constant value: **128**

### STAR\_SIZE

```
public static final int STAR_SIZE
```

Constant value: **10**

### STARTER\_PANEL\_PADDING

```
public static final java.awt.Dimension STARTER_PANEL_PADDING
```

### INFO\_PANEL\_PADDING

```
public static final java.awt.Dimension INFO_PANEL_PADDING
```

### LATERAL\_BAR\_PADDING

```
public static final java.awt.Dimension LATERAL_BAR_PADDING
```

### V\_SPACER\_S

```
public static final java.awt.Dimension V_SPACER_S
```



---

## V\_SPACER\_M

```
public static final java.awt.Dimension V_SPACER_M
```

---

## V\_SPACER\_L

```
public static final java.awt.Dimension V_SPACER_L
```

## Constructors

### Dimensions

```
public Dimensions()
```

## com.aldodaquino.cobra.gui.constants

### Class Images

```
java.lang.Object
  |
  +--com.aldodaquino.cobra.gui.constants.Images
```

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

Images used in the Graphic Interface.

**Author:**

Aldo D'Aquino.

**Version:**

1.0.

## Fields

### logo

```
public static final javax.swing.ImageIcon logo
```

### loading

```
public static final javax.swing.ImageIcon loading
```

### emptyStar

```
public static final javax.swing.ImageIcon emptyStar
```

### filledStar

```
public static final javax.swing.ImageIcon filledStar
```

## Constructors

### Images

```
public Images()
```

## Methods

(continued from last page)

## getScaled

```
public static javax.swing.ImageIcon getScaled( javax.swing.ImageIcon icon,  
        int size)
```

Returns the scaled version of an image.

### Parameters:

`icon` - the original image.

`size` - the size that you want the final image to have.

### Returns:

another ImageIcon, scaled.

## com.aldodaquino.cobra.gui.constants

### Class Strings

java.lang.Object

└--com.aldodaquino.cobra.gui.constants.Strings

---

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

Strings constants for the app.

**Author:**

Aldo D'Aquino.

**Version:**

1.0.

---

## Fields

### appName

```
public static final java.lang.String appName
```

Constant value: **COBrA DAPP**

## Constructors

### Strings

```
public Strings()
```

---

**Package**

**com.aldodaquino.cobra.gui.panels**

## com.aldodaquino.cobra.gui.panels

### Class AuthorInfoPanel

```
java.lang.Object
├-- java.awt.Component
│   ├── java.awt.Container
│   │   ├── javax.swing.JComponent
│   │   │   ├── javax.swing.JPanel
│   │   │   │   ├── com.aldodaquino.cobra.gui.components.InfoPanel
│   │   │   │   │   └-- com.aldodaquino.cobra.gui.panels.AuthorInfoPanel
```

#### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

```
public class AuthorInfoPanel
extends InfoPanel
```

Shows the author charts in the catalog.

#### Author:

Aldo D'Aquino.

#### See Also:

[the parent class.](#)

#### Version:

1.0.

## Methods

### newWindow

```
public static void newWindow(Status status,
                             java.lang.String author)
```

Open a new window with this panel.

#### Parameters:

status - the Status.

author - of the content.

## com.aldodaquino.cobra.gui.panels

### Class AuthorPanel

```
java.lang.Object
├-- java.awt.Component
│   ├── java.awt.Container
│   │   ├── javax.swing.JComponent
│   │   │   ├── javax.swing.JPanel
│   │   │   │   ├── com.aldodaquino.cobra.gui.components.AsyncPanel
│   │   │   │   └-- com.aldodaquino.cobra.gui.panels.AuthorPanel
```

#### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

---

```
public class AuthorPanel
extends AsyncPanel
```

The author panel, a main panel showed after the starter panel if the user have chosen the author role.

#### Author:

Aldo D'Aquino.

#### Version:

1.0.

---

## Constructors

### AuthorPanel

```
public AuthorPanel(Status status)
```

Constructor.

#### Parameters:

status - the Status object.

