

# **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.

### Fields

#### DEFAULT\_PORT

```
private static final int DEFAULT_PORT
```

Constant value: **8080**

#### CONTENT\_FILE\_PATH

```
private static final java.lang.String CONTENT_FILE_PATH
```

Constant value: **author\_content\_files/**

### 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:**

(continued on next page)

(continued from last page)

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

---

## deploy

```
private static void deploy(com.sun.net.httpserver.HttpExchange request,  
    Status status)
```

Handler for the /deploy url.

### Parameters:

request - a POST request with JSON encoded data containing: privateKey of the author, name of the content, genre of the content (can be null), price of the content (if null is set to 0), port on which is running the server socket that uploads the file.  
status - the server [object](#).

---

## access

```
private static void access(com.sun.net.httpserver.HttpExchange request,  
    Status status)
```

Handler for the /access url.

### Parameters:

request - a POST request with JSON encoded data containing: privateKey of the author, name of the content, genre of the content (can be null), price of the content (if null is set to 0).  
status - the server [object](#).

---

**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 CobraHttpHelper.HttpRequestHandler

java.lang.Object

└-com.aldodaquino.cobra.connections.CobraHttpHelper.HttpRequestHandler

### All Implemented Interfaces:

com.sun.net.httpserver.HttpHandler

---

private static class **CobraHttpHelper.HttpRequestHandler**  
extends java.lang.Object  
implements com.sun.net.httpserver.HttpHandler

## Fields

### consumer

final java.util.function.BiConsumer **consumer**

### status

final com.aldodaquino.cobra.connections.Status **status**

## Constructors

### HttpRequestHandler

**HttpRequestHandler**(java.util.function.BiConsumer consumer,  
[Status](#) status)

## Methods

### handle

public void **handle**(com.sun.net.httpserver.HttpExchange request)



## 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.

---

## Fields

### window

```
private static javax.swing.JFrame window
```

## 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.

---

### setContent

```
private static void setContent(javax.swing.JPanel replacement)
```

---

### showMainPanel

```
private static void showMainPanel(Status status)
```

---

## 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

#### **privateKey**

```
private java.lang.String privateKey
```

#### **credentials**

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

#### **catalogManager**

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

#### **role**

```
private int role
```

(continued from last page)

## 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.

**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
```

(continued from last page)

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.

**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.

---

## getRole

```
int getRole()
```

Returns the role for the current user.

**Returns:**

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.

---

## newDialog

```
private static void newDialog(java.lang.String title,  
    java.lang.String msg,  
    int type)
```

---

## 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.

---

(continued from last page)

**Parameters:**

defaultName - specify the original filename of the incoming file.

**Returns:**

a File.

---

## fileDialog

```
private static java.io.File fileDialog(boolean isOpenDialog,  
    java.lang.String filename)
```

---

**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](#), [DeployContentPanel](#), [PickUserPanel](#), [VotingPanel](#), [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.

---

**startLoading**

```
private static void startLoading(javax.swing.JFrame window)
```

---

**stopLoading**

```
private static void stopLoading(javax.swing.JFrame window)
```

## 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.

## Fields

### colNames

```
private static final java.lang.String colNames
```

## Constructors

### AuthorContentTable

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

Constructor.

#### Parameters:

status - the Status object.

contents - a List of Content objects.

## Methods

### prepareRows

```
private static java.lang.Object[][] prepareRows(java.util.List contents)
```

(continued from last page)

---

## **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.

## Fields

### catalogAddressField

```
private final javax.swing.JTextField catalogAddressField
```

### connectCallback

```
private final java.util.function.Consumer connectCallback
```

## 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.

## Methods



(continued from last page)

## **connect**

```
private void connect()
```

## 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)

## newPasswordField

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

Returns a new JTextField for password. Characters are replaced with dots.

**Parameters:**

e - the ActionListener called when enter is pressed.

**Returns:**

a JTextField.

---

## newField

```
private static javax.swing.JTextField newField(java.awt.event.ActionListener e,  
boolean isPassword)
```

---

## 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.

---

## newTitledBorderPanel

```
static javax.swing.JPanel newTitledBorderPanel(java.lang.String title)
```

Returns a panel with a border with the specified title centered and a vertical layout.

**Parameters:**

title - the title string.

**Returns:**

a JPanel.

---

(continued from last page)

## **newSpinner**

