

Eclipse Scout Release Notes

Version 22.0

Table of Contents

About This Release	2
Demo Applications	3
UI Brush Up	4
Reworked Action Elements	4
Scroll Shadow	4
Icons for Notification and Message Boxes	5
Consistent Severity Handling	5
Better Focus Indication	6
Toggle Menu Support	6
Improved Combo Menu	6
Closable Views	6
Better Popups	7
New 3rd Party requirements	8
Java	8
Browsers	8
Native Notification Support	9
Text Post-Processors	10
Log4j support removed	11
Browser Field: Support for Posting Messages	12
REST LookupCall support for hierarchical data	13
REST API Documentation Generation: Support for inheritance	14
New Filter Field	15
Menu on Form Field	16
Lazy Loading for Tables and Detail Forms (Scout JS)	17
Focus of Invalid Field (Scout JS)	18



Looking for something else? Visit <https://eclipsescout.github.io> for all Scout related documentation.

About This Release

The latest version of this release is: 22.0.2.

You can see the [detailed change log](#) on GitHub.

Coming from an older Scout version? Check out the [Migration Guide](#) for instructions on how to obtain the new version and upgrade existing applications.

Demo Applications

The demo applications for this version can be found on the [features/version/22.0.2](#) 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/>

UI Brush Up

The user interface has been completely reworked. The goal was to modernize it and improve the usability without changing basic user interactions. Modern design principles have been applied for a fresh look, an improved user guidance and increased readability. Various widgets have been beautified like date field, smart field, table and many more.

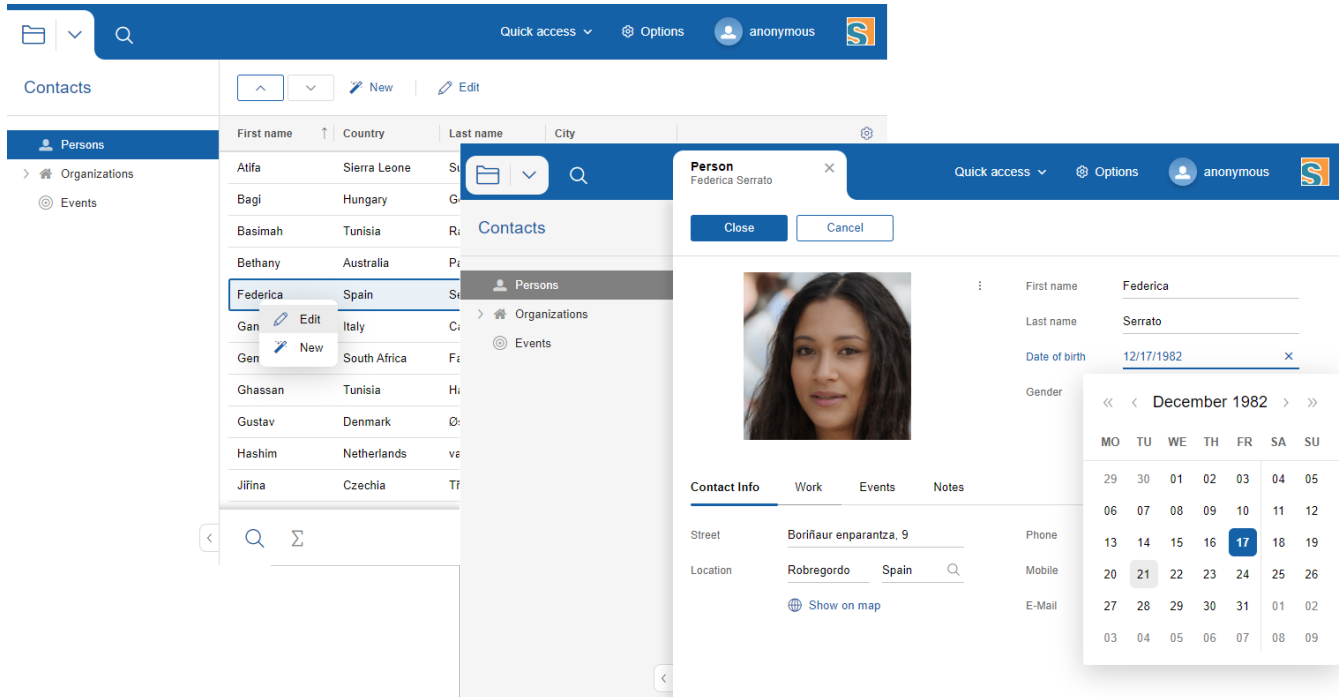


Figure 1. Brush Up

The following topics describe some notable features of the brush-up.

Reworked Action Elements

Action elements like buttons, links, menus. etc. are now more user-friendly due to increased click area, clear hover effect and aligned colors.





Figure 2. Menu

Scroll Shadow

There is now a scroll shadow on every scrollable element to indicate that not all elements are visible.