## com.aldodaquino.cobra.gui.panels

### Class ContentInfoPanel

```

java.lang.Object
  |
  +- java.awt.Component
        |
        +- java.awt.Container
              |
              +- javax.swing.JComponent
                    |
                    +- javax.swing.JPanel
                          |
                          +- com.aldodaquino.cobra.gui.components.AsyncPanel
                                |
                                +- com.aldodaquino.cobra.gui.panels.ContentInfoPanel
  
```

#### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
 javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

public class **ContentInfoPanel**  
 extends [AsyncPanel](#)

Shows info about a content. Allows also to buy and consume the content or gift it to another user.

#### Author:

Aldo D'Aquino.

#### Version:

1.0.

## Methods

### newWindow

```

public static void newWindow(Status status,
                             java.lang.String address)
  
```

Open a new window with this panel.

#### Parameters:

status - the Status.

address - of the content.



## com.aldodaquino.cobra.gui.panels

### Class CustomerPanel

```
java.lang.Object
├-- java.awt.Component
│   ├── java.awt.Container
│   │   ├── javax.swing.JComponent
│   │   │   ├── javax.swing.JPanel
│   │   │   │   ├── com.aldodaquino.cobra.gui.components.AsyncPanel
│   │   │   │   │   ├── com.aldodaquino.cobra.gui.components.UpgradablePanel
│   │   │   │   │   │   └-- com.aldodaquino.cobra.gui.panels.CustomerPanel
```

#### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

```
public class CustomerPanel
```

```
extends UpgradablePanel
```

The customer panel, a main panel showed after the starter panel if the user have chosen the customer role.

**Author:**

Aldo D'Aquino.

**Version:**

1.0.

## Constructors

### CustomerPanel

```
public CustomerPanel(Status status)
```

Constructor.

**Parameters:**

status - the Status object.

## com.aldodaquino.cobra.gui.panels Class GenreInfoPanel

```
java.lang.Object
├-- java.awt.Component
│   ├── java.awt.Container
│   │   ├── javax.swing.JComponent
│   │   │   ├── javax.swing.JPanel
│   │   │   │   ├── com.aldodaquino.cobra.gui.components.InfoPanel
│   │   │   │   │   └-- com.aldodaquino.cobra.gui.panels.GenreInfoPanel
```

### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

```
public class GenreInfoPanel
extends InfoPanel
```

Shows the genre charts in the catalog.

#### Author:

Aldo D'Aquino.

#### See Also:

[the parent class.](#)

#### Version:

1.0.

## Methods

### newWindow

```
public static void newWindow(Status status,
                             java.lang.String genre)
```

Open a new window with this panel.

#### Parameters:

status - the Status.

genre - of the content.

## com.aldodaquino.cobra.gui.panels

### Class StarterPanel

```

java.lang.Object
  |
  +- java.awt.Component
        |
        +- java.awt.Container
              |
              +- javax.swing.JComponent
                    |
                    +- javax.swing.JPanel
                          |
                          +- com.aldodaquino.cobra.gui.components.AsyncPanel
                                |
                                +- com.aldodaquino.cobra.gui.components.UpgradablePanel
                                      |
                                      +- com.aldodaquino.cobra.gui.panels.StarterPanel

```

#### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

public class **StarterPanel**  
extends [UpgradablePanel](#)

The starter panel. Main panel showed when the app starts. Manage the user login and require all the data to start the app.

#### Author:

Aldo D'Aquino.

#### Version:

1.0.

## Constructors

### StarterPanel

```
public StarterPanel(java.util.function.Consumer whenDone)
```

Constructor.

#### Parameters:

whenDone - a consumer of Status object called when all the requested fields in the status object are completed.

