Common Codeunits

- CodeunitTranslator
  - IntegrationFieldFixCPR
  - IntegrationFieldFixDate
  - IntegrationFieldLookupKey
  - IntegrationFieldValueMap
- StatusAction
  - SignNemIdActionStatus
  - SignNemIdActionStatusGenerator
  - ChangeParentStatusAction
  - Set Date On Status
- FormEvent
  - BuildUserProfileFromRecord
  - Gantt Sync CodeUnit
  - Recalculate Parent
  - LoadInfoFromCVR
  - Http request on save
- Public Pages
  - Hello World
  - PublicTokenBuilder
  - MigrateFilesToChildren
- Pages
  - Clone System
  - Copy Structure

- Clone records
- Static Content
- Users By Group
- Users Online
- Website Iframe
- Solution Structure View
- Gantt Diagram
- Move a File
- Convert database to UTF-8
- Bulk importing files
- Update resume for entire entity
- Cleanup Inactive Model Parts
- Fix Docx Tags
- Reinvite Users
- Generate Video Thumbnails
- Export Template To WebDav
- Bulk import files
- Dynamically change users exclusive group
- Is Date weekend/holiday

CodeunitTranslator

IntegrationFieldFixCPR

What it does

Cleans up external data in CPR format, for use in a scheduled integration.

String are cleaned using the CPR translateInput method. Data will be cleaned before inserting without the use of this codeunit, but lookups and such might fail if the CPR is formatted poorly.

How to invoke

Insert value in field mapping:

Designer > Integration > Import sources > [integration] > [fieldmapping] > Codeunit handler

```
dk.p2e.blanket.codeunit.common.IntegrationFieldFixCPR
```

Configuration

None.

Developer info

- Type: CodeunitTranslator
- Security: None
- Classpath: dk.p2e.blanket.codeunit.common.IntegrationFieldFixCPR

IntegrationFieldFixDate

What it does

Cleans up external data in date format, for use in a scheduled integration.

String are cleaned using the date translateInput method. Data will be cleaned before inserting without the use of this codeunit, but lookups and such might fail if the date is not formatted as YYYY-MM-DD.

How to invoke

Insert value in field mapping Designer > Integration > Import sources > [integration] > [fieldmapping] > Codeunit handler

```
dk.p2e.blanket.codeunit.common.IntegrationFieldFixDate
```

Configuration

[All possible configurations and where to edit them]

Developer info

- Type: CodeunitTranslator
- Security: None
- Classpath: dk.p2e.blanket.codeunit.common.IntegrationFieldFixDate

IntegrationFieldLookupKey

What it does

Looks up another value based on the string provided, for use in a scheduled integration.

A lookup will be executed substituting the @VALUE@ in the SQL expression passed to the codeunit.

Note that the @VALUE@ is SQL escaped for security reasons.

How to invoke

Insert value in field mapping Designer > Integration > Import sources > [intergration] > [fieldmapping] > Codeunit handler

```
dk.p2e.blanket.codeunit.common.IntegrationFieldLookupKey
```

Configuration

Insert an SQL expression in:

[fieldmapping] > Configuration values

Example:

```
SELECT Name FROM user WHERE Email = '@VALUE@';
```

Developer info

- Type: CodeunitTranslator
- Security: None

- Classpath: dk.p2e.blanket.codeunit.common.IntegrationFieldLookupKey

IntegrationFieldValueMap

What it does

Looks up another value based on the string provided, for use in a scheduled integration.

Values will be swapped from a list provided to this codeunit.

An empty string will be returned if no match can be found.

How to invoke

Insert value in field mapping Designer > Integration > Import sources > [integration] > [fieldmapping] > Codeunit handler

```
dk.p2e.blanket.codeunit.common.IntegrationFieldValueMap
```

Configuration

Insert value maps seperated by "@" in:

[fieldmapping] > Configuration values

Example:

```
Tesla Model S@Large car
Hummer@Large car
Toyota Corolla@Medium car
Suzuki Swift@Small car
```

Developer info

- Type: CodeunitTranslator
- Security: None
- Classpath: dk.p2e.blanket.codeunit.common.IntegrationFieldValueMap

StatusAction

StatusAction

SignNemIdActionStatus

What it does

1. Send a document from a document field to signing with NemID.
2. The user will be notified via email containing a link for where to sign the document
3. When signing a document is generated including a NemID signature
4. After signing the documents saved in the same field carrying the postfix _signed

How to set up

Add a status action and set the codeunit Classname to:
dk.tempusserva.signing.nemid.SignNemIdActionStatus

Set the following entity configurations

- **Signer.FieldFil** Field NAME containing the file to be signed
- **Signer.FieldFilID** Field ID containing the file to be signed
- **Signer.FieldEmail** Field NAME containing the email of the signer
- **Signer.FieldCPR** Reserved for future use (specify any field)
- **Signer.ServerToken** Code to connect to the server (ask your TS contact)
- **Signer.StatusError** StatusID to set on the data if failed
- **Signer.StatusSigned** StatusID to set on the data if succes
- **Signer.EmailSubject** Header on the email sent to the signer
- **Signer.EmailBody** Content of the email sent to the signer

Developer info

- Type: CodeunitStatusaction
- Security: none
- Classpath: dk.tempusserva.signing.nemid.SignNemIdActionStatus

SignNemIdActionStatusGenerator

What it does

1. Generates a document based on a defined template
2. Send a document from a document field to signing with NemID.
3. The user will be notified via email. The message contains a link to where to sign the document
4. When signing a document is generated including a NemID signature
5. After signing, the document is saved in the same field carrying the postfix _signed

All steps after 1 are the same as SignNemIdActionStatus

How to setup

Add a status action and set the codeunit Classname to:
dk.tempusserva.signing.nemid.SignNemIdActionStatusGenerator

Set the following entity configurations (same as SignNemIdActionStatus)

- **Signer.FieldFil** Field NAME containing the file to be signed
- **Signer.FieldFilID** Field ID containing the file to be signed
- **Signer.FieldEmail** Field NAME containing the email of the signer
- **Signer.FieldCPR** Reserved for future use (specify any field)
- **Signer.ServerToken** Code to connect to the server (ask your TS contact)
- **Signer.StatusError** StatusID to set on the data if failed
- **Signer.StatusSigned** StatusID to set on the data if succes
- **Signer.EmailSubject** Header on the email sent to the signer
- **Signer.EmailBody** Content of the email sent to the signer

In addition the following entity configurations must be set

- **Signer.TemplateID** What template to use to generate the output
- **Signer.FieldID** Field to store the output in
- **Signer.FileName** Name of the output file (ex. "contract.docx")

Developer info

- Type: CodeunitStatusaction
- Security: none
- Classpath: dk.tempusserva.signing.nemid.SignNemIdActionStatusGenerator

ChangeParentStatusAction

What it does

Changes the status of a parent record, once all siblings are in a given state(status).

