FormEvent

- BuildUserProfileFromRecord
- Gantt Sync CodeUnit
- Recalculate Parent
- LoadInfoFromCVR
- Http request on save

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

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

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

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

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