Eclipse Scout
Migration Guide

Scout Team

Version 7.1

Table of Contents

About This Document

This document describes all relevant changes **from Eclipse Scout 7.0 to Eclipse Scout 7.1**. If existing code has to be migrated, instructions are provided here.

If you are upgrading from version 6.1, please also read the migration guide for the 7.0 (*Oxygen preview*) release:
https://eclipsescout.github.io/7.0/migration-guide.html

Service Release Migration

The following changes were made after the initial 7.1 release (Eclipse Photon release). Additionally follow these instructions when updating to a *service release*.

Photon.1 (7.1.100) Release Expected on September, 2018

Attention: The here described functionality has not yet been released and is part of an upcoming release.

API Changes (Java)

CalendarComponent

Removed deprecated method `Date[] getCoveredDays()` on `org.eclipse.scout.rt.client.ui.basic.calendar.CalendarComponent`.

Migration: Use `Range<Date> getCoveredDaysRange()` instead with start and end date.

Java 8 Required

The required Java Runtime Environment (JRE) to run an Eclipse Scout application has changed: Starting with Eclipse Scout 7.1, a **Java 8 runtime is required**.

The Scout 7.1 Runtime does not support Java 9 yet. The Java 9 support is planned for Eclipse *Photon* release (Scout 8.0) in summer 2018.

To reflect this change, some existing Scout classes have been migrated to the newly available Java 8 classes:

Old Scout class:	Replaced with Java 8 equivalent:
`org.eclipse.scout.rt.platform.filter.IFilter`	`java.util.function.Predicate`
`org.eclipse.scout.rt.platform.filter.AlwaysFilter`	`java.util.function.Predicate` e.g. `a → true`

Old Scout class:	Replaced with Java 8 equivalent:
`org.eclipse.scout.rt.platform.filter.NotFilter`	`java.util.function.Predicate#negate`
`org.eclipse.scout.rt.platform.filter.OrFilter`	`java.util.function.Predicate#or`
`org.eclipse.scout.rt.platform.util.concurrent.IConsumer`	`java.util.function.Consumer`
`org.eclipse.scout.rt.platform.util.concurrent.IFunction`	`java.util.function.Function`
`org.eclipse.scout.rt.platform.util.concurrent.IBiConsumer`	`java.util.function.BiConsumer`
`org.eclipse.scout.rt.platform.util.concurrent.IBiFunction`	`java.util.function.BiFunction`

Removed Constants/Methods From ClientUIPreferences

Since the new Html UI was introduced with Scout 5.2 some constants and methods in `ClientUIPreferences` had no effect on the UI anymore. Since Scout is now a browser application we don't have any control over the application window anymore. The form bounds are stored in the local browser storage thus they're not required in the Java model anymore.

The following constants/methods were dropped:

- ~~`APPLICATION_WINDOW_MAXIMIZED`~~
- ~~`APPLICATION_WINDOW_BOUNDS`~~
- ~~`FORM_BOUNDS`~~
- ~~`getFormBounds()`~~
- ~~`setFormBounds(IForm form, Rectangle bounds)`~~
- ~~`getApplicationWindowMaximized()`~~
- ~~`setApplicationWindowPreferences(BoundsSpec r, boolean maximized)`~~

Removed Property "focusable" From IFormField, AbstractFormField

Since the new Html UI was introduced with Scout 5.2 the property `focusable` had no effect on the UI anymore. Instead the UI uses sensible defaults for each field type. For instance: a *LabelField* is never focusable, a normal *StringField* is always focusable, as long as it is enabled. Since the property was rarely used, we removed the related code:

- Method ~~`boolean IFormField#isFocusable()`~~
- Method ~~`void IFormField#setFocusable(boolean f)`~~
- Field ~~`boolean IFormField#PROP_FOCUSABLE`~~
- Method ~~`boolean AbstractFormField#getConfiguredFocusable()`~~
- Class ~~`AbstractNonFocusableButton`~~
- Class ~~`AbstractNonFocusableRadioButton`~~

Migration: Since there is no replacement for the `focusable` property, remove all code that uses one of the methods/properties listed above.