First setup

Set the variables

- myParentField
- parentStatusID

How to invoke

Add a status action that invokes this codeunit.

Options

If the setup is unconventional you can overwrite the parent reference, using a config named "parentEntity" and you can include records in "hidden" states by setting "excludeHiddenStatus" to "false".

Configuration

All configs for this codeunit support adding ".[StatusID]" add the end of the config name, to set a specific value, for that status. Otherwise the codeunit will fallback to the config without ".[StatusID]" at the end.

Developer info

- Type: CodeunitStatusaction
- Security: none
- Classpath: com.tsnocode.codeunit.common.ChangeParentStatusAction

StatusAction

Set Date On Status

What it does

Sets one or more fields to today's date when the codeunit is run.

First setup

Create a Configuration on the entity named "SetDateOnStatus.[StatusID]", which contains the system name(s) of the field(s) to be corrected.
If multiple fields, separate with spaces.

How to invoke

Add the codeunit to a status action.

Configuration

[All possible configurations and where to edit them]

Developer info

- Type: CodeunitStatusaction
- Security: access to the record
- Classpath: com.tsnocode.codeunit.common.SetDateOnStatus

FormEvent

FormEvent

BuildUserProfileFromRecord

What it does

Creates a TS user from data in a record (name, email)

- Creates a user
- Assigns to a group
- Sends profile to user

How to invoke

Set the codeunit on the entity to:
dk.p2e.blanket.codeunit.common.BuildUserProfileFromRecord

Configuration

Set the folowing configutations

- fieldUserLink
- fieldDataName
- fieldDataMail
- defaultGroupID
- requiredStatusID

Developer info

- Type: CodeunitFormevents
- Security: None
- Classpath: dk.p2e.blanket.codeunit.common.BuildUserProfileFromRecord

FormEvent

Gantt Sync CodeUnit

What it does

This is a codeunit that syncs data from one entity to another.
It also calculates start/end dates if a duration and only one of the dates are given.
If all three are given, an error will be thrown.
It was developed to sync data from a couple of entities to a single one, to display the data as a gantt chart.

Notes

If a duration field is set in the config, the following logic is used when calculating start and end dates.
Default is to copy the dates.
If one of the dates isn't given, then it is calculated based on duration (whole days) and the given date.

All statusIDs have to be set and it isn't possible to map from multiple states to a single state.

How to invoke

[Steps to enable codeunit]

Configuration

Configuration	Description
StructureMigrationCodeUnit.From.Field.Duration	Name of the column in the database containing the duration of the task (not required)
StructureMigrationCodeUnit.From.Field.EndDate	Name of the column in the database containing the enddate of the task

StructureMigrationCodeUnit.From.Field.Parent	Name of the column in the database containing the DataID of the parent task
StructureMigrationCodeUnit.From.Field.Progress	Name of the column in the database containing the percentage progress of the task
StructureMigrationCodeUnit.From.Field.Serial	Name of the column in the database containing the unique "code" of the task
StructureMigrationCodeUnit.From.Field.StartDate	Name of the column in the database containing the startdate of the task
StructureMigrationCodeUnit.From.Field.Title	Name of the column in the database containing the title of the task
StructureMigrationCodeUnit.To.Field.EndDate	Name of the column in the database where the end will be stored
StructureMigrationCodeUnit.To.Field.Parent	Name of the column in the database where the DataID of the parent task will be stored
StructureMigrationCodeUnit.To.Field.Progress	Name of the column in the database where the percentage progress of the task will be stored
StructureMigrationCodeUnit.To.Field.Serial	Name of the column in the database where the unique "code" of the task will be stored
StructureMigrationCodeUnit.To.Field.StartDate	Name of the column in the database where the startdate of the task will be stored
StructureMigrationCodeUnit.To.Field.Title	Name of the column in the database where the title of the task will be stored
StructureMigrationCodeUnit.To.Table	Name of the table in the database where all the tasks will be stored
StructureMigrationCodeUnit.Value.Parent	Static value. Use this to staticly set a DataID as parent task for all tasks synced (not required)
StructureMigrationCodeUnit.From.StatusID.Active	StatusID from source entity that should be converted to To.StatusID.Active
StructureMigrationCodeUnit.From.StatusID.Waiting	StatusID from source entity that should be converted to To.StatusID.Waiting
StructureMigrationCodeUnit.From.StatusID.Suspended	StatusID from source entity that should be converted to To.StatusID.Suspended
StructureMigrationCodeUnit.From.StatusID.Completed	StatusID from source entity that should be converted to To.StatusID.Completed
StructureMigrationCodeUnit.From.StatusID.Failed	StatusID from source entity that should be converted to To.StatusID.Failed
StructureMigrationCodeUnit.To.StatusID.Unknown	StatusID from the target entity. Given to tasks that have a status, that isn't defined in the config
StructureMigrationCodeUnit.To.StatusID.Active	StatusID from target entity that matches From.StatusID.Active

StructureMigrationCodeUnit.To.StatusID.Waiting	StatusID from target entity that matches From.StatusID.Waiting
StructureMigrationCodeUnit.To.StatusID.Suspended	StatusID from target entity that matches From.StatusID.Suspended
StructureMigrationCodeUnit.To.StatusID.Completed	StatusID from target entity that matches From.StatusID.Completed
StructureMigrationCodeUnit.To.StatusID.Failed	StatusID from target entity that matches From.StatusID.Failed

Developer info

- Type: CodeunitFormevents
- Security: Requires session
- Classpath: dk.tempusserva.gantt.StructureMigrationCodeUnit

FormEvent

Recalculate Parent

What it does

A simple codeunit that allows you to force a complete re-calculation of an associated parent record.

