Eclipse Scout

Release Notes

Scout Team

Version 11.0

Table of Contents

About This Release . . . 1
- Service Releases . . . 1
- Obtaining the Latest Version . . . 1
- Demo Applications . . . 2

Mobile Improvements . . . 3
- Default Detail Form and Tile Overview . . . 3
- More Information in Outline . . . 4
- New Style for Views . . . 5
- Fullscreen Dialogs . . . 6
- Wizard: Improved Progress Field . . . 6
- New MobilePopup . . . 6

New ESLint Settings . . . 7

Custom Webpack Config . . . 8

New Widgets . . . 9
- Breadcrumb Bar Field . . . 9
- Chart . . . 9

Widget Enhancements . . . 13
- Table: New Property 'compact' . . . 13
- Menu: New Property 'textPosition' . . . 13
- Form . . . 14
- Widget: ExecFocusIn, ExecFocusOut . . . 14
- TabBox: Support for LabelVisible . . . 14

About This Release

The latest version of this release is: 11.0-SNAPSHOT.

You can see the detailed change log on GitHub.

Coming from an older Scout version? Check out the Migration Guide!

Service Releases

Scout 11.0 will continue to be maintained for a while and new builds may be released from time to time. Beside bug fixes, these releases may even contain some minor features.

Obtaining the Latest Version

Scout Runtime for Java

Scout Runtime artifacts for Java are distributed using Maven Central:

- 11.0-SNAPSHOT on *Maven Central*
- 11.0-SNAPSHOT on *mvnrepository.com*

Usage example in the parent POM of your Scout application:

```
<dependency>
    <groupId>org.eclipse.scout.rt</groupId>
    <artifactId>org.eclipse.scout.rt</artifactId>
    <version>11.0-SNAPSHOT</version>
    <type>pom</type>
    <scope>import</scope>
</dependency>
```

Scout Runtime for JavaScript

Scout Runtime artifacts for JavaScript are distributed using npm:

- Scout Core Runtime
- All official Scout JavaScript packages

IDE Tooling (Scout SDK)

Starting with Scout 11 the IDE Tooling requires at least Java 11 to run. With the help of the SDK plugins you can still develop applications running with Java 8, but the IDE itself requires Java 11 or newer.

Scout officially supports IntelliJ IDEA and Eclipse for Scout Developers.

Eclipse

You can download the complete Eclipse IDE with Scout SDK included (EPP) here:
Eclipse for Scout Developers

To install the Scout SDK into your existing Eclipse IDE, use this update site:
http://download.eclipse.org/scout/releases/11.0/

IntelliJ IDEA

You can download the Scout plugin for IntelliJ IDEA from the JetBrains Plugin Repository or you can use the plugins client built into IntelliJ IDEA. Please refer to the IntelliJ Help on how to install and manage plugins.

Demo Applications

The demo applications for this version can be found on the features/version/11.0-SNAPSHOT branch of our docs repository on GitHub.

If you just want to play around with them without looking at the source code, you can always use the deployed versions:

- https://scout.bsi-software.com/contacts/
- https://scout.bsi-software.com/widgets/
- https://scout.bsi-software.com/jswidgets/

Mobile Improvements

Default Detail Form and Tile Overview

When no page is selected, the desktop client shows the default detail form, or if not available, the tile outline overview. This feature was disabled on mobile, until now. So if you startup your mobile Scout Classic app you will probably notice the new look. If you don't like it you can always disable it by not creating a default detail form (`Outline#createDefaultDetailForm`) or by setting `Outline#getConfiguredOutlineOverviewVisible` to false.

[outline overview.png] | */home/jenkins/agent/workspace/scoutdoc-docs-*

11.0/docs/build/scout_releasenotes/../../adoc/releasenotes/imgs/outline_overview.png.png

More Information in Outline

Until now, the outline in mobile mode displayed the same summary cell as in desktop mode. Even though you could already create a custom summary cell for mobile, doing so is time-consuming. This is why we decided to implement a different strategy for mobile to compute a better summary cell. It is enabled by default so that every table page will benefit right away.