Malik	Greenland	Lundblad	Qaqortoq
Maurizia	Italy	Piccio	Villanova Del Batti...
Mulan	Switzerland	Chien	Tartegnin

Loaded
20 rows

Selected
One row

Figure 3. Scroll Shadow

Icons for Notification and Message Boxes

Notifications in GroupBoxes (`IGroupBox.setNotification()`), desktop notifications (`IDesktop.addNotification`) and message boxes now support icons.

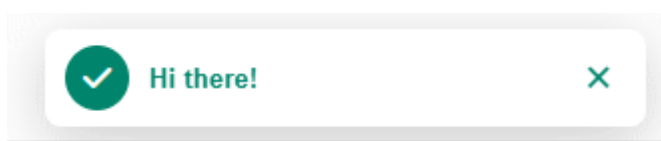
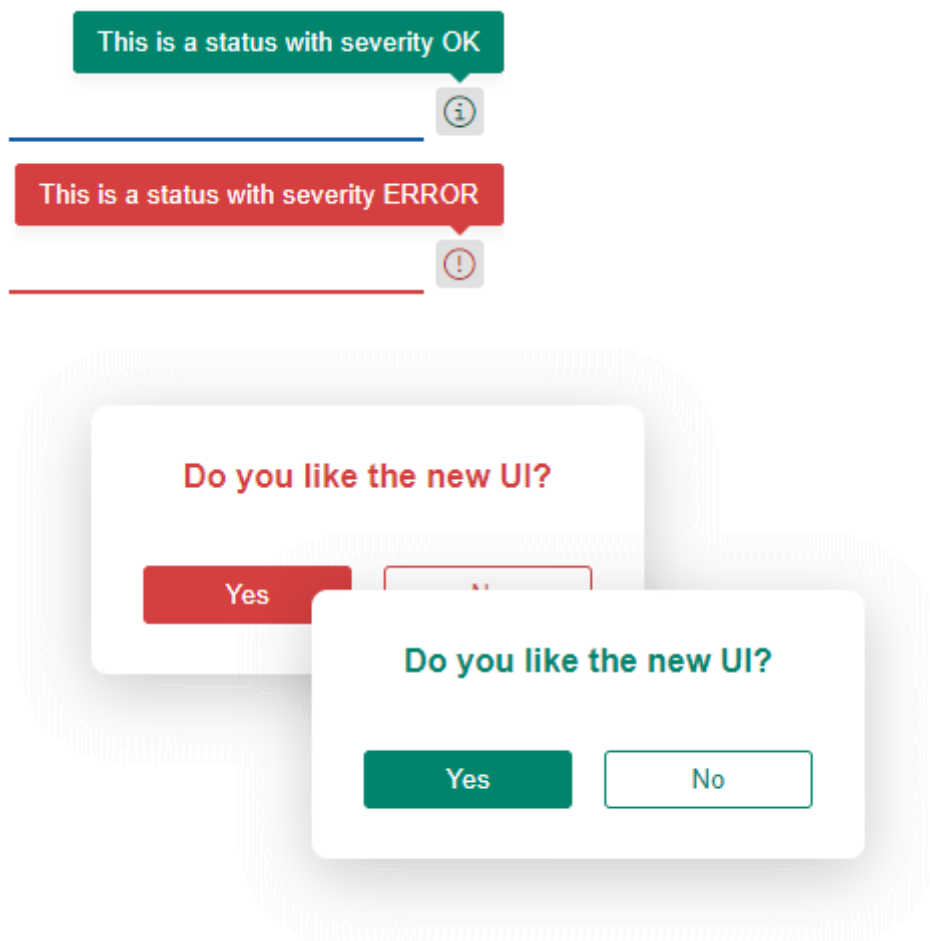


Figure 4. Notification

Consistent Severity Handling

There is a consistent coloring across tooltips, notifications and message boxes.



Better Focus Indication

Focus marker and table/tree selection have been improved to better guide the user.

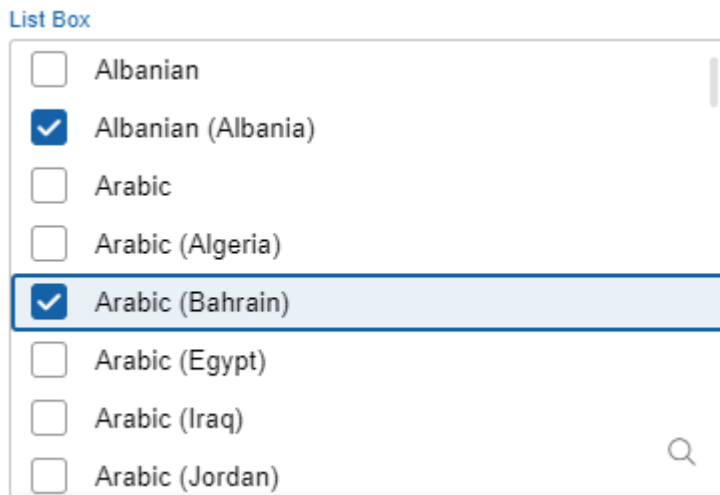


Figure 5. List Box

Toggle Menu Support

Toggable menu items now correctly visualize the selected state.