How to invoke

Set the codeunit Classname of the entity to:
dk.tempusserva.codeunit.common.GenberegnParent

Configuration

Set the following entity configuration

- **GenberegnParent.ParentField** Field NAME containing the DataID of the associated record (eg. ParentReference or Lookup DataID)

Developer info

- Type: CodeunitFormevent
- Security: none
- Classpath: dk.tempusserva.codeunit.common.GenberegnParent

FormEvent

LoadInfoFromCVR

What it does

A codeunit that does a lookup via danish CVR register and updates the given fields of the record.

Only the configurations that are set will be updated, allowing for fetching of specific fields.

First setup

Add a status action and set the codeunit Classname to:
dk.tempusserva.codeunit.common.LoadInfoFromCVR

Set the following entity configurations

- **LoadInfoFromCVR.LookupField** Field SYSTEMNAME containing the CVR to be looked up (required)
- **LoadInfoFromCVR.NameField** Field SYSTEMNAME for the field to be populated with the name of the company
- **LoadInfoFromCVR.AddressField** Field SYSTEMNAME for the field to be populated with the streetname and number of the company
- **LoadInfoFromCVR.PostalcodeField** Field SYSTEMNAME for the field to be populated with the zip/postal code of the company
- **LoadInfoFromCVR.CityField** Field SYSTEMNAME for the field to be populated with the cityname of the company
- **LoadInfoFromCVR.EmailField** Field SYSTEMNAME for the field to be populated with the email of the company
- **LoadInfoFromCVR.PhoneField** Field SYSTEMNAME for the field to be populated with the phonenumber of the company
- **LoadInfoFromCVR.FaxField** Field SYSTEMNAME for the field to be populated with the fax of the company
- **LoadInfoFromCVR.EmployeeCountField** Field SYSTEMNAME for the field to be populated with the employee count of the company
- **LoadInfoFromCVR.IndustryCodeField** Field SYSTEMNAME for the field to be populated with the industry code of the company

- **LoadInfoFromCVR.CompanyCodeField** Field SYSTEMNAME for the field to be populated with the company type code of the company

How to invoke

[Steps to enable codeunit]

Options

[Optional options, that can be set at runtime, eg. url-parameters]

Configuration

[All possible configurations and where to edit them]

Developer info

- Type: CodeunitFormevents
- Security: [eg. requires admin or session]
- Classpath: dk.tempusserva.codeunit.common.LoadInfoFromCVR

FormEvent

Http request on save

This Codeunit is not ready for use!

What it does

This codeunit is capable of performing an HTTP-request with parameters from the updated item and update the item based on the result.

First setup

[Optional, if extra config is required to get the codeunit working]

How to invoke

Add the codeunit `dk.tempusserva.codeunit.common.RequestFormevent` to the entity and add a configuration named `RequestFormevent.Setup` with content based on the following configuration.

It is possible to add both a "before-update" and an "after-update" action.

Configuration

Datapoint		Type	Notes
when	Required	String	Possible values: "after-update" or "before-update". Determines whether the call is executed before or after an item is updated.

status	Optional	Array	An array of the StatusID's where the call should be executed. If "after-update", then this is if the item is in one of these status's after the update, likewise for "before-update". If not given, then the call is executed every time.
call	Required		The actual configuration of the call.
call.url	Required	String	The url that is to be called. This supports input of parameters from the item.
call.method	Optional	String	The HTTP-method of the call, one of: "POST", "GET", "PUT" or "DELETE". Default: "GET".
call.type	Optional	String	The datatype of data returned, supported types: "json", "raw" or "XML". Default: "json".
call.headers	Optional	Map	A map of extra headers that should be set
call.params	Optional	Map	Path parameters that should be set and send.
call.body	Optional	String	A string that will be set and send. Not available for "GET".
call.update	Optional	Map	Fields that should be updated based on the data returned. If type is "raw", only the first item in this list will be updated, and it will be set to the entire response.

Sample

```
[
   {
      "when": "before-update",
      "status": [69],
      "call": {
```

```
        "url": "https://acme.com/resource/[ID]",
        "method": "GET",
        "type": "json",
        "headers": {
          "Authorization": "Bearer XYZ"
        },
        "params": {
          "param": "[DATA]"
        },
        "body": "",
        "update": {
          "[LINK]": "info.link",
          "[NAME]": "info.name"
        }
      }
    }
]
```

Developer info

- Type: CodeunitFormevent
- Security: requires session
- Classpath: dk.tempusserva.codeunit.common.RequestFormevent

Public Pages

Hello World

What it does

Displays "Hello World" in a seperate page. Purpose is learning and testing.

How to invoke

Make a http request:

```
.../main?command=dk.p2e.blanket.codeunit.common.PagePublicHelloWorld
```

Configuration

None

Developer info

- Type: CodeunitPagecontentPublic
- Security: Public access
- Classpath: dk.p2e.blanket.codeunit.common.PagePublicHelloWorld

PublicTokenBuilder

What it does

Usage example: Lets external users resend tokens for data they should have access to.

Builds a token for accessing records using configuration options and dynamic parameters. It requires a record that has a reference to an email, to whom the token will be sent.

Multimode operation based on supplied parameters

1. Email as parameter: Execute silently
2. No email supplied:
   - Display form with email
   - Execute normally

The DataID of the record is collected by making a lookup on the email supplied (see: LookupSolution, LookupFieldName).

An email is then sent to the user with a token for the record in question

How to invoke

Make a http request:

```
main?command=dk.p2e.blanket.codeunit.common.PagePublicTokenBuilder
```

Options

The name of the following parameters are optional and configurable

- AltToken
- EmailMatch

main?command=dk.p2e.blanket.codeunit.common.PagePublicTokenBuilder&AltToken=0

main?command=dk.p2e.blanket.codeunit.common.PagePublicTokenBuilder&EmailMatch=foo@bar.com

main?command=dk.p2e.blanket.codeunit.common.PagePublicTokenBuilder&AltToken=0&EmailMatch=foo@bar.com

Configuration

If no interface (**TokenInterface**) is specified the codeunit will allways exit without doing anything.