[mobile summary cell.png] | */home/jenkins/agent/workspace/scoutdoc-docs-*

11.0/docs/build/scout_releasenotes/../../adoc/releasenotes/imgs/mobile_summary_cell.png.png

Since this new strategy is based on the new feature introduced for tables (Table: New Property 'compact'), it can be easily customized in the same way. If you don't like it you can return to the previous strategy by overriding `AbstractTable#createSummaryCellBuilder`, but why should you ;-) ?

New Style for Views

The device transformer now adjusts the style of a view so that it matches the style of a dialog. Also, the view tabs won't be visible anymore. This should give a more consistent look and should make it easier to use.

[mobile view.png.png] | */home/jenkins/agent/workspace/scoutdoc-docs-*

11.0/docs/build/scout_releasenotes/../../adoc/releasenotes/imgs/mobile_view.png.png.png

If you don't like that behavior, you can disable it in general or for specific forms:

```
BEANS.get(IDeviceTransformationService.class).excludeFormTransformation(this,
MobileDeviceTransformation.USE_DIALOG_STYLE_FOR_VIEW);
```

Fullscreen Dialogs

The device transformer now sets the property `maximized` of dialogs to true so that every dialog will be displayed in full screen. See also Support Maximized.

If you don't like that behavior, you can disable it in general or for specific forms:

```
BEANS.get(IDeviceTransformationService.class).excludeFormTransformation(this,
MobileDeviceTransformation.MAXIMIZE_DIALOG);
```

Wizard: Improved Progress Field

The progress visualization and step activation has been improved for smaller screens.

New MobilePopup

MobilePopup was only available on Scout JS but is now available for Scout Classic as well. A mobile popup slides in from the bottom. This is it, not more. It has already been used for search or tool forms on mobile.

New ESLint Settings

The eslint rules used by Scout have been adjusted to encourage the use of some great ES6 features. The following rules have been turned on: `no-var`, `prefer-arrow-callback`,`prefer-rest-params`, `prefer-spread`.

If you have a dependency to `@eclipse-scout/eslint-config` and you update to the latest version, your project will use the new settings, too. This also means your JavaScript code may report some new warnings. Have a look at the Migration Guide to learn how to handle them (e.g. migrating code or disabling the rules).

Custom Webpack Config

If you have a lot of themes, working on a less file can be frustrating since every theme will be built. To save time, it is now possible to control which themes should be built. To do so, create a `webpack.config.custom.js file` (which can be added to .gitignore because not every developer has the same preferences). Add the following content:

```
const baseConfig = require('./webpack.config');
module.exports = (env, args) => {
  args.themes = args.themes || ['default'];
  return baseConfig(env, args);
};
```

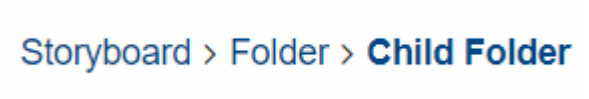
To make this work your base webpack config needs to register the themes using the function baseConfig.addThemes:

```
baseConfig.addThemes(config.entry, {
  themes: args.themes,
  availableThemes: ['default', 'dark'],
  generator: name => [`yourapp-theme${name}`, `./src/main/js/yourapp-
theme${name}.less`]
});
```

New Widgets

Breadcrumb Bar Field

The new breadcrumb bar widget allows to display a hierarchical structure in a horizontal style. Each breadcrumb item is clickable. The breadcrumb can automatically shrink if there is not enough place available.



Chart

The new chart widget allows visualizing data in several ways. It can be used inside a chart field, a chart tile or as a plain chart. The supported chart types include bar (vertical and horizontal), line, bar and line combined, pie, doughnut, bubble, polar area, radar, fulfillment, salesfunnel, speedo and venn. All charts support different color schemes and themes.