```
static javax.swing.JPanel newSpinner()
```

Return a new panel with a centered JLabel containing a loading message with a spinner. Used in the [AsyncPanel](#).

**Returns:**

a JPanel.

## 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.

## Methods

### prepareRows

```

private static java.lang.String[] prepareRows(java.util.List contents)
  
```

## 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.

## Fields

### loader

```
private final javax.swing.JLabel loader
```

### replacingPosition

```
private final java.awt.GridBagConstraints replacingPosition
```

### status

```
private final com.aldodaquino.cobra.gui.Status status
```

## Constructors

(continued from last page)

## LabelPanel

```
LabelPanel(Status status,  
          java.lang.String label)
```

Constructor.

### Parameters:

status - the Status object.  
label - the label for the value.

## 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.

## Fields

### privateKeyInput

```
private final javax.swing.JTextField privateKeyInput
```

### loginCallback

```
private final java.util.function.Consumer loginCallback
```

## 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.

## Methods

(continued from last page)

## **login**

```
private void login()
```

Submit form action. Called when the login button is clicked or enter key is pressed from the private key field.

## 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.

---

## Fields

### status

```
private final com.aldodaquino.cobra.gui.Status status
```

---

### catalogManager

```
private final com.aldodaquino.cobra.main.CatalogManager catalogManager
```

---

### numberSpinner

```
private final javax.swing.JSpinner numberSpinner
```

---

## Constructors

### NewContentsWidget

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

Constructor.

---

(continued from last page)

**Parameters:**

status - the Status object.

## Methods

**getNewContentList**

```
private void getNewContentList()
```

## 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.

## Fields

### STARS\_NUMBER

```
private static final int STARS_NUMBER
```

Constant value: 5

### rating

```
private int rating
```

### enabled

```
private final boolean enabled
```

### stars

```
private final java.util.List stars
```

## 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.

---

### StarPanel

```
private StarPanel(int rating,  
                  boolean enabled)
```

Constructor.

**Parameters:**

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

enabled - true if users can vote, false to show only the rating.

---

## Methods

### getRating

```
public int getRating()
```

Returns the rating.

**Returns:**

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

---

### setRating

```
private void setRating(int rating)
```

---

### getStar

```
private javax.swing.JLabel getStar(boolean filled)
```

## 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.

---

## Fields

### status

```
private final com.aldodaquino.cobra.gui.Status status
```

---

### catalogAddressLabel

```
private javax.swing.JLabel catalogAddressLabel
```

---

### catalogAddressPosition

```
private final java.awt.GridBagConstraints catalogAddressPosition
```

---

### accountAddressLabel

```
private javax.swing.JLabel accountAddressLabel
```

---

## accountAddressPosition

```
private final java.awt.GridBagConstraints accountAddressPosition
```

---

## premiumLabel

```
private javax.swing.JLabel premiumLabel
```

---

## premiumPosition

```
private final java.awt.GridBagConstraints premiumPosition
```

## 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.

---

### newAccountAddressLabel

```
private javax.swing.JLabel newAccountAddressLabel()
```

---

### newCatalogAddressLabel

```
private javax.swing.JLabel newCatalogAddressLabel()
```

---

### newPremiumLabel

```
private javax.swing.JLabel newPremiumLabel()
```

---

## 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.

## Fields

### colNames

```
private static final java.lang.String colNames
```

## 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

### prepareRows

```
private static java.lang.Object[][] prepareRows(java.util.List contents)
```

(continued from last page)

---

## **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.

## Fields

### WINDOW\_TITLE

```
static final java.lang.String WINDOW_TITLE
```

Constant value: **About the author**

## Constructors

### AuthorInfoPanel

```
AuthorInfoPanel(Status status,  
                 java.lang.String author)
```

Constructor.

#### Parameters:

status - the Status object.  
 author - the author address.

## Methods

(continued from last page)

## **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.

## Fields

### status

```
private final com.aldodaquino.cobra.gui.Status status
```

### catalogManager

```
private final com.aldodaquino.cobra.main.CatalogManager catalogManager
```

### tableContainer