Removal of Old "SmartField", Renaming of "SmartField2" Back to "SmartField"

In Scout 7.0 a new smart field implementation named *SmartField2* was created. The old implementation was still available, but its use was discouraged. With Scout 7.1 the old *SmartField* implementation was finally dropped. The new implementation *SmartField2* was renamed back to *SmartField*.

The following table lists the different names used in the past Scout releases. (The naming applies to all associated files, such as interfaces, abstract field classes, Java packages and JavaScript and LESS files.)

Scout <= 6.1	**Scout 7.0**	**Scout >= 7.1**
SmartField *(old implementation)*	SmartField *(old implementation)*	—
—	SmartField2 *(new implementation)*	SmartField *(new implementation)*

Migration:

- If you have already used the new *SmartField2* in Scout 7.0 you must move / rename all references back to the (new) *SmartField*.
- If you are migrating from an older Scout version (<= 6.1), you *might* have to adjust your code to the new implementation. See the corresponding 7.0 migration guide.

IContentAssistField Classes and Interfaces Removed

The base class and interfaces of the old SmartField implementation was IContentAssistField. With the new SmartField implementation this interfaces and all classes containing the word "ContentAssist" in their name, have been either removed or renamed to "SmartField". If your code references one of these classes you should simply try to rename all references. Since the API of the new SmartField is almost the same as the old, this should work and should cause no or few changes in your code. The following classes have been removed without replacement:

- ~~`org.eclipse.scout.rt.client.ui.form.fields.smartfield.AbstractMixedSmartField`~~
- ~~`org.eclipse.scout.rt.client.ui.form.fields.smartfield.ContentAssistFieldEvent`~~
- ~~`org.eclipse.scout.rt.client.ui.form.fields.smartfield.ContentAssistFieldListener`~~
- ~~`org.eclipse.scout.rt.client.ui.form.fields.smartfield.ContentAssistFieldTable`~~
- ~~`org.eclipse.scout.rt.client.ui.form.fields.smartfield.IMixedSmartField`~~

Since the new SmartField implementation does not have a proposal chooser model anymore these classes have also been removed. If you must have a special implementation of a proposal chooser, you must implement a proposal chooser in JavaScript (see: *ProposalChooser.js*), which renders the data and lookup rows it receives from the server-side SmartField. The following classes have been

removed without replacement:

- ~~org.eclipse.scout.rt.client.ui.form.fields.smartfield.AbstractProposalChooser~~
- ~~org.eclipse.scout.rt.client.ui.form.fields.smartfield.IProposalChooser~~
- ~~org.eclipse.scout.rt.client.ui.form.fields.smartfield.IProposalChooserProvider~~
- ~~org.eclipse.scout.rt.client.ui.form.fields.smartfield.TableProposalChooser~~
- ~~org.eclipse.scout.rt.client.ui.form.fields.smartfield.TreeProposalChooser~~

GroupBox: Moved "minWidthInPixel"

The property `minWidthInPixel` has been moved to *LogicalGridLayoutConfig*.

Migration: Instead of using `getConfiguredMinWidthInPixel` you should now set the property as follows:

```
@Override
protected LogicalGridLayoutConfig getConfiguredBodyLayoutConfig() {
  return super.getConfiguredBodyLayoutConfig()
      .withMinWidth(400);
}
```

Adjusted Behavior of Widget Initialization

The goal was to harmonize all the init methods (`initField`, `initTile`, `initForm` etc.) and to make sure, `init()` is only executed once. This is important for dynamic widgets like accordion or tiles. These widgets initialize the newly added children by themselves so that the caller does not need to take care of it. For these cases it is important that `init()` is not called multiple times.

But: there may be cases which require `init()` to be called multiple times, like reseting a search form. For such cases, `reinit()` has to be used from now on. Also, after the widget is disposed, `init()` may be called again. So remember: `execInit` may be called more than once in some circumstances. This is existing behavior!