## Methods

### getMinimumSize

```
public java.awt.Dimension getMinimumSize()
```

---

**Package**

**com.aldodaquino.cobra.main**

## com.aldodaquino.cobra.main Class CatalogManager

```
java.lang.Object
|
|--com.aldodaquino.cobra.main.ContractManager
|   |
|   |--com.aldodaquino.cobra.main.CatalogManager
```

public class **CatalogManager**  
extends ContractManager

An higher level Catalog Manager. Contains methods that call the methods in the `com.aldodaquino.cobra.contracts.CatalogContract` generated by Web3j and parse and aggregate the result in a more comfortable representation.

**Author:**

Aldo D'Aquino.

**Version:**

1.0.

## Constructors

### CatalogManager

```
public CatalogManager(org.web3j.crypto.Credentials credentials)
```

Deploy and manage a new catalog contract.

**Parameters:**

`credentials` - your account credentials.

### CatalogManager

```
public CatalogManager(org.web3j.crypto.Credentials credentials,  
                      java.lang.String contractAddress)
```

Load and manage an existent catalog contract.

**Parameters:**

`credentials` - your account credentials.

`contractAddress` - the existent contract address on blockchain.

## Methods

### listenNewContentAvailable

```
public void listenNewContentAvailable(java.util.function.BiConsumer callback)
```

Subscribe a callback for new content available events.

**Parameters:**

`callback` - a BiConsumer of content name and address.

(continued from last page)

## listenNewContentAvailable

```
public void listenNewContentAvailable(java.lang.Runnable callback)
```

Subscribe a callback for new content available events.

### Parameters:

callback - a Runnable.

---

## listenAccessGranted

```
public void listenAccessGranted(java.util.function.BiConsumer callback)
```

Subscribe a callback for access granted events for this user.

### Parameters:

callback - a BiConsumer of content address and content name.

---

## listenAccessGranted

```
public void listenAccessGranted(java.lang.String user,  
    java.util.function.BiConsumer callback)
```

Subscribe a callback for access granted events for the specified user.

### Parameters:

user - the user for which be registered.

callback - a BiConsumer of content address and content name.

---

## listenBecomesPremium

```
public void listenBecomesPremium(java.lang.Runnable callback)
```

Subscribe a callback for becomes premium events for this user.

### Parameters:

callback - a Runnable.

---

## listenBecomesPremium

```
public void listenBecomesPremium(java.lang.String user,  
    java.lang.Runnable callback)
```

Subscribe a callback for becomes premium events for this user.

### Parameters:

user - the user for which be registered.

callback - a Runnable.

---

## listenFeedbackAvailable

```
public void listenFeedbackAvailable(java.util.function.BiConsumer callback)
```

Subscribe a callback for feedback available events for this user.

### Parameters:

callback - a BiConsumer of content address and content name.

---

(continued from last page)

## listenPaymentAvailable

```
public void listenPaymentAvailable(java.lang.String content,  
    java.util.function.BiConsumer callback)
```

Subscribe a callback for payment available events for this user.

### Parameters:

content - the content of which listen to.

callback - a BiConsumer of content address and content name.

---

## listenCatalogClosed

```
public void listenCatalogClosed(java.lang.Runnable callback)
```

Subscribe a callback for payment available events for this user.

### Parameters:

callback - a Runnable.

---

## hasAccess

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

Check if the user has access to a content.

### Parameters:

address - the content address.

### Returns:

boolean if has access, false otherwise.

---

## hasAccess

```
public boolean hasAccess(java.lang.String address,  
    java.lang.String user)
```

Check if the specified user has access to a content.

### Parameters:

address - the content address.

user - the user address.

### Returns:

boolean if has access, false otherwise.

---

## buyContent

```
public boolean buyContent(java.lang.String address,  
    java.math.BigInteger price)
```

Buy a content.

### Parameters:

address - the content address.

price - the content price.

### Returns:

a boolean representing the operation outcome.

---

## giftContent

```
public boolean giftContent(java.lang.String address,  
                           java.lang.String user,  
                           java.math.BigInteger price)
```

Gift a content to another user.

### Parameters:

address - the content address.  
user - the user address.  
price - the content price.

### Returns:

a boolean representing the operation outcome.