```
private final javax.swing.JScrollPane tableContainer
```

### table

```
private javax.swing.JTable table
```

## Constructors

(continued from last page)

## AuthorPanel

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

Constructor.

### Parameters:

status - the Status object.

## Methods

### deployContent

```
private void deployContent()
```

---

### onDeployed

```
private void onDeployed(java.lang.String address)
```

---

### updateTable

```
private void updateTable()
```

---

### withdrawSelected

```
private void withdrawSelected()
```

## 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.

---

## Fields

### WINDOW\_TITLE

```
private static final java.lang.String WINDOW_TITLE
```

Constant value: **Content info**

---

### status

```
private final com.aldodaquino.cobra.gui.Status status
```

---

### catalogManager

```
private final com.aldodaquino.cobra.main.CatalogManager catalogManager
```

---

### content

```
private final com.aldodaquino.cobra.main.Content content
```

---

## contentManager

```
private final com.aldodaquino.cobra.main.ContentManager contentManager
```

## Constructors

### ContentInfoPanel

```
private ContentInfoPanel(Status status,  
                        java.lang.String address)
```

Constructor.

**Parameters:**

status - the Status object.  
address - the content address.

## 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.

---

### view

```
private void view()
```

---

### retrieveContent

```
private void retrieveContent()
```

---

### gift

```
private void gift()
```

---

### prepareLink

```
private javax.swing.JPanel prepareLink(java.lang.String label,  
                                       java.lang.String value,  
                                       javax.swing.JPanel onClickPanel,  
                                       java.lang.String windowTitle)
```

---

## prepareStar

```
private javax.swing.JPanel prepareStar(java.lang.String label,  
    int rating)
```

## 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.

---

## Fields

### status

```
private final com.aldodaquino.cobra.gui.Status status
```

---

### catalogManager

```
private final com.aldodaquino.cobra.main.CatalogManager catalogManager
```

---

### tableContainer

```
private final javax.swing.JScrollPane tableContainer
```

---

### table

```
private java.awt.Component table
```



---

## lateralBar

```
private final javax.swing.JPanel lateralBar
```

---

---

## userInfo

```
private final com.aldodaquino.cobra.gui.components.UserInfo userInfo
```

---

---

## chartWidget

```
private com.aldodaquino.cobra.gui.components.ChartWidget chartWidget
```

---

---

## chartWidgetPosition

```
private final java.awt.GridBagConstraints chartWidgetPosition
```

---

---

## showViews

```
private boolean showViews
```

---

## Constructors

### CustomerPanel

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

Constructor.

**Parameters:**

status - the Status object.

## Methods

### update

```
private void update()
```

---

---

### getTable

```
private java.awt.Component getTable()
```

---

(continued from last page)

## **buyPremium**

```
private void buyPremium()
```

---

## **giftPremium**

```
private void giftPremium()
```

## com.aldodaquino.cobra.gui.panels

### Class DeployContentPanel

```
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.DeployContentPanel
```

#### All Implemented Interfaces:

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

---

class **DeployContentPanel**  
extends [AsyncPanel](#)

Panel to deploy a new content.

#### Author:

Aldo D'Aquino.

#### Version:

1.0.

---

## Fields

### addressField

```
private final javax.swing.JTextField addressField
```

---

### portField

```
private final javax.swing.JTextField portField
```

---

### nameField

```
private final javax.swing.JTextField nameField
```

---

### genreField

```
private final javax.swing.JTextField genreField
```

---

(continued from last page)

---

## priceField

```
private final javax.swing.JTextField priceField
```

---

## status

```
private final com.aldodaquino.cobra.gui.Status status
```

---

## deployCallback

```
private final java.util.function.Consumer deployCallback
```

---

## file

```
private java.io.File file
```

# Constructors

## DeployContentPanel

```
DeployContentPanel(Status status,  
                    java.util.function.Consumer deployCallback)
```

Constructor.

### Parameters:

status - the Status object.

deployCallback - a Consumer of content address, called after that the content has been deployed.

# Methods

## deploy

```
private void deploy()
```

Submit form action. Called when the deploy button is clicked.

## 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.

## Fields

### WINDOW\_TITLE

```
static final java.lang.String WINDOW_TITLE
```

Constant value: **About the genre**

## Constructors

### GenreInfoPanel