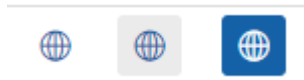


Figure 6. Toggle Menu

Improved Combo Menu

The combo menu got more flexible and can be used with the button style as well.

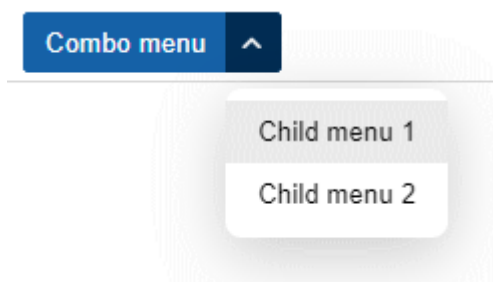


Figure 7. Combo Menu

Closable Views

Views can now be closed using an "x" icon, even if they are not active.

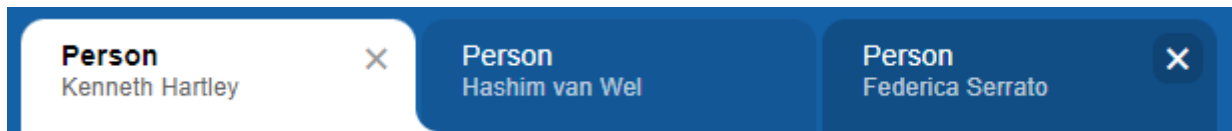


Figure 8. Views

Better Popups

Popups are now animated and have a cleaner look.

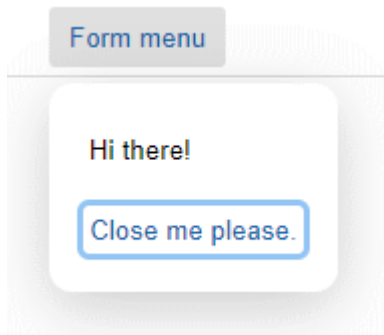


Figure 9. Popup

New 3rd Party requirements

Java

The Java 8 support has been dropped in Scout 22. Scout 22 older than 22.0.3 only supports Java 11. Scout 22.0.3 or newer supports Java 17 too. Check the migration guide for instructions on how to update to Java 17.

Furthermore, the minimal Maven version has been raised to 3.6.3.

See the migration guide for instructions on how to update.

Please note that Scout 22 makes use of **Jakarta EE 8** artifacts. These artifacts already use the new **jakarta.*** Maven coordinates but still contain the old **javax.*** Java packages. This ensures backwards compatibility with older **Java EE 8** containers. The transition to **Jakarta EE 9.1** or newer is planned for a future release.

Browsers

The minimal supported browsers have been updated. Scout 22 has the following requirements:

- Mozilla Firefox >= 69
- Chromium (like Google Chrome, Microsoft Edge, Brave or Opera) >= 71
- Apple Safari >= 12.1

Please note that Microsoft Internet Explorer and Microsoft Edge Legacy are no longer supported by Scout. If you are still using Internet Explorer or Edge Legacy, we strongly recommend updating to a newer browser.

Native Notification Support

It's now possible to send notifications via the [Notification API](#). The `DesktopNotification` has some new properties to control the behavior: **`nativeOnly`**, **`nativeNotificationVisibility`** and more.

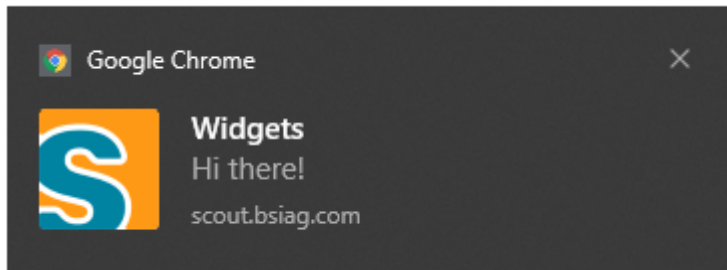


Figure 10. Native Notification

Text Post-Processors

A new bean **org.eclipse.scout.rt.platform.nls.ITextPostProcessor** has been introduced which allows to modify all texts returned by a TextProviderService.

This allows applying application wide logic on all translated texts including the ones provided by Scout itself.

Log4j support removed

The Support for [Log4j](#) logging has been dropped as Log4j 1 is end-of-life. If you still rely on Log4j 1 please consult the migration notes for instructions how to update.