---

## buyPremium

```
public boolean buyPremium()
```

Buy a premium subscription.

### Returns:

a boolean representing the operation outcome.

---

## giftPremium

```
public boolean giftPremium(java.lang.String user)
```

Gift a premium subscription to another user.

### Parameters:

user - the user address.

### Returns:

a boolean representing the operation outcome.

---

## isPremium

```
public boolean isPremium()
```

Return true if the user has an active premium subscription.

### Returns:

a boolean representing the operation outcome.

---

## vote

```
public void vote(java.lang.String content,  
                int enjoy,  
                int valueForMoney,  
                int contentMeaning)
```

Set rating for a content.

### Parameters:

content - the content address.  
enjoy - the rating for the enjoy category.

---



(continued from last page)

valueForMoney - the rating for the value for money category.

contentMeaning - the rating for the content meaning category.

---

## getContentInfo

```
public Content getContentInfo(java.lang.String address)
```

Returns all the info and ratings of a content.

**Parameters:**

address - the content address.

**Returns:**

a list of Content objects.

---

## getContentList

```
public java.util.List getContentList()
```

Returns a list of all contents in the Catalog.

**Returns:**

a list of Content objects.

---

## getContentListWithViews

```
public java.util.List getContentListWithViews()
```

Returns a list of all contents in the Catalog and its views.

**Returns:**

a list of Content objects.

---

## getAuthorContents

```
public java.util.List getAuthorContents(java.lang.String author)
```

Return the list of all the content of a given author.

**Parameters:**

author - the authors address.

**Returns:**

a list of Content objects.

---

## getNewContentList

```
public java.util.List getNewContentList(int n)
```

Return the n latest releases.

**Parameters:**

n - the number of item that you want in the list.

**Returns:**

List of Content objects with the latest n contents.

---

(continued from last page)

## getLatest

```
public Content getLatest()
```

Return the latest release.

**Returns:**

String[] where the first element is the name of the content and the second is the address.

---

## getLatestByGenre

```
public Content getLatestByGenre(java.lang.String genre)
```

Return the latest release for a genre.

**Parameters:**

genre - the chosen genre.

**Returns:**

String[] where the first element is the name of the content and the second is the address.

---

## getLatestByAuthor

```
public Content getLatestByAuthor(java.lang.String author)
```

Return the latest release of an author.

**Parameters:**

author - the author address.

**Returns:**

String[] where the first element is the name of the content and the second is the address.

---

## getMostPopular

```
public Content getMostPopular()
```

Return the most popular content.

**Returns:**

Content.

---

## getMostPopularByGenre

```
public Content getMostPopularByGenre(java.lang.String genre)
```

Return the most popular content for a genre.

**Parameters:**

genre - the chosen genre.

**Returns:**

String[] where the first element is the name of the content and the second is the address.

---

## getMostPopularByAuthor

```
public Content getMostPopularByAuthor(java.lang.String author)
```

(continued from last page)

Return the most popular content of an author.

**Parameters:**

author - the author address.

**Returns:**

String[] where the first element is the name of the content and the second is the address.

---

## getMostRated

```
public Content getMostRated(java.lang.String category)
```

Return the highest rated content.

**Parameters:**

category - the category name for which you want to know the rating.

**Returns:**

Content.

---

## getMostRatedByGenre

```
public Content getMostRatedByGenre(java.lang.String genre,  
    java.lang.String category)
```

Return the highest rated content for a genre.

**Parameters:**

genre - the chosen genre.

category - the category name for which you want to know the rating.

**Returns:**

Content.

---

## getMostRatedByAuthor

```
public Content getMostRatedByAuthor(java.lang.String author,  
    java.lang.String category)
```

Return the highest rated content of an author.

**Parameters:**

author - the author address.

category - the category name for which you want to know the rating.

**Returns:**

Content.

---

## withdraw

```
public java.math.BigInteger withdraw(java.lang.String address)
```

Collect the payout for a content. Can be called only on content of the current user.

**Parameters:**

address - the address of the content.