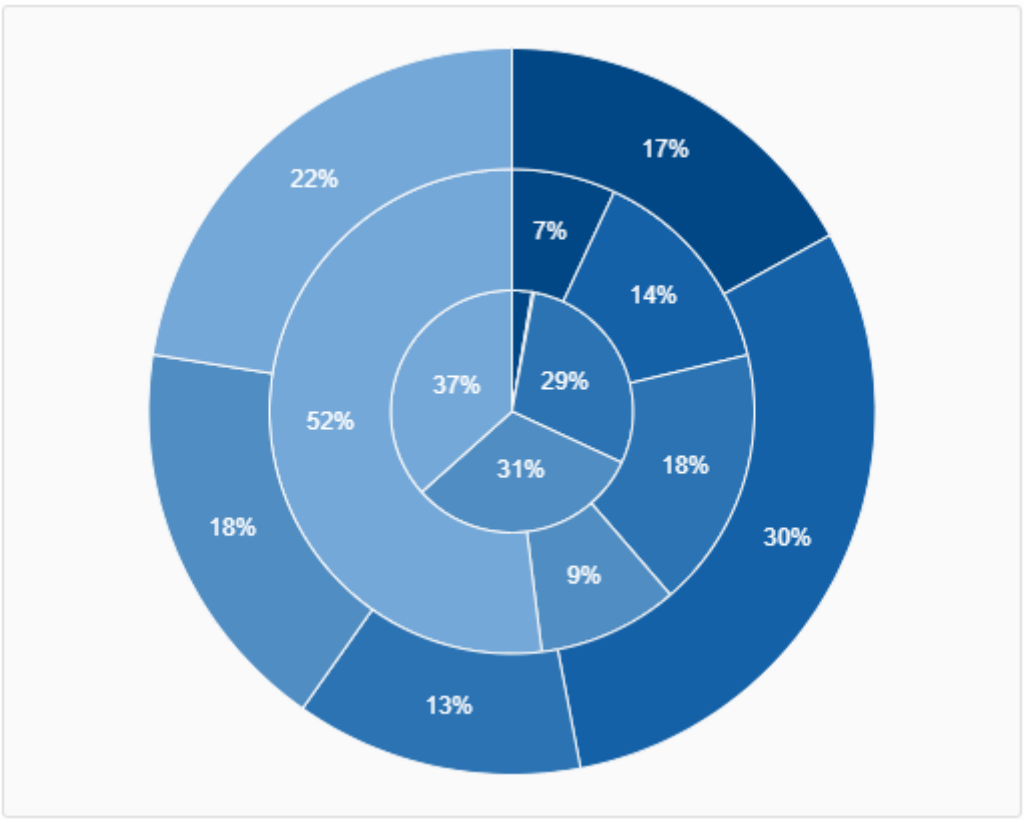


Figure 1. Pie chart in a tile using the default color scheme.

Figure 2. Combined bar line chart in a tile using the alternative inverted color scheme.