Browser Field: Support for Posting Messages

Browser fields provide a new method **postMessage(data, targetOrigin)** to send arbitrary data to the embedded web page.

To prevent malicious data from being sent to your application, the origin of the message should always be validated in the callback handler (**execPostMessage** in Java, **message** event listener in Scout JS). Scout will now do this check automatically for you if you configure the valid origins in the new property **trustedMessageOrigins**.

REST LookupCall support for hierarchical data

Scout lookup calls may return hierarchically linked rows (parent-child relation). The support for hierarchical lookup rows using REST lookup calls (e.g. the `parentId` attribute definition) was moved from `AbstractLookupRowDo` to the new class `AbstractHierarchicalLookupRowDo`.

In order to support hierarchical lookups using REST, change your lookup row implementation and extend the new base class `AbstractHierarchicalLookupRowDo`.

Listing 1. Example

```
@TypeName("ExampleLookupRow")
public class ExampleLookupRowDo extends AbstractHierarchicalLookupRowDo<ExampleId> {

    @Override
    public DoValue<ExampleId> id() {
        return doValue(ID);
    }

    @Override
    public DoValue<ExampleId> parentId() {
        return doValue(PARENT_ID);
    }

    // ...
}
```

REST API Documentation Generation:

Support for inheritance

The annotation `org.eclipse.scout.rt.rest.doc.ApiDocDescription` is now marked as `@Inherited`. This allows inheritance for API documentation (e.g. on an abstract REST resource).

New Filter Field

The widgets `Table.js`, `Tree.js`, `TileGrid.js` and `TileAccordion.js` now support a text filter. This filter field is shown while typing if the widget is focused or can be opened by clicking the icon in the bottom right corner.



Figure 11. A TreeBox with the filter icon in the bottom right corner.

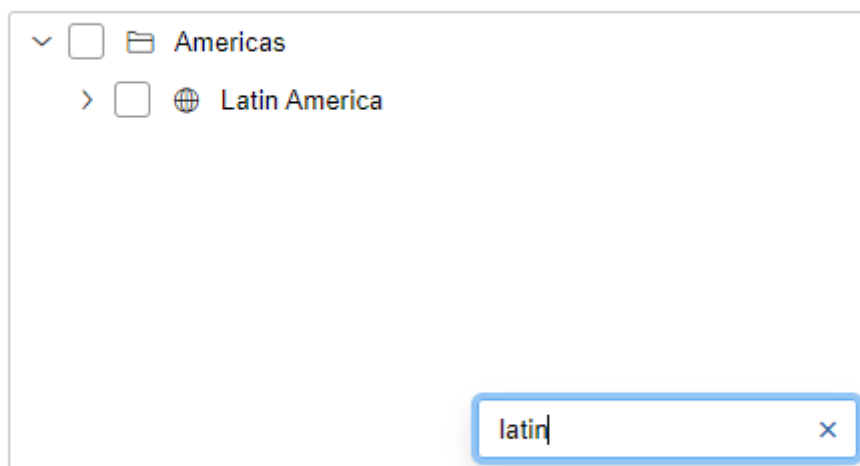


Figure 12. A TreeBox with the filter field, filtering all nodes for 'latin'.

This feature is enabled by default and can be turned off using the property `textFilterEnabled`. For more information about the filter API and the filter field see [Technical Guide for Scout JS](#).

Menu on Form Field

Until now, only some form fields supported menus. In order to generalize the functionality the menu support has been moved to `IFormField.java`. This reduces duplicated code and enables menus on every form field, e.g. on a tree field.

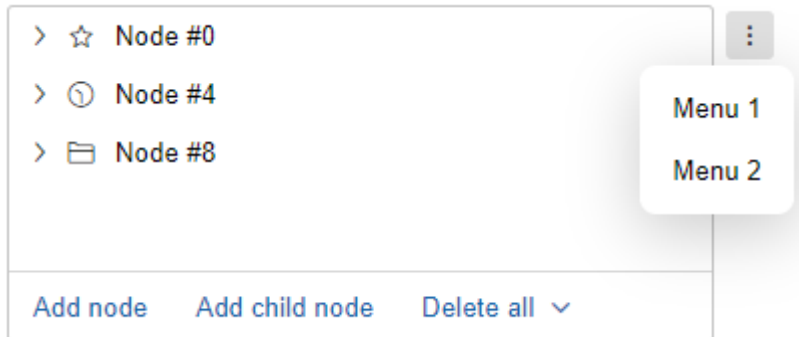


Figure 13. A TreeField with Menus

Lazy Loading for Tables and Detail Forms (Scout JS)

The detail elements of a page are now loaded lazily which improves initial loading time significantly.

Focus of Invalid Field (Scout JS)

The first invalid field now gets the focus when a form should be saved to better guide the user.



Do you want to improve this document? Have a look at the [sources](#) on GitHub.