```
GenreInfoPanel(Status status,
               java.lang.String genre)
```

Constructor.

#### Parameters:

status - the Status object.

genre - the genre.

## Methods

(continued from last page)

## 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 PickUserPanel

```

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.PickUserPanel
  
```

#### All Implemented Interfaces:

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

class **PickUserPanel**  
 extends [AsyncPanel](#)

Asks the user for another user address.

#### Author:

Aldo D'Aquino.

#### Version:

1.0.

## Fields

### addressField

```
private final javax.swing.JTextField addressField
```

### giftCallback

```
private final java.util.function.Consumer giftCallback
```

## Constructors

### PickUserPanel

```
PickUserPanel(java.util.function.Consumer giftCallback)
```

Constructor.

#### Parameters:

`giftCallback` - a Consumer of user address, invoked when the button is clicked or enter is pressed if the address has a valid format.

## Methods

(continued from last page)

**gift**

```
private void gift()
```

Submit form action. Called when the gift button is clicked or enter key is pressed from the address field.



## 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.

## Fields

### status

```
private final com.aldodaquino.cobra.gui.Status status
```

### whenDone

```
private final java.util.function.Consumer whenDone
```

### userInfo

```
private final com.aldodaquino.cobra.gui.components.UserInfo userInfo
```

### loginForm

```
private final javax.swing.JPanel loginForm
```

---

## catalogForm

```
private javax.swing.JPanel catalogForm
```

---

## roleForm

```
private javax.swing.JPanel roleForm
```

---

---

## replacingPosition

```
private final java.awt.GridBagConstraints replacingPosition
```

## 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()
```

---

### loginCallback

```
private void loginCallback(java.lang.String privateKey)
```

---

### connectCallback

```
private void connectCallback(java.lang.String catalogAddress)
```

---

### deployCallback

```
private void deployCallback()
```

---

(continued from last page)

## **postConnect**

```
private void postConnect()  
    throws javax.naming.OperationNotSupportedException
```

---

## **disconnectCallback**

```
private void disconnectCallback()
```

---

## **deleteCallback**

```
private void deleteCallback()
```

---

## **browseCallback**

```
private void browseCallback()
```

---

## **manageCallback**

```
private void manageCallback()
```

---

## com.aldodaquino.cobra.gui.panels

### Class VotingPanel

```
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.VotingPanel
```

#### All Implemented Interfaces:

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

class **VotingPanel**  
extends [AsyncPanel](#)

A panel to ask the user to vote a content consumed recently.

**Author:**

Aldo D'Aquino.

**Version:**

1.0.

## Constructors

### VotingPanel

```
VotingPanel( java.lang.String contentName,  
             java.lang.String contentAddress,  
             CatalogManager catalogManager)
```

Constructor.

**Parameters:**

contentName - the content name.

contentAddress - the content address.

catalogManager - the CatalogManager loaded or deployed by the user.

---

**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.

---

### Fields

#### **catalog**

```
private final com.aldodaquino.cobra.contracts.CatalogContract catalog
```

---

#### **newContentAvailableBiConsumers**

```
private final java.util.List newContentAvailableBiConsumers
```

---

#### **accessGrantedMap**

```
private final java.util.Map accessGrantedMap
```

---

#### **becomesPremiumMap**

```
private final java.util.Map becomesPremiumMap
```

---

#### **feedbackAvailableBiConsumer**

```
private final java.util.List feedbackAvailableBiConsumer
```

---

#### **paymentAvailableMap**

```
private final java.util.Map paymentAvailableMap
```

---

(continued from last page)

---

## catalogClosedRunnables

```
private final java.util.List catalogClosedRunnables
```

## 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

### listenForEvents

```
private void listenForEvents()
```

---

### getName

```
private java.lang.String getName(java.lang.String contentAddress)
```

---

### 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.

---

### listenNewContentAvailable

```
public void listenNewContentAvailable(java.lang.String authorOrGenre,  
    java.util.function.BiConsumer callback)
```

(continued from last page)

Subscribe a callback for new content available events for a specified genre or author.

**Parameters:**

authorOrGenre - the genre or the author address for which listen to.  
callback - a BiConsumer of content name and address.

---

## 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.

---

## listenPaymentAvailable

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

---



(continued from last page)

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.