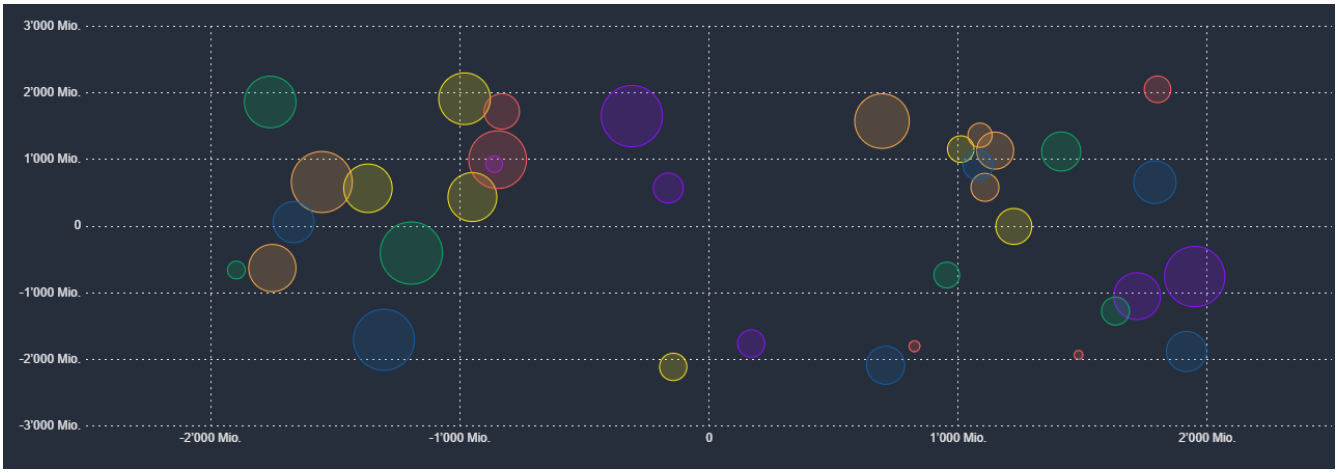


Figure 3. Bubble chart using the rainbow color scheme and the dark theme.

There are numerous styling options, e.g. tension and fill for line and radar charts, stacked bar charts or a transparent and a checkable mode.

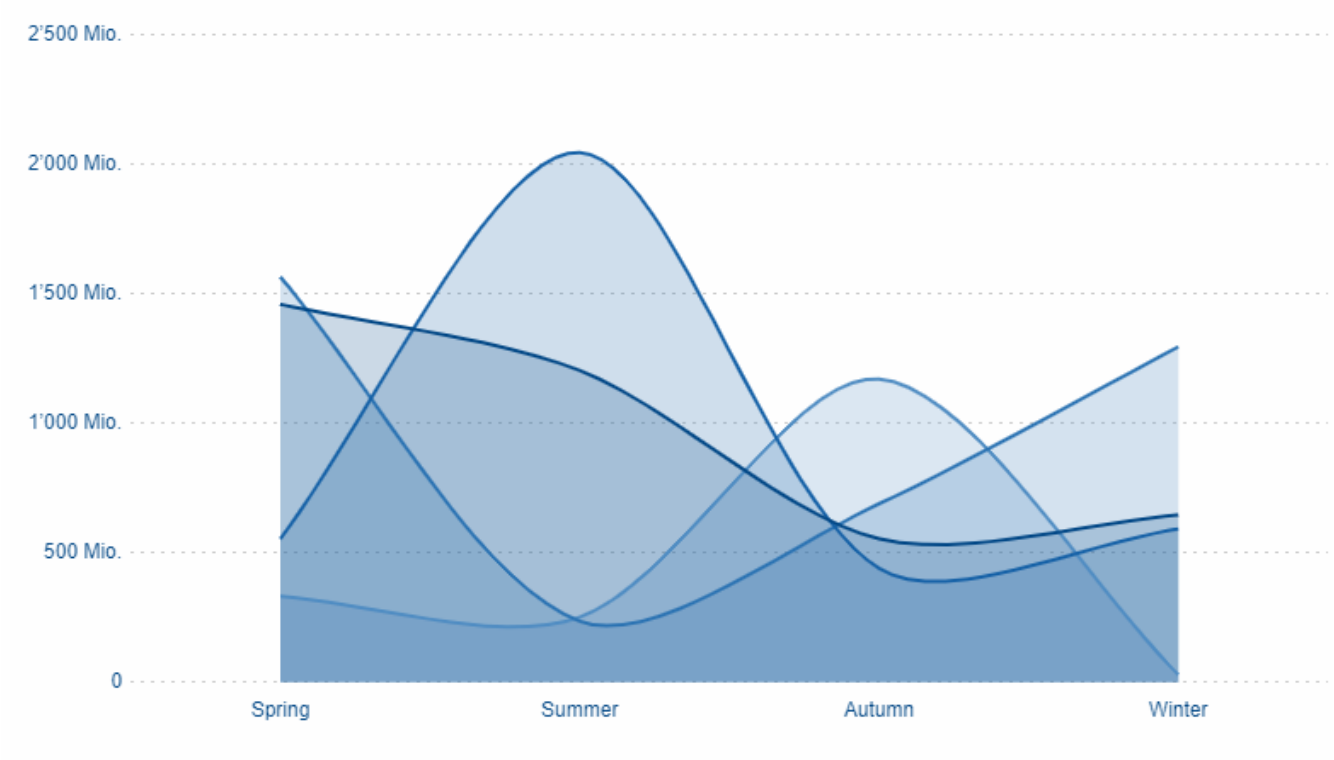


Figure 4. Tension and fill for line charts.