We also renamed the initConfig guard of `IFormField` from `isInitialized` to `isInitDone` to make clear what initialization has been done. It has furthermore been moved to `AbstractWidget` so that individual widgets don't have to care about it and to use the same pattern as for `postInitConfig`, `init` and `dispose`.

Migration:

If you used one of the deprecated methods (`initField`, `initAction` etc.), replace them with one of the following methods: `init`, `reinit` or `initInternal`.

- Use `init` if you created a field and need to initialize it.
- Use `reinit` if you explicitly want to reinitialize an already initialized field.
- Use `initInternal` if your custom widget overrides `initField`.

GzipServletFilter

Replaced init parameters `get_pattern` and `post_pattern` with `content_types`. If you set these init parameters in your web.xml, replace or remove them accordingly.

HttpCacheControl

The argument `pathInfo` has been removed from the method `HttpCacheControl.checkAndSetCacheHeaders` since it has no effect anymore.

HttpProxy

HTTP Proxy doesn't set cache control `no-chache` header anymore.

Config Properties

Descriptions

Config properties based on `org.eclipse.scout.rt.platform.config.IConfigProperty` include a description text. This description is stored in the new `description()` method.

All properties must now implement this new method and return a description text of that property. The class `org.eclipse.scout.rt.platform.config.ConfigDescriptionExporter` can be used to export these descriptions. By default an AsciiDoctor exporter is included.

Default value

Config properties based on `org.eclipse.scout.rt.platform.config.IConfigProperty` include a default value. The default value is stored in the `getDefaultValue()` method.

The method was moved from `org.eclipse.scout.rt.platform.config.AbstractConfigProperty<DATA_TYPE, RAW_TYPE>` to the interface. Therefore the visibility has changed from protected to public.

Validation

The concreate implementation `org.eclipse.scout.rt.platform.config.ConfigPropertyValidator` which validates the configuration of `config.properties` files will also check if a configured value matches the default value. In case it does a info message (warn in development mode) will be logged but the platform will still start.

To minimize configuration files such entries should be removed from `config.properties` files.

Renamed Config Property Keys

The following config property keys have been renamed (the old keys are no longer valid and must be renamed accordingly):

Table 1. Config Property Renames

Old Key	New Key
scout.auth.anonymous.enabled	scout.auth.anonymousEnabled
scout.auth.cookie.enabled	scout.auth.cookieEnabled
scout.auth.cookie.maxAge	scout.auth.cookieMaxAge
scout.auth.cookie.name	scout.auth.cookieName
scout.auth.cookie.session.validate.secure	scout.auth.cookieSessionValidateSecure
scout.auth.credentials.plaintext	scout.auth.credentialsPlaintext
scout.auth.token.ttl	scout.auth.tokenTtl
scout.server.url	scout.backendUrl
session.jobCompletionDelayOnSessionShutdown	scout.client.jobCompletionDelayOnSessionShutdown
org.eclipse.scout.memory	scout.client.memoryPolicy
notification.user.authenticator	scout.client.notificationSubject
org.eclipse.scout.testing.client.ClientSessionProviderWithCache#expiration	scout.client.testingSessionTtl
user.area	scout.client.userArea
org.eclipse.scout.rt.server.clientnotification.ClientNotificationService#maxMessages	scout.clientnotification.chunkSize
org.eclipse.scout.rt.server.clientnotification.ClientNotificationService#blockingTimeout	scout.clientnotification.maxNotificationBlockingTimeOut
org.eclipse.scout.rt.server.clientnotification.ClientNotificationNodeQueue#capacity	scout.clientnotification.nodeQueueCapacity
org.eclipse.scout.rt.server.clientnotification.ClientNotificationRegistry#m_queueExpireTime	scout.clientnotification.notificationQueueExpireTime
org.eclipse.scout.rt.server.services.common.clustersync.ClusterSynchronizationService#user	scout.clustersync.user
scout.beans.createTunnelToServerBeans	scout.createTunnelToServerBeans
scout.csp.enabled	scout.cspEnabled
scout.csp.directive	scout.cspDirective
scout.dev.mode	scout.devMode
scout.external.base.url	scout.externalBaseUrl
scout.healthcheck.remoteUrls	scout.healthCheckRemoteUrls
scout.http.apache_connection_time_to_live	scout.http.connectionTtl
scout.http.ignore_proxy	scout.http.ignoreProxyPatterns
scout.http.apache_keep_alive	scout.http.keepAlive
scout.http.apache_max_connections_per_route	scout.http.maxConnectionsPerRoute
scout.http.apache_max_connections_total	scout.http.maxConnectionsTotal
scout.http.proxy	scout.http.proxyPatterns
scout.http.apache_retry_post	scout.http.retryPost
scout.http.transport_factory	scout.http.transportFactory
org.eclipse.scout.rt.server.services.common.imap.AbstractIMAPService#host	scout.imap.host
org.eclipse.scout.rt.server.services.common.imap.AbstractIMAPService#mailbox	scout.imap.mailbox