**Returns:**

(continued from last page)

a BigInteger with the withdrawn amount, 0 otherwise (for example if the specified content is not owned by the user that have done the login).

## com.aldodaquino.cobra.main Class Content

```
java.lang.Object
  |
  +--com.aldodaquino.cobra.main.Content
```

---

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

Defines a content object.

**Author:**  
Aldo D'Aquino.

**Version:**  
1.0.

---

### Fields

#### address

```
public final java.lang.String address
```

---

#### name

```
public final java.lang.String name
```

---

#### author

```
public final java.lang.String author
```

---

#### genre

```
public final java.lang.String genre
```

---

#### price

```
public final java.math.BigInteger price
```

---

#### views

```
public final java.math.BigInteger views
```

---

---

## averageRating

public final int **averageRating**

---

## enjoy

public final int **enjoy**

---

## priceFairness

public final int **priceFairness**

---

## contentMeaning

public final int **contentMeaning**

---

## com.aldodaquino.cobra.main Class ContentManager

```
java.lang.Object
|
+--com.aldodaquino.cobra.main.ContractManager
|
+--com.aldodaquino.cobra.main.ContentManager
```

---

public class **ContentManager**  
extends ContractManager

An higher level Content Manager. Contains methods that call the methods in the `com.aldodaquino.cobra.contracts.DAPPContentManagementContract` generated by Web3j and parse and aggregate the result in a more comfortable representation.

**Author:**

Aldo D'Aquino.

**Version:**

1.0.

---

## Constructors

### ContentManager

```
public ContentManager(org.web3j.crypto.Credentials credentials,  
                      java.lang.String catalogAddress,  
                      java.lang.String name,  
                      java.lang.String genre,  
                      java.math.BigInteger price,  
                      java.lang.String hostname,  
                      int port)
```

Deploy and manage a new content manager contract.

**Parameters:**

credentials - your account credentials.  
catalogAddress - the address of the catalog.  
name - the name you want to assign to this content.  
genre - the genre you want to assign to this content.  
price - the price you want to assign to this content.  
hostname - the hostname of the author server.  
port - the port on which is running the author server.

**Throws:**

Exception - if there is errors when deploying the Content contract.

---

### ContentManager

```
public ContentManager(org.web3j.crypto.Credentials credentials,  
                      java.lang.String contractAddress)
```

Load and manage an existent content manager contract.

**Parameters:**

credentials - your account credentials.  
contractAddress - the existent contract address on blockchain.

---

(continued from last page)

## Methods

### listenContentPublished

```
public void listenContentPublished(java.lang.Runnable callback)
```

Subscribe a callback for content published events.

**Parameters:**

callback - a Runnable.

---

### consumeContent

```
public boolean consumeContent()
```

Consume a bought content. Note that the content is consumed by the user that owns the credentials passed to the constructor.

**Returns:**

a boolean representing the operation outcome.

---

### getHostname

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

Returns the hostname of the author-server.

**Returns:**

String the hostname.

---

### getPort

```
public int getPort()
```

Returns the port number of the author-server.

**Returns:**

the int number of the port.

---



---

**Package**

**com.aldodaquino.javautils**

## com.aldodaquino.javautils

### Class CliHelper

```
java.lang.Object
|
+--com.aldodaquino.javautils.CliHelper
```

---

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

Help parsing the args[]. You can add the option you want retrieve and this class will parse it automatically.

**Author:**

Aldo D'Aquino.

**Version:**

1.0.

---

## Constructors

### CliHelper

```
public CliHelper()
```

## Methods

### addOption

```
public void addOption(java.lang.String shortOpt,
                      java.lang.String longOpt,
                      boolean hasValue,
                      java.lang.String description)
```

Add available option for the CLI.

**Parameters:**

shortOpt - like "-v".

longOpt - like "--verbose".

hasValue - true if the option must have a value (i.e. -o value), false if is an option without value (i.e. --help).

description - the description of the option to be shown in the help message.