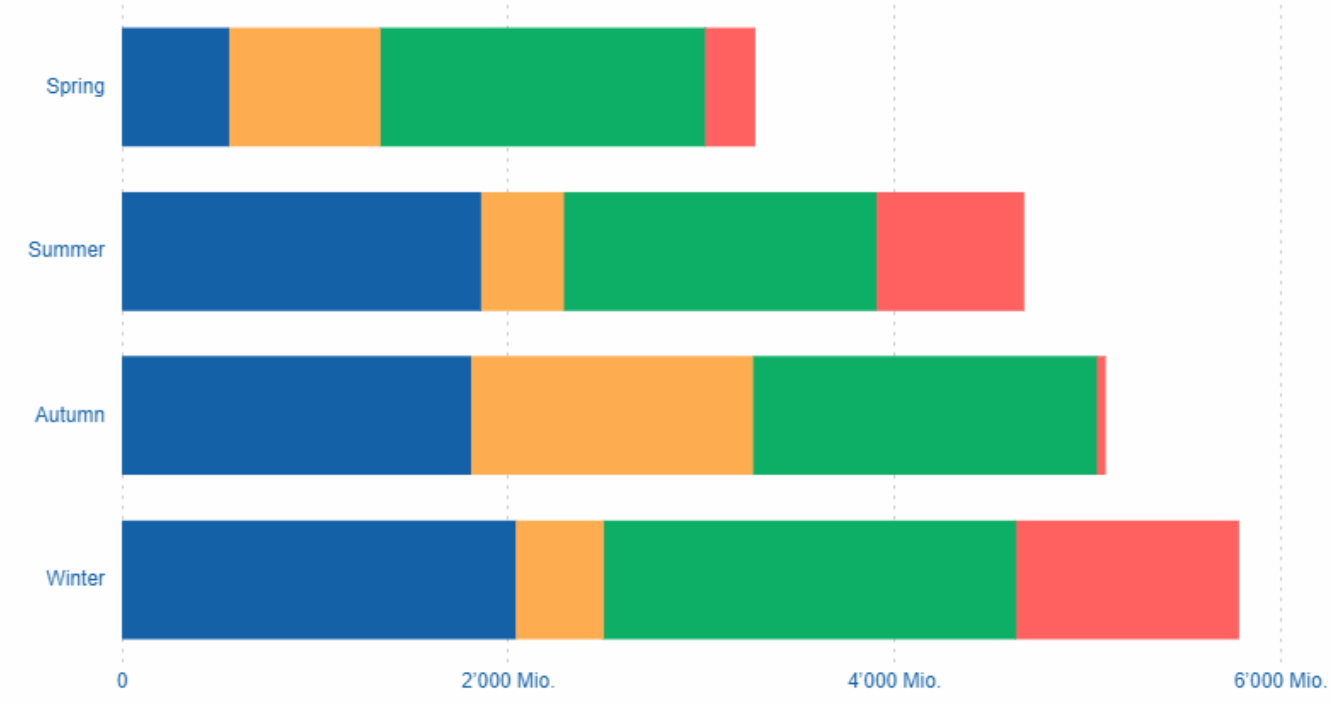


Figure 5. Stacked horizontal bar chart.