The codeunit has the following options

Options for the email input form

These options are required

- FormHeader: HTML before the form and resulting message
- FormFooter: HTML after the form and resulting message
- FormIntroduction: Help text displayed above the form
- FormSubmitButton: Text on submit button
- FormReponseSuccess: Message to user if email is sent succesfully
- FormReponseFailure: Message to user if email failed

Options for building the token

These options are required

- EmailMatch: Name of http parameter for setting the email
- TokenInterface: Interface to build the token for
- LookupSolution: System name of the solution which the token is built for
- LookupFieldName: Field in solution where email of the user can be found

These options are optional

- AltToken: Name of http parameter for selecting the alternative interface
- TokenInterfaceAlt: Interface used if the parameter **AltToken** is set to **1**

Options for the email message

These options are required

- EmailHeader: Subject content
- EmailContent: Body of email

Developer info

- Type: CodeunitPagecontentPublic
- Security: Public accces (with some model restrictions)
- Classpath: dk.p2e.blanket.codeunit.common.PagePublicTokenBuilder

MigrateFilesToChildren

What it does

Helps migrate from a Files: Documents field to a related entity, with a Files: Documents field in single-file mode.

Introduced in version 11879.

How to invoke

Make a http request:

```
.../main?command=com.tsnocode.codeunit.common.MigrateFilesToChildren
```

Options

Required parameters:

Name	Description
From	Entity migrating files from
FromDocument	Field to migrate files from in "From" entity
To	Entity to migrate files to
ToParent	Field that references "From" entity
ToDocument	Field to migrate files to in "To" entity

Optional parameters:

DontKeepMetadata	Don't set CreatedAt and CreatedBy based on when and who uploaded the original file
TEST	Limit to first 10 records in "From" entity

All entity and field names are system names.

Developer info

- Type: CodeunitPagecontentPublic
- Security: requires admin
- Classpath: com.tsnocode.codeunit.common.MigrateFilesToChildren

Pages

Clone System

What it does

Clones the entire webapp to another server.

Only works on linux.

First setup

1. Generate an SSH key for the tomcat-user on the source server `sudo -u tomcat ssh-keygen`
2. Add the public key to a user on the target server that has sudo-access
3. Connect from the source server, as tomcat, to the target server and accept the certificate
4. Add configurations

How to invoke

Invoke the codeunit, as an admin. eg.

`[SERVER]/main?command=common.CloneSystem`

Options

Control the behavior with the following url-parameters.

Parameter	Desctiption
with-users	Also clones the users and their groups
with-files	Also clones the files uploaded
dont-backup-target	Does not create a copy of the database on the target server with the name `_backupyyyyMMddHHmm`
dont-clean	Does not remove the sql files on both systems

Configuration

Config	Default	Description
CloneSystem.targetServer		IP or domain of target server
CloneSystem.targetUser	ec2-user	Name of the user that is being used on the target server
CloneSystem.targetBasePath	Policy:applicationBasePath	Webapps, folder on target server
CloneSystem.targetApp	Policy:applicationName	Application name on target server
CloneSystem.targetDb	DbLive-name	Name of live server on target server

Developer info

- Type: CodeunitPagecontent
- Security: Requires session
- Classpath: com.tsnocode.codeunit.common.CloneSystem

Copy Structure

What it does

Will copy all fields except files and pictures, but all sub child entities (1 level, non recursive).

How to invoke

The page is called with the command:
dk.tempusserva.codeunit.common.PageCopyStructure

Takes the following parameters

- SagID
- DataID

The user activating the function must have the group specified in "AllowCloneGroup"

Example of usage:

```
main?command=dk.tempusserva.codeunit.common.PageCopyStructure&SagID=10&DataID=100
```

Button for the function can be created using a "Action button: Button: Parameterized URL"

Example of URL pattern

```
main?command=dk.tempusserva.codeunit.common.PageCopyStructure&SagID=10&DataID=[DataID]
```

Configuration

[All possible configurations and where to edit them]

Developer info

- Type: CodeunitPagecontent
- Security: [eg. requires admin or session]
- Classpath: dk.tempusserva.codeunit.common.PageCopyStructure

Pages

Clone records

Usage

This function will copy a single record

Configuration guide

Add a button with parameterized URL using:

```
main?command=com.tsnocode.codeunit.common.CloneRecord&SagID=[SagID]&DataID=[DataID]
```

Developer info

- Type: CodeunitPagecontent
- Security: [eg. requires admin or session]
- Classpath: com.tsnocode.codeunit.common.CloneRecord

Static Content

What it does

Displays configurable static HTML code.

How to invoke

Make a http request:

```
main?command=dk.p2e.blanket.codeunit.common.PageStaticContent
```

Configuration

Set up HTML:

Designer > Modules > Static content > **StaticPageContent**

Developer info

- Type: CodeunitPagecontent
- Security: Requires session
- Classpath: dk.p2e.blanket.codeunit.common.PageStaticContent

Users By Group

What it does

Displays list of active users in the system.

Multi mode function

- User normal: No filters
- User exclusive group: Filters users/groups that active user belongs to

You can navigate through the pages

1. Lists exclusive groups + none
2. Lists users in *selected* exclusive group (1)
3. Lists groups for *selected* user (2)

How to invoke

Make a http request:

```
main?command=dk.p2e.blanket.codeunit.common.PageUsersByGroup
```

Configuration

None

Developer info

- Type: CodeunitPagecontent
- Security: Requires UderEditor or Admin
- Classpath: dk.p2e.blanket.codeunit.common.PageUsersByGroup

Users Online

What it does

Displays a list of users with a session on the current server

How to invoke

Make a http request:

```
main?command=common.PageUsersOnline
```

Configuration

None

Developer info

- Type: CodeunitPagecontent
- Security: Requires admin
- Classpath: dk.p2e.blanket.codeunit.common.PageUsersOnline

Website Iframe

What it does

Displays a configurable webpage in an IFrame.

How to invoke

Make a http request:

```
main?command=dk.p2e.blanket.codeunit.common.PageWebsiteIframe
```

Configuration

Set up IFrame options:

Designer > Modules > Static content

- WebsiteIframeURL
- WebsiteIframeWidth
- WebsiteIframeHeight

Developer info

- Type: CodeunitPagecontent
- Security: Requires session
- Classpath: dk.p2e.blanket.codeunit.common.PageWebsiteIframe

Solution Structure View

What it does

Lists the components (fields, status etc.) in the current solution.

