

# Lanterna

Lanterna est une librairie permettant de créer des interfaces graphiques en mode texte.

Le but de ce tp est de voir de la POO à travers cette librairie implémentant beaucoup de pattern.

Lanterna est une bibliothèque Java vous permettant d'écrire des interfaces utilisateur semi-graphiques simples dans un environnement texte uniquement, très similaire aux cursus de la bibliothèque C mais avec plus de fonctionnalités. Lanterna prend en charge les terminaux et les émulateurs de terminaux compatibles xterm tels que konsole, gnome-terminal, mastic, xterm et bien d'autres. Un des principaux avantages de Lanterna est qu'elle ne dépend d'aucune bibliothèque native, mais fonctionne à 100% en Java pur.

De plus, lors de l'exécution de Lanterna sur des ordinateurs dotés d'un environnement graphique (tel que Windows ou Xorg), un émulateur de terminal fourni écrit en Swing sera utilisé plutôt que la sortie standard. De cette façon, vous pouvez développer comme d'habitude à partir de votre IDE (la plupart d'entre eux ne prennent pas en charge les caractères de contrôle ANSI dans leur fenêtre de sortie), puis se déployer sur votre serveur sans interface utilisateur sans changer de code.

## Installation

Dans le pom.xml rajouter :

```
<dependency>
  <groupId>com.googlecode.lanterna</groupId>
  <artifactId>lanterna</artifactId>
  <version>3.0.1</version>
</dependency>
```

## Un terminal

Le noyau de la couche Terminal est l'interface de Terminal que vous devez récupérer. Cela peut être fait soit en instanciant directement l'une des classes concrètes implémentées (UnixTerminal, CygwinTerminal ou SwingTerminal) ou en utilisant DefaultTerminalFactory pour créer un terminal:

```
Terminal terminal = new DefaultTerminalFactory().createTerminal();
```

## Un écran

L'écran est une surface sur laquelle on va dessiner:

```
Terminal terminal = new DefaultTerminalFactory().createTerminal();
Screen screen = new TerminalScreen(terminal);
screen.startScreen();
```

Puis nous allons rajouter un panel:

```
Panel panel = new Panel();
    panel.setLayoutManager(new GridLayout(2));

    final Label lblOutput = new Label("");

    panel.addComponent(new Label("Num 1"));
    final TextBox txtNum1 = new TextBox().setValidationPattern(Pattern.compile("[0-9]*")).addTo(panel);

    panel.addComponent(new Label("Num 2"));
    final TextBox txtNum2 = new TextBox().setValidationPattern(Pattern.compile("[0-9]*")).addTo(panel);

    panel.addComponent(new EmptySpace(new TerminalSize(0, 0)));
    new Button("Add!", new Runnable() {
        @Override
        public void run() {
            int num1 = Integer.parseInt(txtNum1.getText());
            int num2 = Integer.parseInt(txtNum2.getText());
            lblOutput.setText(Integer.toString(num1 + num2));
        }
    }).addTo(panel);

    panel.addComponent(new EmptySpace(new TerminalSize(0, 0)));
    panel.addComponent(lblOutput);
```

Et enfin on lit le panel avec l'ecran via un fenetre

```
BasicWindow window = new BasicWindow();
    window.setComponent(panel);

    // Create gui and start gui
    MultiWindowTextGUI gui = new MultiWindowTextGUI(screen, new DefaultWindowManager(),
new EmptySpace(TextColor.ANSI.BLUE));
```

```
gui.addWindowAndWait( window );
```

Un code plus interessant est de faire rajouter une liste d'objet à une table

```
package be.ethnic.lanterna;

import java.awt.Menu;
import java.awt.MenuBar;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import com.googlecode.lanterna.TextColor;
import com.googlecode.lanterna.gui2.BasicWindow;
import com.googlecode.lanterna.gui2.Borders;
import com.googlecode.lanterna.gui2.Button;
import com.googlecode.lanterna.gui2.DefaultWindowManager;
import com.googlecode.lanterna.gui2.Direction;
import com.googlecode.lanterna.gui2.EmptySpace;
import com.googlecode.lanterna.gui2.GridLayout;
import com.googlecode.lanterna.gui2.Label;
import com.googlecode.lanterna.gui2.LinearLayout;
import com.googlecode.lanterna.gui2.MultiWindowTextGUI;
import com.googlecode.lanterna.gui2.Panel;
import com.googlecode.lanterna.gui2.TextBox;
import com.googlecode.lanterna.gui2.dialogs.FileDialogBuilder;
import com.googlecode.lanterna.gui2.table.Table;
import com.googlecode.lanterna.screen.Screen;
import com.googlecode.lanterna.screen.TerminalScreen;
import com.googlecode.lanterna.terminal.DefaultTerminalFactory;
import com.googlecode.lanterna.terminal.Terminal;

public class MyLanterna extends BasicWindow{

    Table<String> table ;
    private void clearAllRow()
    {
        while ( table.getTableModel().getRowCount()>0)
```

```

    }
    }
    table.getTableModel().removeRow(0);
    }
}

public MyLanterna()
{
    super();
    Panel mainPanel = new Panel();
    mainPanel.setLayoutManager(new BorderLayout(Direction.VERTICAL));
    table = new Table<String>("Nom");
    mainPanel.addComponent(table);
    mainPanel.addComponent(getFormPanel().withBorder(Borders.singleLine("Ajouter un
utilisateur")));

    this.setComponent(mainPanel);
}

}

public Panel getFormPanel() {
    Panel panel = new Panel();
        panel.setLayoutManager(new GridLayout(2));

        final Label lblOutput = new Label("");

        panel.addComponent(new Label("Nom"));
        final TextBox txtNum1 = new TextBox().addTo(panel);
        new Button("Add!", new Runnable() {
            public void run() {
                List<List<String>> value = table.getTableModel().getRows();
                }
                clearAllRow();
                List<String> newV=new ArrayList<String>();
                newV.add(txtNum1.getText());
                }
                value.add(newV);
                for (List<String> str: value)
                {
                    table.getTableModel().addRow(str);
                }
            }
        }
}

```

```
        }
    }).addTo(panel);
    return panel;
}

public static void main(String [] args) throws IOException
{
    Terminal terminal = new DefaultTerminalFactory().createTerminal();
    Screen screen = new TerminalScreen(terminal);
    screen.startScreen();
    MultiWindowTextGUI gui = new MultiWindowTextGUI(screen, new DefaultWindowManager(),
new EmptySpace( TextColor.ANSI.BLUE));
    gui.addWindowAndWait(new MyLanterna());
}
}
```

---

Revision #5

Created 3 November 2019 20:07:38 by Admin

Updated 18 November 2019 19:40:33 by Admin