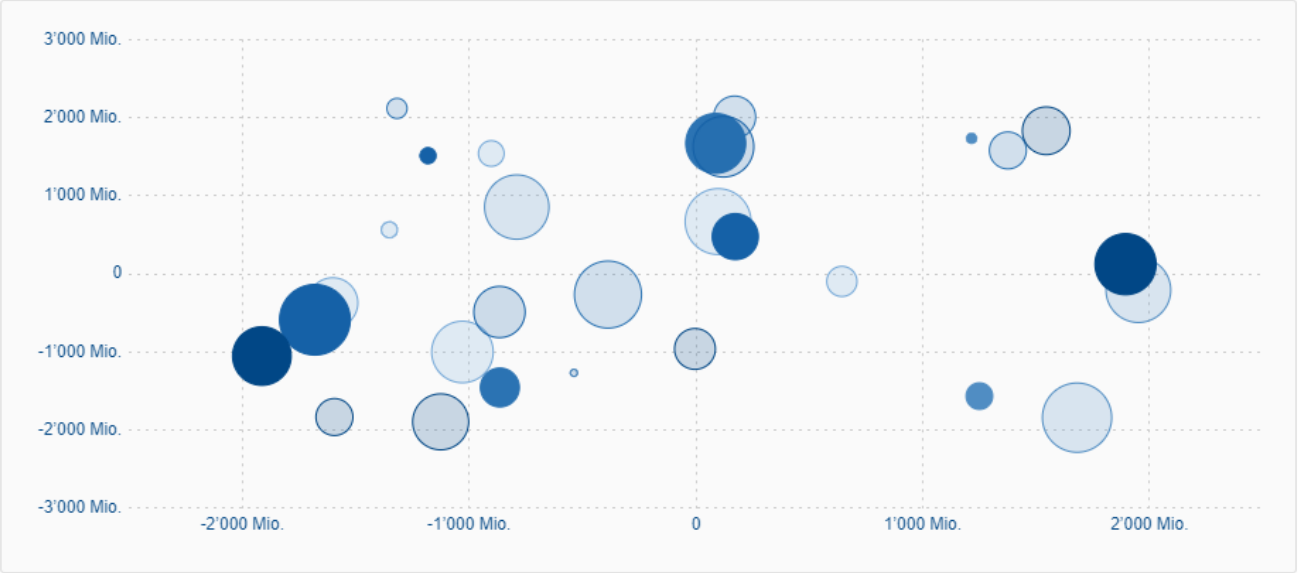


Figure 6. Checkable bubble chart.