How to invoke

Make a http request including a valid SagID:

```
main?SagID=[SagID]&command=common.SolutionStructureView
```

Configuration

None

Developer info

- Type: CodeunitPagecontent
- Security: Requires session
- Classpath: dk.p2e.blanket.codeunit.common.SolutionStructureView

Pages

Gantt Diagram

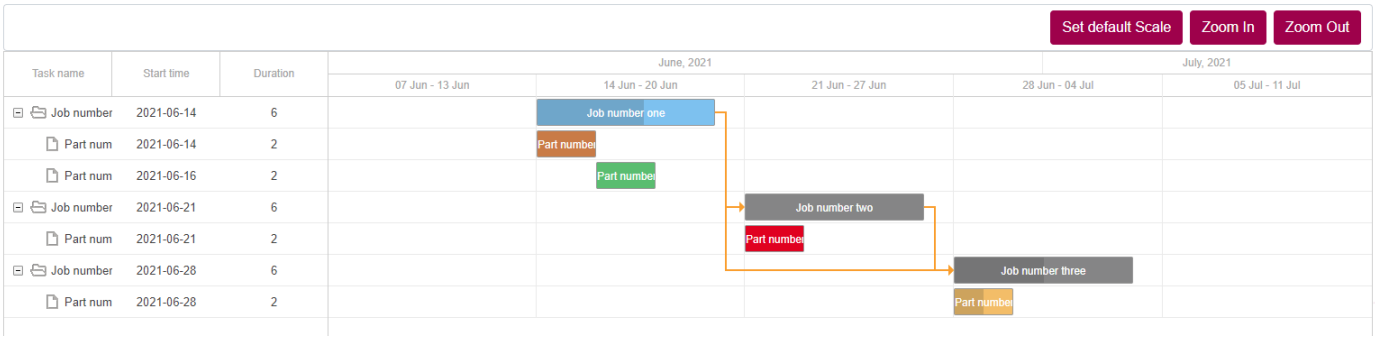
What it does

The platform has a build-in gantt chart generator.
It supports zooming, nesting and relations.

The datastructure has to be a single entity.

If your data isn't contained in a single entity use Gantt Sync CodeUnit.

Sample



How to invoke

To view the gantt chart, access:
main?command=dk.tempusserva.gantt.DhtmlXGanttPage&SagID=[SagID]

Options

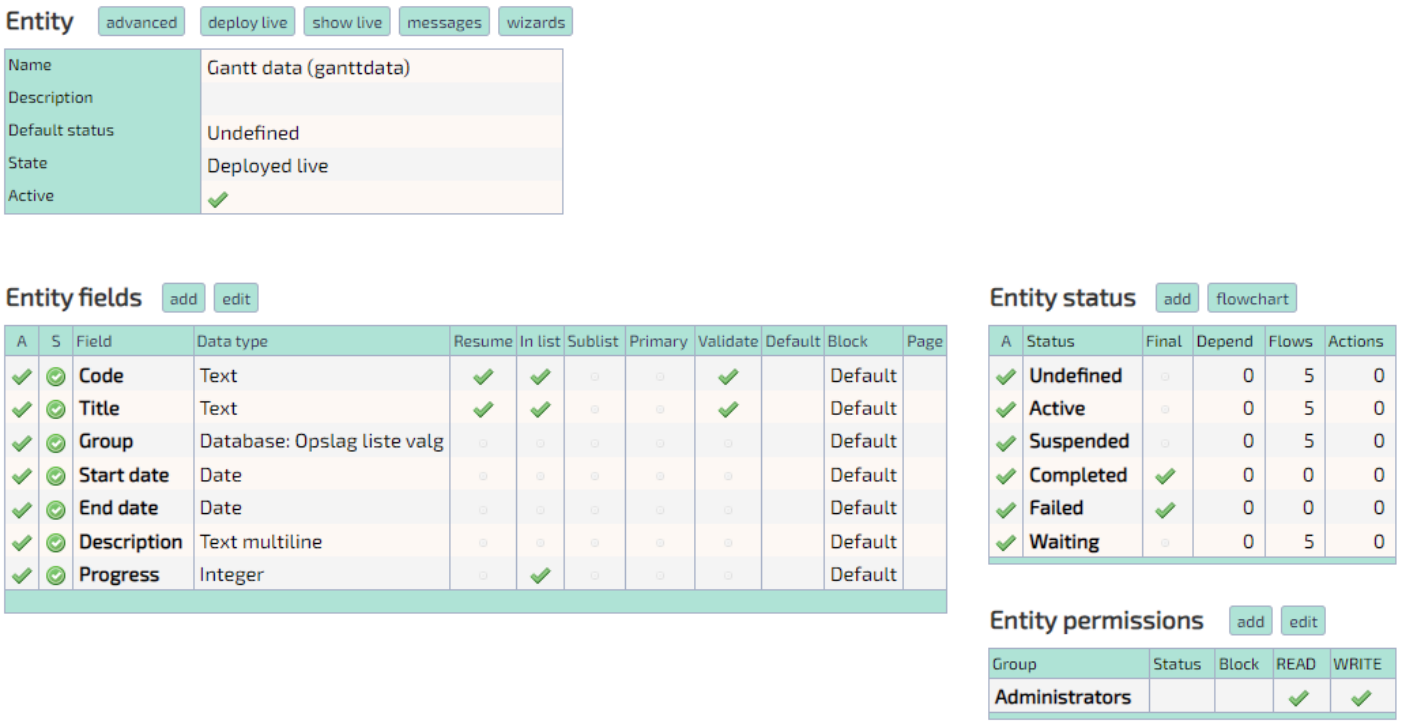
Basic setup

To view a gantt chart, first access the page main?command=dk.tempusserva.gantt.DhtmlXGanttPage to generate the required **static configurations**.
Six fields are required in the table, that the data is being read from:
Code, Title, Group, Start date, End date and Progress.

Code and Title have to be single-line text inputs.
Progress has to be an integer input, it represents how far along the task is, in percent.
Start and End date have to be dates.
Group should store the DataID of the parent-task. Here a database-lookup is used, executing query: SELECT DataID, Resume FROM data_ganttdata and displaying field Resume.

Also at least 5 statuses:
Active, Suspended, Completed, Failed and Waiting.
All other statuses will be marked as Undefined.