Old Key	New Key
org.eclipse.scout.rt.server.services.common.imap.AbstractIMAPService#password	scout.imap.password
org.eclipse.scout.rt.server.services.common.imap.AbstractIMAPService#port	scout.imap.port
org.eclipse.scout.rt.server.services.common.imap.AbstractIMAPService#sslProtocols	scout.imap.sslProtocols
org.eclipse.scout.rt.server.services.common.imap.AbstractIMAPService#userName	scout.imap.username
jandex.rebuild	scout.jandex.rebuild
jaxws.consumer.connectTimeout	scout.jaxws.consumer.connectTimeout
jaxws.consumer.portCache.corePoolSize	scout.jaxws.consumer.portCache.corePoolSize
jaxws.consumer.portCache.enabled	scout.jaxws.consumer.portCache.enabled
jaxws.consumer.portCache.ttl	scout.jaxws.consumer.portCache.ttl
jaxws.consumer.portPool.enabled	scout.jaxws.consumer.portPoolEnabled
jaxws.consumer.readTimeout	scout.jaxws.consumer.readTimeout
jaxws.implementor	scout.jaxws.implementor
jaxws.loghandler.debug	scout.jaxws.loghandlerDebug
jaxws.provider.authentication.basic.realm	scout.jaxws.provider.authentication.basicRealm
jaxws.provider.user.authenticator	scout.jaxws.provider.user.authenticator
jaxws.provider.user.handler	scout.jaxws.provider.user.handler
scout.mom.requestreply.cancellation.topic	scout.mom.requestreply.cancellationTopic
scout.node.id	scout.nodeId
scout.permission.level.check.cache.ttl	scout.permissionLevelCacheTtl
org.eclipse.scout.rt.server.services.common.file.RemoteFileService#rootPath	scout.remotefileRootPath
org.eclipse.scout.rt.server.session.ServerSessionProviderWithCache#expiration	scout.serverSessionTtl
org.eclipse.scout.serviceTunnel.compress	scout.servicetunnel.compress
org.eclipse.scout.rt.servicetunnel.apache_max_connections_per_route	scout.servicetunnel.maxConnectionsPerRoute
org.eclipse.scout.rt.servicetunnel.apache_max_connections_total	scout.servicetunnel.maxConnectionsTotal
org.eclipse.scout.rt.servicetunnel.targetUrl	scout.servicetunnel.targetUrl
org.eclipse.scout.rt.server.services.common.smtp.AbstractSMTPService#debugReceiverEmail	scout.smtp.debugReceiverEmail
org.eclipse.scout.rt.server.services.common.smtp.AbstractSMTPService#defaultFromEmail	scout.smtp.defaultFromEmail
org.eclipse.scout.rt.server.services.common.smtp.AbstractSMTPService#host	scout.smtp.host
org.eclipse.scout.rt.server.services.common.smtp.AbstractSMTPService#password	scout.smtp.password
org.eclipse.scout.rt.server.services.common.smtp.AbstractSMTPService#port	scout.smtp.port
org.eclipse.scout.rt.server.services.common.smtp.AbstractSMTPService#sslProtocols	scout.smtp.sslProtocols
org.eclipse.scout.rt.server.services.common.smtp.AbstractSMTPService#subjectPrefix	scout.smtp.subjectPrefix