**Throws:**

IllegalArgumentException - if the shortOpt or the longOpt already exists in another option.

---

### parse

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

Parse the String[] args of the main and saves the option value.

**Parameters:**

args - the main's args.

---

(continued from last page)

---

## getValues

```
public java.lang.String[] getValues(java.lang.String opt)
```

Return the list of all values of an option. For example -o value1 -o value2 will return [value1, value2].

### Parameters:

opt - the option in the short or the long format (will return the same list).

### Returns:

a String[] containing all the values, empty array if the option has no value or null if this option doesn't exist.

---

## getValue

```
public java.lang.String getValue(java.lang.String opt)
```

Return first value of an option. For example -o value1 -o value2 will return value1. Use it for functions that are intended usable only once.

### Parameters:

opt - the option in the short or the long format (will return the same list).

### Returns:

the first value, empty string if the option has no value or null if this option doesn't exist.

---

## isPresent

```
public boolean isPresent(java.lang.String opt)
```

Return true if the program is launched with the specified option, false otherwise.

### Parameters:

opt - the option in the short or the long format (will return the same list).

### Returns:

true if the program is launched with the specified option, false otherwise.

---

## getHelpMessage

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

Return a formatted help message showing the usage. The message has this format: usage: -o --longopt an option description  
-h --help shows help

### Returns:

String of the message.

---

## getMissingOptionMessage

```
public java.lang.String getMissingOptionMessage(java.lang.String opt)
```

Return a missing option message with this format: Missing an option: -o -option description of the option.

### Parameters:

opt - the short or long code of the option that you want (i.e. "o");

### Returns:

String of the message.

---

## com.aldodaquino.javautils

### Class FileExchange

```
java.lang.Object
|
+--com.aldodaquino.javautils.FileExchange
```

---

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

Utility for exchanging file. Contains method to receive data from a Socket and save it to a File and to read a file and write data to the socket.

**Author:**  
Aldo D'Aquino.

**Version:**  
1.2.

---

## Constructors

### FileExchange

```
public FileExchange()
```

## Methods

### receiveFile

```
public static void receiveFile(java.io.File destFile,
                                java.lang.String hostname,
                                int port)
```

Receive and save a file.

**Parameters:**

destFile - the destination file where save data.  
hostname - of the sender.  
port - of the sender.

---

### openFileSocket

```
public static java.nio.channels.ServerSocketChannel openFileSocket()
```

Opens a socket on which waits an incoming connection from the file recipient.

**Returns:**

a ServerSocketChannel.

---

### startFileSender

```
public static void startFileSender(java.nio.channels.ServerSocketChannel serverSocket,
                                     java.io.File file,
                                     java.lang.Runnable callback)
```

(continued from last page)

When the other client has connected to the socket it behaves as a server and sends the file.

**Parameters:**

serverSocket - the ServerSocketChannel opened in openFileSocket.  
file - the File to be sent.  
callback - optional callback to be run when the upload is finished.

---

## readFile

```
public static void readFile(java.io.File file,  
                             java.nio.channels.SocketChannel outChannel)  
    throws java.io.IOException
```

Read a File using NIO channels. Send the file to a socket.

**Parameters:**

file - File object, the file to be read.  
outChannel - Socket where the file will be sent.

**Throws:**

IOException - if the file not exists or is not readable.

---

## writeFile

```
public static void writeFile(java.nio.channels.SocketChannel inChannel,  
                              java.io.File file)  
    throws java.io.IOException
```

Read the specified Socket using NIO, and save the data to File.

**Parameters:**

inChannel - Socket from where data will be read.  
file - File object, destination of the data.

**Throws:**

IOException - if the file not exists or is not writeable.

---

# com.aldodaquino.javautils

## Class HttpHelper

java.lang.Object

└--com.aldodaquino.javautils.HttpHelper

Direct Known Subclasses:

[CobraHttpHelper](#)

---

public class **HttpHelper**  
extends java.lang.Object

Contains method that help to make http request. Works with JSON body for POST request and query-style GET parameters.