Sample



Entity advanced deploy live show live messages wizards

Name	Gantt data (ganttdata)
Description	
Default status	Undefined
State	Deployed live
Active	✓

Entity fields add edit

A	S	Field	Data type	Resume	In list	Sublist	Primary	Validate	Default	Block	Page
✓	✓	**Code**	Text	✓	✓			✓		Default	
✓	✓	**Title**	Text	✓	✓			✓		Default	
✓	✓	**Group**	Database: Opslag liste valg							Default	
✓	✓	**Start date**	Date							Default	
✓	✓	**End date**	Date							Default	
✓	✓	**Description**	Text multiline							Default	
✓	✓	**Progress**	Integer		✓					Default	

Entity status add flowchart

A	Status	Final	Depend	Flows	Actions
✓	**Undefined**		0	5	0
✓	**Active**		0	5	0
✓	**Suspended**		0	5	0
✓	**Completed**	✓	0	0	0
✓	**Failed**	✓	0	0	0
✓	**Waiting**		0	5	0

Entity permissions add edit

Group	Status	Block	READ	WRITE
Administrators			✓	✓

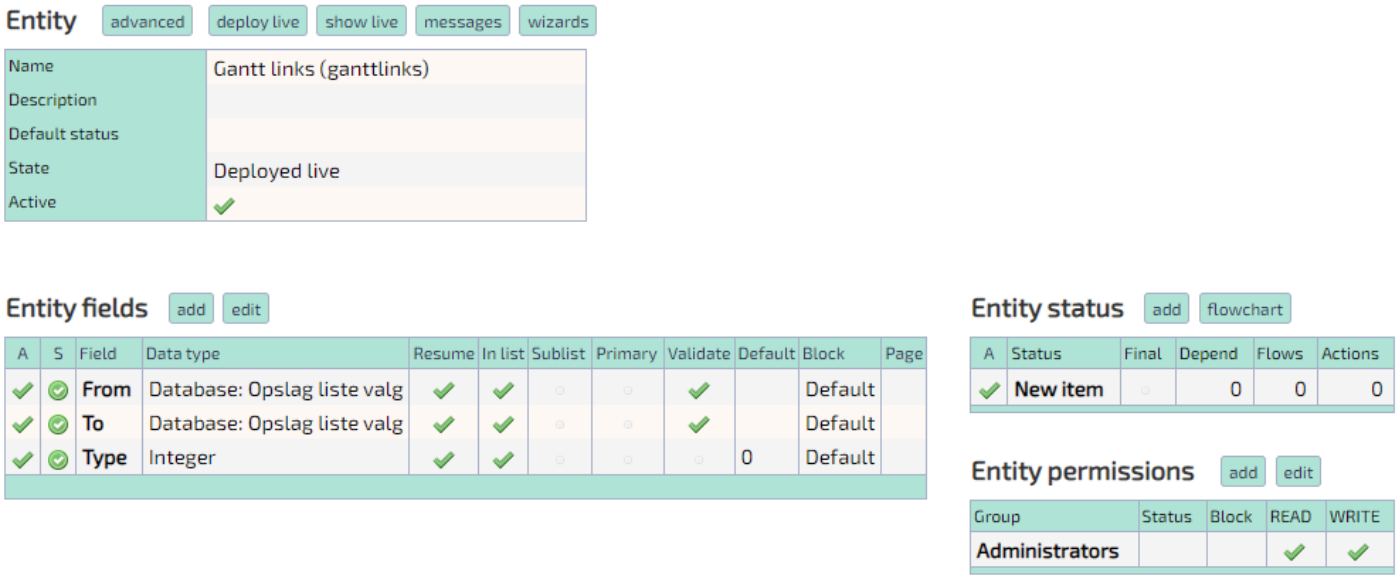
Relations

To view relations between tasks, create another table.
Two fields are required:
From and To.
A third field Type can be added, but isn't required.

From and To have to store the DataID of the tasks that should be linked via a relation. This setup is the same as Group.
Type has to have value of 1 or 0, any other value is overwritten to 0. This defines whether the relation is drawn from the beginning (1) or end (0) of the From-task.

To view the gantt chart, access:
main?command=dk.tempusserva.gantt.DhtmlXGanttPage&SagID=[SagID]&LinkSagID=[RelationSagID]

Sample



Configuration

Static config	Default value	Description
Gantt.Field.Group	GROUP_VALUE	Name of the column that contains the DataID of parent-task
Gantt.Field.StartDate	STARTDATE	Name of the column that contains the task-start date
Gantt.Field.EndDate	ENDDATE	Name of the column that contains the task-end date
Gantt.Field.Code	CODE	Name of the column that contains the task-code
Gantt.Field.Title	TITLE	Name of the column that contains the task-title
Gantt.Field.Progress	PROGRESS	Name of the column that contains the task-progress
Gantt.Filter.Field	TASKTYPE	Name of a column that contains a value that should be filtered to match Gantt.Filter.Value
Gantt.Filter.Value	249	Value that task have to equal in column Gantt.Filter.Field
Gantt.Field.Link.From	FROM	Name of column that contains DataID of source-task in a relation
Gantt.Field.Link.To	TO	Name of column that contains DataID of target-task in a relation

Gantt.Field.Link.Type		Name of column that contains type of relation, not required
Gantt.StatusID.Active	1	ID of status that marks a task as Active
Gantt.StatusID.Waiting	2	ID of status that marks a task as Waiting
Gantt.StatusID.Suspended	3	ID of status that marks a task as Suspended
Gantt.StatusID.Completed	4	ID of status that marks a task as Completed
Gantt.StatusID.Failed	5	ID of status that marks a task as Failed
Gantt.StatusColor.Undefined	#858586	CSS-color of tasks with status Undefined
Gantt.StatusColor.Active	#76C0F1	CSS-color of tasks with status Active
Gantt.StatusColor.Waiting	#F6BF5F	CSS-color of tasks with status Waiting
Gantt.StatusColor.Suspended	#F28F42	CSS-color of tasks with status Suspended
Gantt.StatusColor.Completed	#4DBE6C	CSS-color of tasks with status Completed
Gantt.StatusColor.Failed	#E70015	CSS-color of tasks with status Failed

Advanced usage

It's possible to override the static content.
To do so, add the desired value to the url.
Example:
To override the field used for Title, add *&Gantt.Field.Title=[field-name]* to the url

Exporting

The chart supports exporting to Excel and iCal formats.

It's possible to overwrite the export, be declaring a javascript function named `customExportToExcel` or `customExportToICal`. Either takes one argument: the gantt object.