Old Key	New Key
org.eclipse.scout.rt.server.services.common.smtp.AbstractSMTPService#useAuthentication	scout.smtp.useAuth
org.eclipse.scout.rt.server.services.common.smtp.AbstractSMTPService#username	scout.smtp.username
org.eclipse.scout.rt.server.services.common.smtp.AbstractSMTPService#useSmtps	scout.smtp.useSsl
org.eclipse.scout.rt.server.services.common.jdbc.AbstractSqlService#directJdbcConnection	scout.sql.directJdbcConnection
org.eclipse.scout.rt.server.services.common.jdbc.AbstractSqlService#jdbcDriverName	scout.sql.jdbc.driverName
org.eclipse.scout.rt.server.services.common.jdbc.AbstractSqlService#jdbcDriverUnload	scout.sql.jdbc.driverUnload
org.eclipse.scout.rt.server.services.common.jdbc.AbstractSqlService#jdbcMappingName	scout.sql.jdbc.mappingName
org.eclipse.scout.rt.server.services.common.jdbc.AbstractSqlService#jdbcPoolConnectionBusyTimeout	scout.sql.jdbc.pool.connectionBusyTimeout
org.eclipse.scout.rt.server.services.common.jdbc.AbstractSqlService#jdbcPoolConnectionLifetime	scout.sql.jdbc.pool.connectionIdleTimeout
org.eclipse.scout.rt.server.services.common.jdbc.AbstractSqlService#jdbcPoolSize	scout.sql.jdbc.pool.size
org.eclipse.scout.rt.server.services.common.jdbc.AbstractSqlService#jdbcProperties	scout.sql.jdbc.properties
org.eclipse.scout.rt.server.services.common.jdbc.AbstractSqlService#jdbcStatementCacheSize	scout.sql.jdbc.statementCacheSize
org.eclipse.scout.rt.server.services.common.jdbc.AbstractSqlService#jndiInitialContextFactory	scout.sql.jndi.initialContextFactory
org.eclipse.scout.rt.server.services.common.jdbc.AbstractSqlService#jndiName	scout.sql.jndi.name
org.eclipse.scout.rt.server.services.common.jdbc.AbstractSqlService#jndiProviderUrl	scout.sql.jndi.providerUrl
org.eclipse.scout.rt.server.services.common.jdbc.AbstractSqlService#jndiUrlPkgPrefixes	scout.sql.jndi.urlPkgPrefixes
org.eclipse.scout.rt.server.services.common.jdbc.AbstractSqlService#password	scout.sql.password
org.eclipse.scout.rt.server.services.common.jdbc.AbstractSqlService#transactionMemberId	scout.sql.transactionMemberId
org.eclipse.scout.rt.server.services.common.jdbc.AbstractSqlService#username	scout.sql.username
scout.text.providers.show.keys	scout.texts.showKeys
scout.background.polling.interval	scout.ui.backgroundPollingMaxWaitTime
scout.max.user.idle.time	scout.ui.maxUserIdleTime
scout.ui.model.jobs.await.timeout	scout.ui.modelJobTimeout
scout.sessionstore.housekeepingDelay	scout.ui.sessionstore.housekeepingDelay
scout.sessionstore.housekeepingMaxWaitForShutdown	scout.ui.sessionstore.housekeepingMaxWaitForShutdown
scout.sessionStore.maxWaitForAllShutdown	scout.ui.sessionStore.maxWaitForAllShutdown

Old Key	New Key
scout.sessionStore.valueUnboundMaxWaitForWriteLock	scout.ui.sessionStore.valueUnboundMaxWaitForWriteLock