**Author:**

Aldo D'Aquino.

**Version:**

1.1.

---

## Constructors

### HttpHelper

public **HttpHelper**()

## Methods

### parseGET

public static java.util.Map **parseGET**(com.sun.net.httpserver.HttpExchange request)

Parse a GET request and return a Map with keys equals to parameters name and values equals to the parameters values.

**Parameters:**

request - the HttpExchange request received by the handler.

**Returns:**

Map where both keys and values are strings containing the parameters.

---

### parseQuery

public static java.util.Map **parseQuery**(java.lang.String query)

Parse a query and return a Map with keys equals to parameters name and values equals to the parameters values.

**Parameters:**

query - the String representing the query.

**Returns:**

Map where both keys and values are strings containing the parameters.

---

(continued from last page)

## parsePOST

```
public static java.util.Map parsePOST(com.sun.net.httpserver.HttpExchange request)
```

Parse a POST request and return a Map with keys equals to parameters name and values equals to the parameters values.

**Parameters:**

request - the HttpExchange request received by the handler.

**Returns:**

Map where both keys and values are strings containing the parameters.

---

## parseJson

```
public static java.util.Map parseJson(java.lang.String json)
```

Parse a JSON and return a Map with keys equals to parameters name and values equals to the parameters values.

**Parameters:**

json - the String representing the stringified JSON.

**Returns:**

Map where both keys and values are strings containing the parameters.

---

## sendResponse

```
public static void sendResponse(com.sun.net.httpserver.HttpExchange request,  
    java.lang.String response,  
    int code)
```

Send a response to an HttpExchange request.

**Parameters:**

request - the request.

response - a String containing the response.

code - the status code of the response.

---

## sendResponse

```
public static void sendResponse(com.sun.net.httpserver.HttpExchange request,  
    java.lang.String response)
```

Send a response to an HttpExchange request.

**Parameters:**

request - the request.

response - a String containing the response..

---

## makeGet

```
public static HttpHelper.Response makeGet(java.lang.String url,  
    java.util.Map parameters)
```

Make a GET request on the specified url.

**Parameters:**

url - the url to be called.

parameters - a map containing all the parameters that you want to be passed when the url is called.

---

(continued from last page)

**Returns:**a [HttpHelper.Response](#) object.

---

## makePost

```
public static HttpHelper.Response makePost(java.lang.String url,  
      java.util.Map parameters)
```

Make a POST request on the specified url.

**Parameters:**

url - the url to be called.

parameters - a map containing all the parameters that you want to be passed in the body.

**Returns:**a [HttpHelper.Response](#) object.

---

## jsonifyParameters

```
public static java.lang.String jsonifyParameters(java.util.Map parameters)
```

Return a stringified JSON with the passed parameters.

**Parameters:**

parameters - a Map where both keys and values are strings containing the parameters.

**Returns:**

the stringified JSON.

---

## querifyParameters

```
public static java.lang.String querifyParameters(java.util.Map parameters)
```

Return a string containing the parameters in the query format, ready to be appended to an url for a GET request.

**Parameters:**

parameters - a Map where both keys and values are strings containing the parameters.

**Returns:**

the String in the url format.



## com.aldodaquino.javautils

# Class `HttpHelper.Response`

`java.lang.Object`

└─`com.aldodaquino.javautils.HttpHelper.Response`

public static class **`HttpHelper.Response`**  
extends `java.lang.Object`

Response class returned by the `makeGet` and `makePost` method. Contains two field: the response code and the data String of the response.

**Author:**  
Aldo D'Aquino.

**Version:**  
1.0.

## Fields

### **code**

public final int **code**

### **data**

public final java.lang.String **data**

## Methods

### **toString**

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

Returns this object as a stringified JSON.