Heres an example that removes html markup from the gantt titles before export.

```
function customExportToExcel(gantt) {
  var data = gantt.getDatastore("task").getVisibleItems();
  var rtn = [];
  for (var i = 0; i < data.length; i++) {
    var d = new Date(data[i].start_date);
    rtn[i] = {
      color: data[i].color,
      duration: data[i].duration,
      id: data[i].id,
      open: data[i].open,
      parent: (data[i].parent == "0" || data[i].parent == 0 ? "" : data[i].parent),
      progress: data[i].progress,
      start_date: d.getFullYear() + "-" + ("0"+(d.getMonth()+1)).slice(-2) + "-" + ("0" + d.getDate()).slice(-2) + " "
+ ("0" + d.getHours()).slice(-2) + ":" + ("0" + d.getMinutes()).slice(-2),
      text: $(data[i].text).text(),
    };
  }
  console.log(rtn);
  viewGantt({data: rtn, links: []});
  setTimeout(function() {
    gantt.exportToExcel();
  }, 5);
  setTimeout(loadGantt, 100);
}
```

Developer info

- Type: CodeunitPagecontent
- Security: Requires session
- Classpath: dk.tempusserva.gantt.DhtmlXGanttPage

Move a File

What it does

Moves a file from one document-field to another, on the same entity.

Works between document-fields with and without signing.

How to invoke

Add the following snippet to the entity scripts. This enables movement between all document-fields on the entity.

It adds another column to the file-lists, with a "Move" link. When clicked it opens a popup, with a dropdown where the new field can be selected.

```
const moveFile = (id, to) => {
  const params = new URLSearchParams(window.location.search)
  $.ajax({
    url: "main",
    method: "GET",
    data: {
      "command": "dk.tempusserva.codeunit.common.MoveFile",
      "SagID": params.get("SagID"),
      "DataID": params.get("DataID"),
      "FileID": id.split("_")[2],
      "FieldFrom": id.split("_")[1],
      "FieldTo": to,
    },
    success: (data) => {
      if (data == "File moved") {
        $(`#${id}`).find("td").last().html("Moved")
      } else {
```

```
        alert(data)
      }
    },
  })
}

$(() => {
  const moveFileAction = () => {
    console.log("yay")
    moveFile($("#modeFileFrom").val(), $("#moveFileTo").val())
    $("#moveFilePopup").remove()
  }
  $(() => {
    $(".tableFiles").each((i,e) => {
      let t = $(e)
      t.find("thead").find("tr").each((i,e) => { $(e).append("<th></th>") })
      t.find("tbody").find("tr").each((i,e) => { $(e).append(`<td><a href="#"
class="moveFile">Move</a></td>`) })
    })
    $(".moveFile").on("click", (e) => {
      e.preventDefault()
      let t = $(e.currentTarget)
      let id = t.parent().parent().attr("id")
      let r = `<select class="form-control" id="moveFileTo">`
      r += `<option value="">Select a field</option>`;
      $(".uploadFiles").each((i,e) => {
        let p = $(e).parent().attr("id").replace("VB_DATA_", "")
        if (id.split("_")[1] != p) {
          let x = $(e).parent().parent().parent().find(`#NB_DATA_${p}`).html()
          r += `<option value="${p}">${x}</option>`
        }
      })
      r += "</select>"
      $("#TempusServaPage").append(`
        <div id="moveFilePopup" style="display: none;">
          <label>Move file to:</label>
          ${r}
          <input type="hidden" id="modeFileFrom" value="${id}"/>
          <a href="javascript:moveFileAction();" class="moveFileAction">Move</a>
```

```
        </div>
      `)
      createJqueryDialog("moveFilePopup")
    })
  })
})
```

Configuration

None

Developer info

- Type: CodeunitPagecontent (raw)
- Security: Requires session
- Classpath: dk.tempusserva.codeunit.common.MoveFile

Convert database to UTF-8

What it does

Special characters causing errors

The default support for UTF8 is 3 byte.

Using 4 byte UTF content in text areas, can cause errors, that looks like

```
java.sql.SQLException: Incorrect string value: '\xC2\x96 Z. ...' for column 'NOTES' at row 1
```

How to invoke

Update the database to full UTF8 support using the following command

```
main?command=dk.p2e.blanket.codeunit.common.PageConvertDatabaseToUTF8
```

Note an administration profile is needed to complete the operation.

You might also have to update the tomcat server.xml file.
You have to add `useBodyEncodingForURI="true"` to the connectors.

Note that this codeunit has build in protection against being excuted multiple times at once, and should be executed every time a new field or entity has been added to the database.

Issue before version 11315 on MariaDB

Due to a difference in how MySQL and MariaDB stores "DEFALUT NULL", when executing this codeunit on a MariaDB server, all columns with the default value of null would be updated to have a default value of 'NULL' (the string NULL, not the empty value null).

To find the affected columns and generate all the required sql to fix the issue, run the following query.
Remember to reaplce `[LIVE DB]` with the name of the live database of your instance!

```
SELECT a.TABLE_NAME, a.COLUMN_NAME, CONCAT('ALTER TABLE ', a.TABLE_NAME, ' ALTER ', a.COLUMN_NAME,
' SET DEFAULT NULL;') AS upd
FROM information_schema.columns AS a
INNER JOIN INFORMATION_SCHEMA.TABLES b ON
a.TABLE_CATALOG = b.TABLE_CATALOG AND
a.TABLE_SCHEMA = b.TABLE_SCHEMA AND
a.TABLE_NAME = b.TABLE_NAME AND
LOWER(b.table_type) != 'view'
WHERE a.COLUMN_DEFAULT = '\'NULL\'' AND
a.TABLE_SCHEMA = '[LIVE DB]'
```

This should create one row for every column in the database that has a default value of the string 'NULL'.

Take the output of the third column, copy all rows and execute them against the database. Should look something like this:

```
ALTER TABLE form ALTER SagNavnI18N SET DEFAULT NULL;
ALTER TABLE form ALTER EntityIcon SET DEFAULT NULL;
ALTER TABLE form ALTER Description SET DEFAULT NULL;
ALTER TABLE form ALTER DescriptionI18N SET DEFAULT NULL;
ALTER TABLE form ALTER MailDomains SET DEFAULT NULL;
ALTER TABLE form ALTER MailSecurity SET DEFAULT NULL;
ALTER TABLE form ALTER Codeunit SET DEFAULT NULL;
ALTER TABLE form ALTER Instructions SET DEFAULT NULL;
```

Go back and execute the first query, it now should return 0 rows.

Options

None.

Configuration

None.

Developer info

- Type: CodeunitPagecontent
- Security: Requires administrator
- Classpath: dk.p2e.blanket.codeunit.common.PageConvertDatabaseToUTF8

Pages

Bulk importing files

What it does

This page describes how to setup the bulk-import codeunit, to import a lot of files at once and generate a separate record in an entity, for each file.