---

(continued from last page)

## 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.  
valueForMoney - the rating for the value for money category.

---

(continued from last page)

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 CatalogManager.ContentList

java.lang.Object

└-com.aldodaquino.cobra.main.CatalogManager.ContentList

```
private class CatalogManager.ContentList
extends java.lang.Object
```

### Fields

#### addresses

java.util.List **addresses**

#### names

java.util.List **names**

#### authors

java.util.List **authors**

#### genres

java.util.List **genres**

#### prices

java.util.List **prices**

#### views

java.util.List **views**

#### averageRatings

java.util.List **averageRatings**

---

## enjoyRatings

`java.util.List` **enjoyRatings**

---

---

## priceFairnessRatings

`java.util.List` **priceFairnessRatings**

---

---

## contentMeaningRatings

`java.util.List` **contentMeaningRatings**

---

## Constructors

### ContentList

`ContentList()`

## Methods

### getFilteredContentList

`java.util.List` **getFilteredContentList**(`java.util.List` filterBy,  
`java.lang.Object` filterValue)

Returns a list of all contents that has the parameter where equals to value. The list is not filtered if where is null.

**Parameters:**

`filterBy` - a list of this class that can be addresses, names, authors, genres, prices or views.  
`filterValue` - the value that filterBy must have.

**Returns:**

a list of Content objects.



## 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
```

---

# Constructors

## Content

```
Content(java.lang.String address,  
        byte[] name)
```

---

## Content

```
Content(java.lang.String address,  
        byte[] name,  
        java.math.BigInteger views)
```

---

## Content

```
Content(java.lang.String address,  
        byte[] name,  
        java.lang.String author,  
        byte[] genre,  
        java.math.BigInteger price,  
        java.math.BigInteger views,  
        java.math.BigInteger averageRating,  
        java.math.BigInteger enjoy,  
        java.math.BigInteger priceFairness,  
        java.math.BigInteger 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.

### Fields

#### **content**

```
private final com.aldodaquino.cobra.contracts.DAPPContentManagementContract content
```

#### **contentPublishedRunnables**

```
private final java.util.List contentPublishedRunnables
```

### 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.

(continued from last page)

**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.

## 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.

---

## com.aldodaquino.cobra.main Class ContractManager

java.lang.Object

└-com.aldodaquino.cobra.main.ContractManager

Direct Known Subclasses:

[CatalogManager](#), [ContentManager](#)

---

class **ContractManager**

extends java.lang.Object

An higher level Contract Manager. Contains basic methods of all the contracts, such as the deploy, load and suicide function, but also the get owner and get address functions.

**Author:**

Aldo D'Aquino.

**Version:**

1.0.

---

### Fields

#### **web3**

private org.web3j.protocol.Web3j **web3**

---

#### **gasPrice**

private java.math.BigInteger **gasPrice**

---

#### **gasLimit**

private java.math.BigInteger **gasLimit**

---

#### **credentials**

org.web3j.crypto.Credentials **credentials**

---

#### **contract**

private org.web3j.tx.Contract **contract**

---

(continued from last page)

## contractClass

```
private java.lang.Class contractClass
```

## owner

```
private java.lang.String owner
```

## Constructors

### ContractManager

```
ContractManager(org.web3j.crypto.Credentials credentials)
```

Save credentials, connect to web3 and save the gas information.

**Parameters:**

credentials - your account credentials.

## Methods

### deploy

```
org.web3j.tx.Contract deploy(java.lang.Class contractClass)
```

Deploy a new contract of class contractClass and return the contractClass instance.

**Parameters:**

contractClass - the class of the contract that you want deploy.

**Returns:**

the deployed contract as contractClass instance.

### load

```
org.web3j.tx.Contract load(java.lang.Class contractClass,  
    java.lang.String contractAddress)
```

Load an existent contract of class contractClass and return the contractClass instance.

**Parameters:**

contractClass - the class of the contract that you want to load.

contractAddress - the address of the contract.

**Returns:**

the loaded contract as contractClass instance.

### getAddress

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

Returns the contract address.

**Returns:**

(continued from last page)

a string containing the contract address.

---

## **getOwner**

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

Returns the contract owner.

### **Returns:**

a string containing the contract owner.