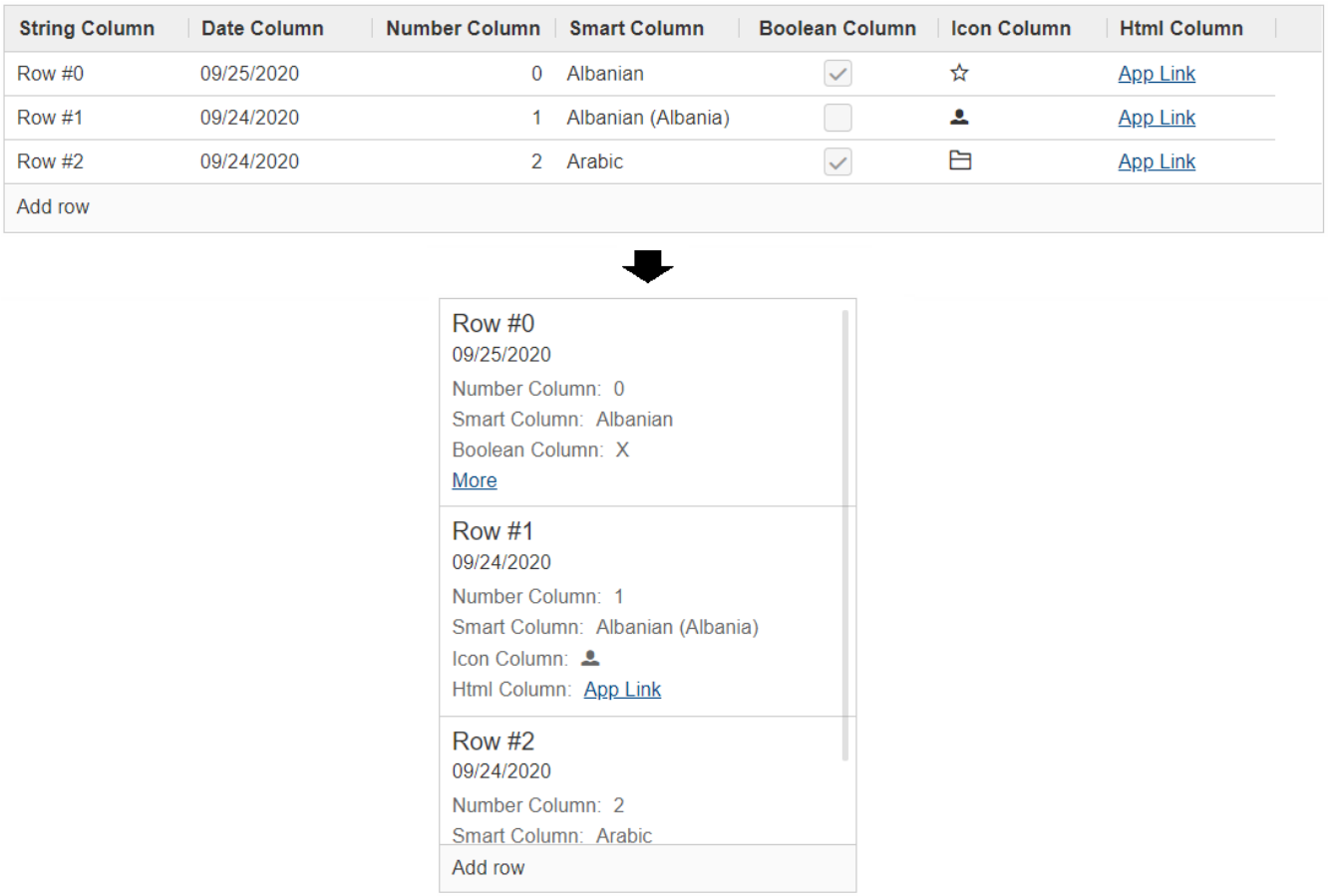
For more information about the chart widget see Technical Guide for Scout JS.

Widget Enhancements

Table: New Property 'compact'

It is now possible to set a table into compact mode. This will hide all columns and insert a new one containing the content of every other column. The content will be displayed vertically which means the cells of a row will be put below each other. This new style is perfect if the width of the table is limited (e.g. on a mobile phone).

The table uses a so called `ITableCompactHandler` which does the conversion. If you don't like how the content is arranged you can easily control the conversion by adjusting the compact handler or by writing your own one.



Menu: New Property 'textPosition'

With this new property, the text can be positioned on the bottom of the icon

[menu text bottom.png] | */home/jenkins/agent/workspace/scoutdoc-docs-*

11.0/docs/build/scout_releasenotes/../../adoc/releasenotes/imgs/menu_text_bottom.png.png

Form

Support Maximized

The property `maximized` is an old property but had no effect with the current UI. With Scout 11 the property will finally be interpreted. Setting it to true on a dialog will make it use the whole screen.

New Property 'headerVisible'

With this new property you can manually control whether the header should be visible. Until now it depended on the display hint property. This could be useful when embedding the form in a popup or another widget.

Widget: ExecFocusIn, ExecFocusOut

Every widget now supports focus event. You can either use a `WidgetListener` or implement `execFocusIn` and `execFocusOut`. The currently focused element is available using `IDesktop#getFocusedElement()`.

Important: the focus tracking needs to be explicitly activated before it can be used. Just call `IDesktop#setTrackFocus(true)` to do so. And remember to deactivate it again once you don't need it anymore to not produce unnecessary requests.

TabBox: Support for LabelVisible

Setting the property `labelVisible` to false will now hide the tab area. Until now, the property had no effect for tab boxes.

Do you want to improve this document? Have a look at the sources on GitHub.