

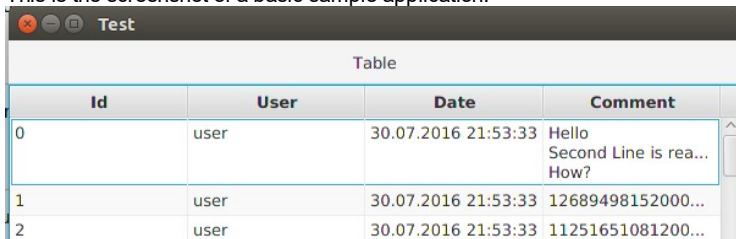
jLuger.de - JavaFX: Text rendering in TableView

[Home](#) [Blog](#) [Articles](#) [Legal Info](#) [About Me](#)

This post is about text rendering in JavaFX's `TableView`. The `TableView` uses `TextCells` that descend (indirectly) from `Labeled`. You may expect that this way it should support text rendering but unfortunately this is only true when you have basic requirements. As soon as you want fancy things, like displaying long lines of text completely, you are out of basic support.

Please note that all samples were created and tested with `jdk1.8.0_102`. I hope that later versions won't have this bugs and thus this post may become irrelevant.

This is the screenshot of a basic sample application:



Id	User	Date	Comment
0	user	30.07.2016 21:53:33	Hello Second Line is rea... How?
1	user	30.07.2016 21:53:33	12689498152000...
2	user	30.07.2016 21:53:33	11251651081200...

The view:

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.web.*?>
<?import java.lang.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.layout.AnchorPane?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity"
  minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0"
  prefWidth="600.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1">
  <center>
  </center>
  <top>
  </top>
  <bottom>
  <center>
    <TableView fx:id="tableView" prefHeight="200.0" prefWidth="200.0" BorderPane.alignment="CENTER">
      <columns>
        <TableColumn fx:id="idTableColumn" prefWidth="75.0" text="Id" />
        <TableColumn fx:id="userTableColumn" prefWidth="75.0" text="User" />
        <TableColumn fx:id="dateTableColumn" prefWidth="75.0" text="Date" />
        <TableColumn fx:id="commentTableColumn" prefWidth="75.0" text="Comment" />
      </columns>
      <columnResizePolicy>
        <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
      </columnResizePolicy>
    </TableView>
  </center>
</BorderPane>
```

The controller:

```
public class TableTestOneController {
    @FXML
    private TableView<Content> tableView;
    @FXML
    private TableColumn<Content, String> idTableColumn;
    @FXML
    private TableColumn<Content, String> userTableColumn;
    @FXML
    private TableColumn<Content, String> dateTableColumn;
    @FXML
    private TableColumn<Content, String> commentTableColumn;

    @FXML
    public void initialize() {
        ObservableList<Content> contentList = FXCollections.observableArrayList();
        for (int i = 0; i < 100; i++) {
            contentList.add(Content.createRandomInstance());
        }
        contentList.get(0).setComment("Hello\nSecond Line is really really really long\nHow?");
        tableView.setItems(contentList);
        idTableColumn.setCellValueFactory(
            (CellDataFeatures<Content, String> p) -> new ReadOnlyStringWrapper(p.getValue().getId()));
    }
}
```

```

        userTableColumn.setCellValueFactory(
            (CellDataFeatures<Content, String> p) -> new ReadOnlyStringWrapper(p.getValue().getUser());
        );
        dateTableColumn.setCellValueFactory(
            (CellDataFeatures<Content, String> p) -> new ReadOnlyStringWrapper(p.getValue().getDate());
        );
        commentTableColumn.setCellValueFactory(
            (CellDataFeatures<Content, String> p) -> new ReadOnlyStringWrapper(p.getValue().getComment());
        );
    }
}

```

The main class:

```

public class StartTableTestOne extends Application {
    public static void main(String[] args) {
        launch(args);
    }

    @Override
    public void start(Stage primaryStage) throws Exception {
        TableTestOneController controller = new TableTestOneController();
        FXMLLoader loader = new FXMLLoader(getClass().getResource("TableTestOne.fxml"));
        loader.setController(controller);
        Parent screenParent;
        try {
            screenParent = loader.load();
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
        Scene scene = new Scene(screenParent);
        primaryStage.setTitle("Test");
        primaryStage.setScene(scene);
        primaryStage.show();
    }
}

```

As you can see the long text is cutted of with three dots. To fix this and get a line wrap instead of a cutting I've done some research in the javadoc and found out the following:

- Each column has a CellFactory that does the rendering.
- The default CellFactory creates an instance of a subclass of a TableCell.
- TableCell inherits from Labeled with has the method setWrapText.

The result of this was my own CellFactory:

```

public static final Callback<TableColumn<Content,String>, TableCell<Content,String>> WRAPPING_CELL_FACTORY =
    new Callback<TableColumn<Content,String>, TableCell<Content,String>>() {

        @Override public TableCell<Content,String> call(TableColumn<Content,String> param) {
            TableCell<Content,String> tableCell = new TableCell<Content,String>() {
                @Override protected void updateItem(String item, boolean empty) {
                    if (item == getItem()) return;

                    super.updateItem(item, empty);

                    if (item == null) {
                        super.setText(null);
                        super.setGraphic(null);
                    } else {
                        super.setText(item);
                        super.setGraphic(null);
                    }
                }
            };
            tableCell.setWrapText(true);
            return tableCell;
        }
    };
}

```

I've setted this in the initialize method in my controller:

```

commentTableColumn.setCellFactory(WRAPPING_CELL_FACTORY);

```

The result was this:

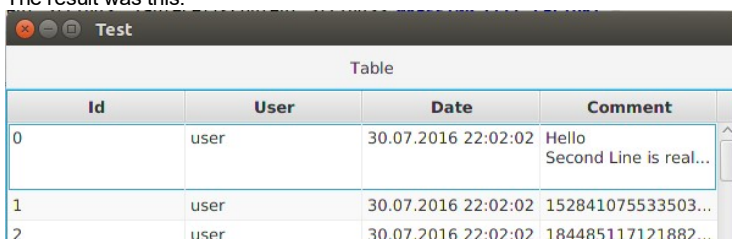


Table			
Id	User	Date	Comment
0	user	30.07.2016 22:02:02	Hello Second Line is real...
1	user	30.07.2016 22:02:02	152841075533503...
2	user	30.07.2016 22:02:02	184485117121882...

Note that the text still cutted of but now the last line disappears too! Not quite the result I've wanted. So I've looked around and found out that the setGraphic method accepts not only graphics but any Node (the base class for a lot of elements in JavaFX).

The next step was to create a Label in the updateItem method of the CellFactory, set setWrapText(true) on it and

give it to setGraphic. And nothing changed. I've got the same output as above.

After some more experiments I've put the Label in a VBox. The code of the CellFactory was no this:

```
public static final Callback<TableColumn<Content,String>, TableCell<Content,String>> WRAPPING_CELL_FACTORY =
    new Callback<TableColumn<Content,String>, TableCell<Content,String>>() {

        @Override public TableCell<Content,String> call(TableColumn<Content,String> param) {
            TableCell<Content,String> tableCell = new TableCell<Content,String>() {
                @Override protected void updateItem(String item, boolean empty) {
                    if (item == getItem()) return;

                    super.updateItem(item, empty);

                    if (item == null) {
                        super.setText(null);
                        super.setGraphic(null);
                    } else {
                        super.setText(null);
                        Label l = new Label(item);
                        l.setWrapText(true);
                        VBox box = new VBox(l);
                        super.setGraphic(box);
                    }
                }
            };
            return tableCell;
        }
    };
```

It worked only half way:

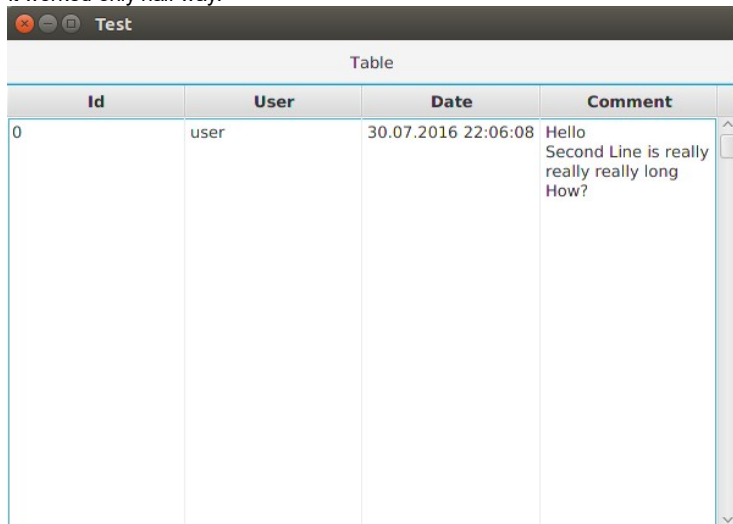


Table			
Id	User	Date	Comment
0	user	30.07.2016 22:06:08	Hello Second Line is really really really long How?

All text is visible but there is a huge gap to the next cell. A little fun fact for it: If I leave the `l.setWrapText(true)` out it won't have the gab (but also cut of the text). To get around this I've tried to add a height change listener to the Label which should adjust the height of the VBox. The code is this:

```
l.heightProperty().addListener((observable,oldValue,newValue)-> {
    box.setPrefHeight(newValue.doubleValue()+7);
});
```

The result was this:

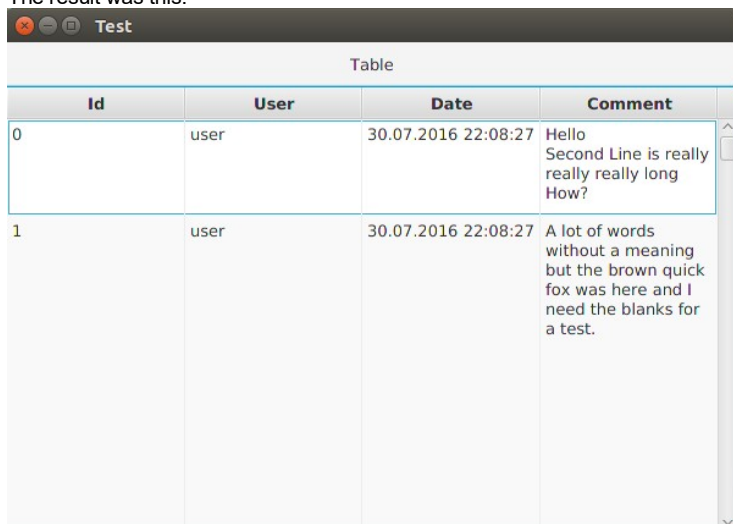


Table			
Id	User	Date	Comment
0	user	30.07.2016 22:08:27	Hello Second Line is really really really long How?
1	user	30.07.2016 22:08:27	A lot of words without a meaning but the brown quick fox was here and I need the blanks for a test.

The first row was correctly rendered but not the second one. The size of the second row got correct as soon as I've started to scroll or even when I've put the application in the background and then in the foreground. So I've added a `this.getTableRow().requestLayout();` to the listener which did nothing. That was because my listener was called in the layout phase and in that the call is ignored. To get around this I've wrapped it in `Platform.runLater` although I was already running in the `JavaFX-Thread`.

The complete `CellFactory` code:

```
public static final Callback<TableColumn<Content,String>, TableCell<Content,String>> WRAPPING_CELL_FACTORY =
    new Callback<TableColumn<Content,String>, TableCell<Content,String>>() {

        @Override public TableCell<Content,String> call(TableColumn<Content,String> param) {
            TableCell<Content,String> tableCell = new TableCell<Content,String>() {
                @Override protected void updateItem(String item, boolean empty) {
                    if (item == getItem()) return;

                    super.updateItem(item, empty);

                    if (item == null) {
                        super.setText(null);
                        super.setGraphic(null);
                    } else {
                        super.setText(null);
                        Label l = new Label(item);
                        l.setWrapText(true);
                        VBox box = new VBox(l);
                        l.heightProperty().addListener((observable,oldValue,newValue)-> {
                            box.setPrefHeight(newValue.doubleValue()+7);
                            Platform.runLater(()->this.getTableRow().requestLayout());
                        });
                        super.setGraphic(box);
                    }
                }
            };
            return tableCell;
        }
    };
```

The result:

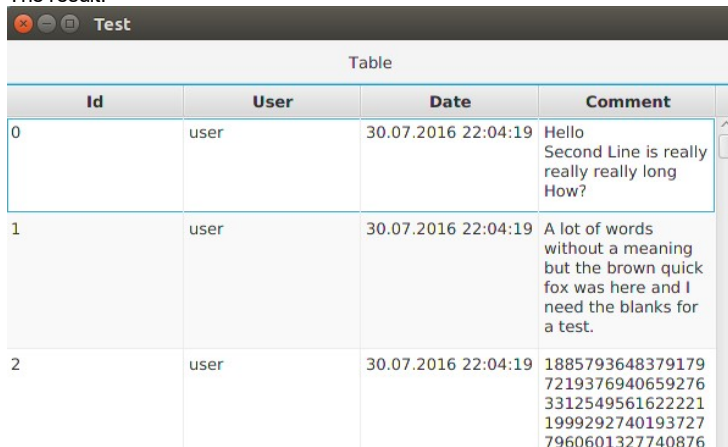


Table			
Id	User	Date	Comment
0	user	30.07.2016 22:04:19	Hello Second Line is really really really long How?
1	user	30.07.2016 22:04:19	A lot of words without a meaning but the brown quick fox was here and I need the blanks for a test.
2	user	30.07.2016 22:04:19	1885793648379179 7219376940659276 3312549561622221 1999292740193727 7960601327740876