---

## **suicide**

```
public boolean suicide()
```

Call the suicide function on the contract.

### **Returns:**

true if the contract suicide has been committed, false in case of errors.

## com.aldodaquino.cobra.main Class Utills

```
java.lang.Object
└--com.aldodaquino.cobra.main.Utills
```

---

class **Utills**  
extends java.lang.Object

Some utilities for the blockchain. Contains method to get gas information and to convert bytes32 to Strings and vice-versa.

**Author:**  
Aldo D'Aquino.

**Version:**  
1.0.

---

### Fields

#### **BLOCK\_GAS\_LIMIT**

```
private static final java.lang.String BLOCK_GAS_LIMIT
```

Constant value: **5000000**

### Constructors

#### **Utills**

```
Utills()
```

### Methods

#### **buildWeb3**

```
static org.web3j.protocol.Web3j buildWeb3()
```

Build a Web3j protocol using the default provider.

**Returns:**  
a Web3j instance.

---

#### **getGasPrice**

```
static java.math.BigInteger getGasPrice(org.web3j.protocol.Web3j web3)
```

Return the average gas price.

**Parameters:**  
web3 - a web3j instance.

**Returns:**  
a BigInteger of the gas price, 0 in case of error.

---



## getGasLimit

```
static java.math.BigInteger getGasLimit(org.web3j.protocol.Web3j web3)
```

Return the maximum gas limit that we can use in a transaction.

**Parameters:**

web3 - a web3j instance.

**Returns:**

a BigInteger of the gas limit, 0 in case of error.

---

## getBalance

```
static java.math.BigInteger getBalance(org.web3j.protocol.Web3j web3,  
    java.lang.String address)
```

## bytes32ToString

```
static java.lang.String bytes32ToString(byte[] bytes32)
```

Convert a bytes32 in a String.

**Parameters:**

bytes32 - the byte[].

**Returns:**

the String.

---

## stringToBytes32

```
static byte[] stringToBytes32(java.lang.String string)
```

Convert a String in a bytes32.

**Parameters:**

string - the String.

**Returns:**

the byte[].

---

---

**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.

---

## Fields

### cliOptions

```
private final java.util.ArrayList cliOptions
```

---

### cliFlags

```
private final java.util.ArrayList cliFlags
```

---

### maxLongOptLength

```
private int maxLongOptLength
```

---

## 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.

---

(continued from last page)

**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.

---

## addFlag

```
private void addFlag(java.lang.String shortOpt,  
    java.lang.String longOpt,  
    java.lang.String description)
```

---

## 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.

---

## 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).

(continued from last page)

**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.

---

**isNotAnOption**

```
private boolean isNotAnOption(java.lang.String string)
```

## com.aldodaquino.javautils Class CliHelper.CliObject

java.lang.Object

└─com.aldodaquino.javautils.CliHelper.CliObject

Direct Known Subclasses:

[CliFlag](#), [CliOption](#)

---

```
private class CliHelper.CliObject
extends java.lang.Object
```

---

### Fields

#### shortOpt

```
final java.lang.String shortOpt
```

---

#### longOpt

```
final java.lang.String longOpt
```

---

#### description

```
java.lang.String description
```

---

### Constructors

#### CliObject

```
CliObject(java.lang.String shortOpt,
          java.lang.String longOpt,
          java.lang.String description)
```

---

### Methods

#### isEqual

```
boolean isEqual(java.lang.String opt)
```

## com.aldodaquino.javautils Class CliHelper.CliOption

```
java.lang.Object
├── com.aldodaquino.javautils.CliHelper.CliObject
│   └── com.aldodaquino.javautils.CliHelper.CliOption
```

```
private class CliHelper.CliOption
extends CliHelper.CliObject
```

### Fields

#### values

```
final java.util.ArrayList values
```

### Constructors

#### CliOption

```
CliOption(java.lang.String shortOpt,
           java.lang.String longOpt,
           java.lang.String description)
```

### Methods

#### parse

```
boolean parse(java.lang.String opt,
              java.lang.String value)
```

#### getValues

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

## com.aldodaquino.javautils Class CliHelper.CliFlag

```
java.lang.Object
├─ com.aldodaquino.javautils.CliHelper.CliObject
│   └─ com.aldodaquino.javautils.CliHelper.CliFlag
```