First setup

First, setup an entity with at least two fields: A text-field for the filename and a document field for the document.
Second, create or edit the following 5 "Static content".

1. BulkImport.SagID, the SagID of the entity that is the target of the import
2. BulkImport.UserGroupID, The GroupID of the usergroup that is allowed to access the bulk-import system
3. BulkImport.FileTitleField, The systemname of the field where the name of the file should be stored
4. BulkImport.FileFieldID, The FieldID of the field where the uploaded document should be stored
5. BulkImport.StatusID, The StatusID that uploaded documents should have

Third, edit the url on the site and access
main?command=dk.tempusserva.codeunit.bulkimport.ImportPage
Fourth, upload the files.

Next step could be to index all the files, allowing the content of the files to be searched and queried, maybe through ChatGPT.

How to invoke

[Steps to enable codeunit]

Options

[Optional options, that can be set at runtime, eg. url-parameters]

Configuration

[All possible configurations and where to edit them]

Developer info

- Type: CodeunitPagecontent
- Security: [eg. requires admin or session]
- Classpath: dk.tempusserva.codeunit.bulkimport.ImportPage

Update resume for entire entity

What it does

Updates the resume for all records in a given entity

How to invoke

The page is called with the command:
com.tsnocode.codeunit.backend.UpdateEntityResumes

Requires the following parameter

- SagID

Example of usage:

```
main?command=com.tsnocode.codeunit.backend.UpdateEntityResumes&SagID=10&RecordLimit=10000
```

Options

Takes the following parameters

- RecordLimit (default 1000)
- BatchSize (default 100)

Developer info

- Type: CodeunitPagecontent

- Security: requires admin
- Classpath: com.tsnocode.codeunit.backend.UpdateEntityResumes

Cleanup Inactive Model Parts

What it does

Removes all states in an entity, that have no records and are marked as inaktive.

How to invoke

Call the codeunit, with the SagID to clean.

```
main?command=com.tsnocode.codeunit.backend.CleanupInactiveModelParts&SagID=[SagID]
```

Developer info

- Type: CodeunitPagecontent
- Security: administrator
- Classpath: com.tsnocode.codeunit.backend.CleanupInactiveModelParts

Fix Docx Tags

What it does

Tries to fix templating tags in a docx template.

How to invoke

Call the codeunit, along with a TemplateID

```
main?command=com.tsnocode.codeunit.backend.FixDocxTags&TemplateID=[TemplateID]
```

Developer info

- Type: CodeunitPagecontent
- Security: administrator
- Classpath: com.tsnocode.codeunit.backend.FixDocxTags

Pages

Reinvite Users

What it does

Resets passwords and sends out welcome messages, to all of the listed emails.

First setup

Setup the configuration "ReinviteUsersGroupID".

How to invoke

Call the page

```
main?command=com.tsnocode.codeunit.backend.ReinviteUsers
```

Input the emails that should be reset and re-invited, one email pr line, and submit the form.

Configuration

ReinviteUsersGroupID

Developer info

- Type: CodeunitPagecontent
- Security: administrator and specific group
- Classpath: com.tsnocode.codeunit.backend.ReinviteUsers

Pages

Generate Video Thumbnails

What it does

Generates missing thumbnails for files in an entity.

First setup

Make sure the application is using filesystem storage (filesystemStorageActive)

How to invoke

Call the page

```
main?command=com.tsnocode.codeunit.common.GenerateVideoThumbnails&SagID=[SagID]
```

Configuration

You can adjust the thumbnail size via the policy uploadPictureSizeThumbnail

Developer info

- Type: CodeunitPagecontent
- Security: administrator
- Classpath: com.tsnocode.codeunit.common.GenerateVideoThumbnails

Export Template To WebDav

What it does

Exports a template, saves it to the record, and opens it via webdav.

First setup

Add an action button with the correct parameters

How to invoke

Call the page, e.g. via an action button

```
main?command=com.tsnocode.codeunit.common.SaveTemplateToWebDav&SagID=[SagID]&DataID=[DataID]&TemplateID=[TemplateID]&Filename=[Filename]&Field=[Field]
```

Options

- SagID
- DataID
- TemplateID
- Filename, Name of the generated file
- Field, Field systemname to save the generated file to

Developer info

- Type: CodeunitPagecontent

- Security: session
- Classpath: com.tsnocode.codeunit.common.SaveTemplateToWebDav

Pages

Bulk import files

What it does

Imports files and creates a record for each of them.

First setup

[Optional, if extra config is required to get the codeunit working]

How to invoke

Open the page

```
main?command=com.tsnocode.codeunit.system.bulkimport.ImportPage
```

Options

[Optional options, that can be set at runtime, eg. url-parameters]

Configuration

[All possible configurations and where to edit them]

Developer info

- Type: CodeunitPagecontent
- Security: admin, datahandler, or specific group
- Classpath: com.tsnocode.codeunit.system.bulkimport.ImportPage

Dynamically change users exclusive group

What it does

Gives the user a list of exclusive groups defined in the application. The user can select one and will have their exclusive group set to that value untill they sign out.

Also supports changing the users exclusive group, if the already have one.

Does not support advanced security with multiple exclusive groups.

First setup

Enable the policy securityExclusiveGroupSet.

If the option to change groups is required, enable the policy securityExclusiveGroupChange.

How to invoke

Add a button that links to this codeunit

```
main?command=dk.p2e.blanket.codeunit.common.PageSetExclusiveGroup
```

Options

None

Configuration

None

Developer info

- Type: CodeunitPagecontent
- Security: Session
- Classpath: dk.p2e.blanket.codeunit.common.PageSetExclusiveGroup

Is Date weekend/holiday

What it does

Tests wether a date is a weekend date and/or a holiday (based on those that are input via the designer).

Available from version 11968.

First setup

Just make sure that you maintain the list of holidays (in the designer).

How to invoke

Make an http request to

```
main?command=com.tsnocode.codeunit.frontend.TestDate&Date=[DATE]
```

This will return a json object.

```
{
  "date": "2026-05-14",
  "isHoliday": true,
  "isWeekend": false,
  "error": false,
  "holidayName": "Kristihimmelfart"
}
```

If `error` is true, then a message (`msg`) will also be available.
If `isHoliday` is true, then a `holidayName` should also be returned. (As of version 12007)

Developer info

- Type: [Type+link]
- Security: [eg. requires admin or session]
- Classpath: [full class path]