Customizing CSP Rules Via Config Property

The new config property `scout.cspDirective` makes subclassing and replacing the `ContentSecurityPolicy` class obsolete as you can configure all CSP settings with this property now. An example from the Scout Widgets application:

```
@Replace
public class WidgetsContentSecurityPolicy extends ContentSecurityPolicy {

  @Override
  protected void initDirectives() {
    super.initDirectives();
    // Demo app uses external images in html field and custom widgets -> allow it
    withImgSrc("*");
  }
}
```

This class was deleted and replaced by a config property in *config.properties*:

```
# CSP - Demo app uses external images in html field and custom widgets -> allow it
scout.cspDirective[img-src]=*
```

IUiServletRequestHandler

The methods `handleGet` and `handlePost` on `IUiServletRequestHandler` were replaced by the single method `handle`. This new method is called for all HTTP methods.

To retrieve the HTTP method, call `getMethod` on `HttpServletRequest`. When using `AbstractUiServletRequestHandler` no migration should be required because `AbstractUiServletRequestHandler` delegates to the Java methods for the common HTTP methods `handleGet`, `handlePost`, `handlePut` and `handleDelete`.

Methods `proxyGet` and `proxyPost` on `HttpProxy` are replaced by the common method `proxy`.

API Changes (JavaScript)

Rename of LESS Variables

If you created a custom theme, you might have to adjust some LESS variables.

- Splitted `@group-title-padding-y` into `@group-box-title-padding-top` and `@group-box-title-padding-bottom`
- Splitted `@tree-node-padding` into `@tree-node-padding-y`, `@tree-node-padding-left` and `@tree-node-padding-right`
- Renamed `@group-title-border-width` to `@group-box-title-border-width`
- Renamed `@group-margin-bottom` to `@group-box-body-padding-bottom`
- Renamed `@group-margin-top` to `@group-box-body-padding-top`
- Added `@group-box-title-margin-top`
- Renamed `@tabbox-padding-x` to `@tab-item-padding-x`
- Renamed `@tabbox-focus-arrow-width` to `@tab-item-focus-arrow-width`
- Renamed `@tabbox-border-width` to `@tab-area-border-width`
- Renamed `@compact-outline-node-padding-v` to `@compact-outline-node-padding-y`
- Renamed `@box-margin-v` to `@box-margin-y`
- Renamed `@outline-breadcrumb-node-padding-v` to `@outline-breadcrumb-node-padding-y`
- Renamed `@tile-padding-h` to `@tile-field-padding-x`
- Renamed `@tile-padding-v` to `@tile-field-padding-y`
- Renamed `@planner-header-buttons` to `@planner-header-button-height`
- Renamed `@calendar-header-buttons` to `@calendar-header-button-height`
- Renamed `@logical-grid-height` to `@logical-grid-row-height`
- Renamed `@applink-color` to `@link-color`

Automatic Preloading of Web Fonts

Scout can now detect the Web Fonts (*.woff files) to preload automatically. It's therefore no longer necessary to list the font names manually in the boostrap argument of `scout.App`.

Migration:

Remove the `fonts` property from the bootstrap parameter object passed to the *init()* function of your Scout app.

For example, the default *index.js* file generated by the Scout "helloworld" archetype looks like this:

```
$(document).ready(function() {
  var app = new scout.RemoteApp();
  app.init({
    bootstrap: {
      fonts: ['scoutIcons'] // <-- this property is no longer required
    }
  });
});
```

If no other init options remain, the file can be simplified to:

```
$(document).ready(function() {
  var app = new scout.RemoteApp();
  app.init();
});
```

To find all files that need migration, search for the text `bootstrap: {` in all **.js* files in your workspace. The files are called *index.js* by default and are usually located at *your.project.ui.html/src/main/resources/WebContent/res*.

This migration is recommended but optional. Listing all fonts to preload manually still works. To disable font preloading entirely, set the *fonts* bootstrap property to an empty array `[]`.

Other Changes

—

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