---

```
private class CliHelper.CliFlag
extends CliHelper.CliObject
```

---

### Fields

#### **found**

```
boolean found
```

### Constructors

#### **CliFlag**

```
CliFlag(java.lang.String shortOpt,
        java.lang.String longOpt,
        java.lang.String description)
```

### Methods

#### **parse**

```
boolean parse(java.lang.String opt)
```

---

#### **isPresent**

```
boolean isPresent()
```



## 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.

---

## makeRequest

```
private static HttpHelper.Response makeRequest(java.lang.String url,  
                                                 java.lang.String method,  
                                                 java.lang.String parameters)
```

---

## 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**

## Constructors

### **Response**

private **Response**(int code,  
                  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 4  
ACCESS\_API\_PATH 6  
accessGrantedMap 78  
accountAddressLabel 45  
accountAddressPosition 46  
addFlag 100  
addOption 99  
address 89  
addresses 87  
addressField 67, 71  
appName 55  
AsyncPanel 20  
author 89  
AuthorContentTable 22  
AuthorInfoPanel 57  
AuthorPanel 59  
authors 87  
averageRating 90  
averageRatings 87

## B

becomesPremiumMap 78  
biggestPriceFairnessLabel 32  
BLOCK\_GAS\_LIMIT 96  
browseCallback 75  
buildWeb3 96  
buyContent 81  
buyPremium 65, 82  
bytes32ToString 97

## C

catalog 78  
catalogAddressField 24  
catalogAddressLabel 45  
catalogAddressPosition 45  
catalogClosedRunnables 79  
CatalogForm 24  
catalogForm 74  
CatalogManager 79

catalogManager 9, 13, 38, 59, 61, 64  
ChartWidget 26  
chartWidget 65  
chartWidgetPosition 65  
CliFlag 104  
cliFlags 99  
CliHelper 99  
CliObject 102  
CliOption 103  
cliOptions 99  
CobraHttpHelper 7  
code 110  
colNames 22, 47  
Colors 50  
ComponentFactory 27  
connect 24  
connectCallback 24, 74  
connectCatalog 14  
consumeContent 92  
consumer 8  
Content 90  
content 61, 91  
CONTENT\_FILE\_PATH 3  
ContentInfoPanel 62  
ContentList 30, 88  
ContentManager 91, 92  
contentManager 62  
contentMeaning 90  
contentMeaningRatings 88  
contentPublishedRunnables 91  
contract 93  
contractClass 93  
ContractManager 94  
credentials 9, 13, 93  
CustomerPanel 65

## D

data 110  
DEFAULT\_PORT 3  
deleteCallback 75  
deploy 4, 68, 94  
DEPLOY\_API\_PATH 6  
deployCallback 68, 74  
deployCatalog 14

deployContent 60  
DeployContentPanel 68  
description 102  
Dimensions 52  
disconnectCallback 75  
disconnectCatalog 14  
doAsync 20

## E

emptyStar 53  
enabled 41  
enjoy 90  
enjoyRatings 88

## F

feedbackAvailableBiConsumer 78  
file 68  
fileDialog 18  
FileExchange 105  
filledStar 53  
found 104

## G

gasLimit 93  
gasPrice 93  
genre 89  
genreField 67  
GenreInfoPanel 69  
genres 87  
getAddress 94  
getAuthorContents 83  
getBalance 97  
getCatalogManager 15  
getContentInfo 83  
getContentList 83  
getContentListWithViews 83  
getFilteredContentList 88  
getGasLimit 97  
getGasPrice 96  
getHelpMessage 101  
getHostname 92  
getLatest 83

getLatestByAuthor 84  
getLatestByGenre 84  
getMinimumSize 74  
getMissingOptionMessage 101  
getMostPopular 84  
getMostPopularByAuthor 84  
getMostPopularByGenre 84  
getMostRated 85  
getMostRatedByAuthor 85  
getMostRatedByGenre 85  
getNewContentList 39, 83  
getOwner 95  
getPort 92  
getPrivateKey 15  
getRating 42  
getRole 15  
getScaled 53  
getStar 42  
getTable 65  
getTheName 79  
getUserAddress 14  
getValue 100  
getValues 100, 103  
gift 62, 71  
giftCallback 71  
giftContent 81  
giftPremium 66, 82  
GREEN 50

