

# Eclipse Scout

## *Migration Guide*

Scout Team

Version 10.0

# Table of Contents

About This Document .....	1
Service Releases .....	1
API Changes (Java) .....	1
StrictSimpleDateFormat .....	1
ObjectUtility .....	1
Data Objects .....	1
CacheBuilder .....	2
Move ICache and transactional Map .....	2
Authorization API .....	2
TestingUtility → BeanTestingHelper .....	3
API Changes (JavaScript) .....	3
REST Service Changes .....	3
Renamings in ErrorDo .....	4
Different HTTP status codes .....	4

# About This Document

This document describes all relevant changes **from Eclipse Scout 9.0 to Eclipse Scout 10.0**. If existing code has to be migrated, instructions are provided here.

## Service Releases

Scout 10.0 will continue to be maintained for a while and a new build may be released from time to time, following the [Simultaneous Release Cycle](#). Beside bugfixes, these service releases may even contain some minor features.

The following changes were made after the initial 10.0 release.

### Simrel 2020-03 — TBD



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

## API Changes (Java)

### StrictSimpleDateFormat

`org.eclipse.scout.rt.jackson.dataobject.StrictSimpleDateFormat` was removed. Use `org.eclipse.scout.rt.platform.util.date.StrictSimpleDateFormat` instead.

### ObjectUtility

`nv1Optional()` was renamed to `nv1Opt()`.

## Data Objects

The Scout data object support was moved from the Scout platform to the module `org.eclipse.scout.rt.dataobject`. The package imports of all data object related classes therefore changed: From `org.eclipse.scout.rt.platform.dataobject` to `org.eclipse.scout.rt.dataobject`

### Renamings

`org.eclipse.scout.rt.client.ui.desktop.datachange.DoChangeEvent` →  
`org.eclipse.scout.rt.client.ui.desktop.datachange.ItemDataChangeEvent`

### Dependencies

All modules which use data objects were extended with a dependency to `org.eclipse.scout.rt.dataobject`

- `org.eclipse.scout.rt.rest`

- `org.eclipse.scout.rt.mom.api`

## Renamings in ErrorDo

- `org.eclipse.scout.rt.rest.error.ErrorDo#status` → `org.eclipse.scout.rt.rest.error.ErrorDo#httpStatus`
- `org.eclipse.scout.rt.rest.error.ErrorDo#code` → `org.eclipse.scout.rt.rest.error.ErrorDo#errorCode`

## CacheBuilder

The following methods on `CacheBuilder` were removed, since they were unused and covered unused, old functionality:

- Method `org.eclipse.scout.rt.shared.cache.CacheBuilder.addCacheInstance(ICache<K, V>)`
- Method `org.eclipse.scout.rt.shared.cache.CacheBuilder.getCacheInstances()`

## Move ICache and transactional Map

`AbstractTransactionalMap` and its concrete implementations `ConcurrentTransactionalMap` and `CopyOnWriteTransactionalMap` have been moved to `org.eclipse.scout.rt.platform.util.collection`.

`ICache`, its implementations and cache wrappers have been moved to `org.eclipse.scout.rt.platform.cache`.

## Authorization API

The authorization API of scout was extended and moved from `org.eclipse.scout.rt.shared` into its own module. You may check the technical guide for further details.

- Introduced `IPermissionCollection` and `IPermission` interfaces
- Let all current scout permission (e.g. `CopyToClipboardPermission`) implement `IPermission`
- All scout permission names are now prefixed with `scout.`
- `RemoteServiceAccessPermission#getName` returns a stable name instead of the service operation pattern
- Deleted `BasicHierarchyPermission`. If required, you may copy from an older version of scout.
- `org.eclipse.scout.rt.shared.services.common.security.IAccessControlService` moved to `org.eclipse.scout.rt.security`
- `IAccessControlService#getPermissionLevel` removed; use `ACCESS#getGrantedPermissionLevel` instead
- `IAccessControlService#checkPermission` removed; use instead `ACCESS#check`
- `IAccessControlService#getPermissions` must now **never** return `null`. Instead `NonePermissionCollection` or `AllPermissionCollection` may be returned.
- `org.eclipse.scout.rt.shared.services.common.security.ACCESS` moved to `org.eclipse.scout.rt.security.ACCESS`

- `ACCESS#check` now fails if argument is `null` (before succeeds).
- `org.eclipse.scout.rt.shared.services.common.security.AbstractAccessControlService` moved to `org.eclipse.scout.rt.security`
- `AbstractAccessControlService#getIdOfCurrentUser` moved to `Sessions#getCurrentUserId()`

## Load Permissions

With the new `IPermissionCollection`, loading of permissions in `AbstractAccessControlService#execLoadPermissions` has changed.

- Create a new instance by calling `BEANS.get(DefaultPermissionCollection.class)` instead of `new java.security.Permissions()`.
- Add permissions with a permission level: `permissions.add(new ReadUsersPermission(), PermissionLevel.ALL);`
- Do not forget to set permission collection as read only: `permissions.setReadOnly();`

There is also a `AllPermissionCollection` which may be used instead of `DefaultPermissionCollection`.

## TestingUtility → BeanTestingHelper

The following methods are deprecated. Use the corresponding methods on `BeanTestingHelper` via `BeanTestingHelper.get()` instead:

- `registerBeans`
- `registerBean`
- `unregisterBean`
- `unregisterBeans`
- `mockConfigProperty`

The following replacement regex can be applied on all Java files:

`\bTestingUtility\.(registerBeans|registerBean|unregisterBean|unregisterBeans|mockConfigProperty)` to `BeanTestingHelper.get().$1`

The following methods are deprecated and will be removed in a future release without a replacement:

- `registerWithReplace`
- `registerWithTestingOrder`
- `clearHttpAuthenticationCache`

## API Changes (JavaScript)

## REST Service Changes

Any changes which may change how REST consumer or provider behave.

# Renamings in ErrorDo

`org.eclipse.scout.rt.rest.error.ErrorDo` used by  
`org.eclipse.scout.rt.rest.client.proxy.ErrorDoRestClientExceptionTransformer` and some  
`org.eclipse.scout.rt.rest.exception.AbstractExceptionMapper<E>` was slightly changed:

- `ErrorDo#status` → `ErrorDo#httpStatus`
- `ErrorDo#code` → `ErrorDo#errorCode`

## Different HTTP status codes

A REST service client using `ErrorDoRestClientExceptionTransformer` will now transform

- any client request error (HTTP `4xx` status codes) into a `VetoException`
- `403 - Forbidden` into a `org.eclipse.scout.rt.dataobject.exception.AccessForbiddenException`
- `404 - Not Found` into a `org.eclipse.scout.rt.dataobject.exception.ResourceNotFoundException`

The `org.eclipse.scout.rt.rest.exception.VetoExceptionMapper` used by a REST service provide will now create an error response with status `400 - Bad Request` (this was formerly a `403`).



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