**Returns:**  
a String representing the stringified JSON.

# Index

## A

ACCESS\_API\_PATH 5  
addOption 66  
address 61  
appName 44  
AsyncPanel 17  
author 61  
AuthorContentTable 19  
AuthorPanel 47  
averageRating 62

## B

biggestPriceFairnessLabel 26  
buyContent 55  
buyPremium 56

## C

CatalogForm 20  
CatalogManager 53  
catalogManager 7  
ChartWidget 21  
CliHelper 66  
CobraHttpHelper 6  
code 73  
Colors 39  
ComponentFactory 22  
connectCatalog 12  
consumeContent 64  
ContentList 24  
ContentManager 63  
contentMeaning 62  
credentials 7, 11  
CustomerPanel 49

## D

data 73  
DEPLOY\_API\_PATH 5  
deployCatalog 12  
Dimensions 41

disconnectCatalog 12  
doAsync 17

## E

emptyStar 42  
enjoy 62

## F

FileExchange 68  
filledStar 42

## G

genre 61  
getAuthorContents 57  
getCatalogManager 13  
getContentInfo 57  
getContentList 57  
getContentListWithViews 57  
getHelpMessage 67  
getHostname 64  
getLatest 57  
getLatestByAuthor 58  
getLatestByGenre 58  
getMinimumSize 51  
getMissingOptionMessage 67  
getMostPopular 58  
getMostPopularByAuthor 58  
getMostPopularByGenre 58  
getMostRated 59  
getMostRatedByAuthor 59  
getMostRatedByGenre 59  
getNewContentList 57  
getPort 64  
getPrivateKey 13  
getRating 32  
getScaled 42  
getUserAddress 12  
getValue 67  
getValues 66  
giftContent 56  
giftPremium 56  
GREEN 39

## H

hasAccess 55  
highestContentMeaningLabel 26  
highestRatedLabel 25  
hostname 7  
HttpHelper 70

## I

Images 42  
INFO\_PANEL\_PADDING 40  
InfoPanel 26  
isCatalogOwner 12  
isCellEditable 19, 37  
isPremium 56  
isPresent 67

## J

jsonifyParameters 72

## L

LATERAL\_BAR\_PADDING 40  
latestLabel 25  
listenAccessGranted 54  
listenBecomesPremium 54  
listenCatalogClosed 55  
listenContentPublished 63  
listenFeedbackAvailable 54  
listenNewContentAvailable 53  
listenPaymentAvailable 54  
loading 42  
login 11  
LoginForm 28  
Logo 29  
logo 42  
LOGO\_SIZE 40

## M

Main 3, 10  
main 3, 10

makeGet 71  
makePost 72  
mostEnjoyedLabel 25  
mostPopularLabel 25

## N

name 61  
newBorder 22  
newButton 22  
newConfirmDialog 15  
NewContentsWidget 30  
newErrorDialog 14  
newExitDialog 15  
newGBC 34  
newHandler 6  
newMessageDialog 14  
newTextField 22  
newVSpacer 23  
newWindow 14, 46, 48, 50

## O

openFileDialog 15  
openFileSocket 68

## P

parse 66  
parseGET 70  
parseJson 71  
parsePOST 70  
parseQuery 70  
port 7  
price 61  
priceFairness 62  
privateKey 7

## Q

querifyParameters 72

## R

readFile 69

receiveFile 68  
RED 39  
replaceComponent 34  
ROLE\_AUTHOR 11  
ROLE\_CUSTOMER 11  
RoleForm 31

withdraw 59  
writeFile 69

## S

saveFileDialog 15  
sendResponse 71  
setFontSize 15  
setRole 13  
STAR\_SIZE 40  
StarPanel 32  
STARTER\_PANEL\_PADDING 40  
StarterPanel 51  
startFileSender 68  
Status 7, 11  
Strings 44

## T

toString 73

## U

update 27  
updateStatus 36  
UpgradablePanel 34  
UserInfo 36  
Utils 14

## V

V\_SPACER\_L 41  
V\_SPACER\_M 41  
V\_SPACER\_S 40  
views 61  
ViewsContentTable 37  
vote 56

## W

window 17