## H

handle 8  
hasAccess 81  
highestContentMeaningLabel 32  
highestRatedLabel 31  
hostname 9  
HttpHelper 107  
HttpRequestHandler 8

## I

Images 53  
INFO\_PANEL\_PADDING 51  
InfoPanel 32  
isCatalogOwner 15



isCellEditable 23, 48  
isEqual 102  
isNotAnOption 101  
isPremium 82  
isPresent 100, 104

## J

jsonifyParameters 109

## L

LabelPanel 33  
LATERAL\_BAR\_PADDING 51  
lateralBar 65  
latestLabel 31  
listenAccessGranted 80  
listenBecomesPremium 80  
listenCatalogClosed 81  
listenContentPublished 92  
listenFeedbackAvailable 80  
listenForEvents 79  
listenNewContentAvailable 79  
listenPaymentAvailable 80  
load 94  
loader 33  
loading 53  
login 14, 35  
loginCallback 35, 74  
LoginForm 35  
loginForm 73  
Logo 37  
logo 53  
LOGO\_SIZE 51  
longOpt 102

## M

Main 3, 12  
main 3, 12  
makeGet 108  
makePost 109  
makeRequest 109  
manageCallback 75  
maxLongOptLength 99

mostEnjoyedLabel 31  
mostPopularLabel 31

## N

name 89  
nameField 67  
names 87  
newAccountAddressLabel 46  
newBorder 28  
newButton 27  
newCatalogAddressLabel 46  
newConfirmDialog 17  
newContentAvailableBiConsumers 78  
NewContentsWidget 38  
newDialog 17  
newErrorDialog 16  
newExitDialog 17  
newField 28  
newGBC 43  
newHandler 7  
newMessageDialog 16  
newPasswordField 27  
newPremiumLabel 46  
newSpinner 28  
newTextField 27  
newTitledBorderPanel 28  
newVSpacer 28  
newWindow 16, 57, 62, 69  
numberSpinner 38

## O

onDeployed 60  
openFileDialog 17  
openFileSocket 105  
owner 94

## P

parse 100, 103, 104  
parseGET 107  
parseJson 108  
parsePOST 107  
parseQuery 107

paymentAvailableMap 78  
PickUserPanel 71  
port 9  
portField 67  
postConnect 74  
premiumLabel 46  
premiumPosition 46  
prepareLink 62  
prepareRows 22, 30, 47  
prepareStar 63  
price 89  
priceFairness 90  
priceFairnessRatings 88  
priceField 67  
prices 87  
privateKey 9, 13  
privateKeyInput 35

## Q

querifyParameters 109

## R

rating 41  
readFile 106  
receiveFile 105  
RED 50  
replaceComponent 43  
replacingPosition 33, 74  
Response 110  
retrieveContent 62  
role 13  
ROLE\_AUTHOR 13  
ROLE\_CUSTOMER 13  
RoleForm 40  
roleForm 74

## S

saveFileDialog 17  
sendResponse 108  
setContent 12  
setFontSize 17  
setRating 42

setRole 15  
shortOpt 102  
showMainPanel 12  
showViews 65  
STAR\_SIZE 51  
StarPanel 41, 42  
stars 41  
STARS\_NUMBER 41  
STARTER\_PANEL\_PADDING 51  
StarterPanel 74  
startFileSender 105  
startLoading 21  
Status 9, 14  
status 8, 33, 38, 45, 59, 61, 64, 68, 73  
stopLoading 21  
Strings 55  
stringToBytes32 97  
suicide 95

## T

table 59, 64  
tableContainer 59, 64  
toString 110

## U

update 34, 65  
updateStatus 46  
updateTable 60  
UpgradablePanel 43  
UserInfo 46  
userInfo 65, 73  
Utils 16, 96

## V

V\_SPACER\_L 52  
V\_SPACER\_M 52  
V\_SPACER\_S 51  
values 103  
view 62  
views 87, 89  
ViewsContentTable 47  
vote 82

VotingPanel 76

## W

web3 93

whenDone 73

window 12, 20

WINDOW\_TITLE 57, 61, 69

withdraw 85

withdrawSelected 60

writeFile 106