Workflow Action and Notification Customization

- Using the LEK_workflow_action table
    - Meanings of the fields in the action table
- External Workflow Trigger
    - Example:
    - Calling the workflow trigger via HTTP
    - Remote callback when all users finish there jobs
- Workflow Trigger in the application
- Workflow Actions and Notifications
    - Workflow Mail parameters
- Default activated Actions
- Daily and periodical called (Cronjob) Actions

One of the ideas behind translate5 workflows is, that specific actions in the application are triggered automatically by a well defined list of triggering conditions.

The mapping between actions and triggers is predefined but can be set by the admin of a translate5 installation. Currently there is no GUI for that, so the settings has to be done directly in the database in the LEK_workflow_action table.

Using the LEK_workflow_action table

Example of inserting a new workflow action

```
INSERT INTO `LEK_workflow_action` (`workflow`,`trigger`,`inStep`,`byRole`,`userState`,`actionClass`,`action`,
`parameters`,`position`,`description`)
VALUES ('default', 'doCronDaily', null, null, null, 'editor_Workflow_Actions', 'finishOverduedTasks', null,
0,'');
```

Example Content in the LEK_workflow_action table

```
MariaDB [icorrectT5]> select * from LEK_workflow_action;
+----+----------+---------------------------------------------------+-----------+--------+-----------
+-----------------------------+---------------------------+------------+----------+-------------+
| id | workflow | trigger                                           | inStep    | byRole | userState |
actionClass                 | action                    | parameters | position | description  |
+----+----------+---------------------------------------------------+-----------+--------+-----------
+-----------------------------+---------------------------+------------+----------+-------------+
|  1 | default  | handleAllFinishOfARole                            | reviewing | reviewer | finished  |
editor_Workflow_Actions      | segmentsSetUntouchedState |            |        0 |             |
|  2 | default  | handleAllFinishOfARole                            | reviewing | reviewer | finished  |
editor_Workflow_Actions      | taskSetRealDeliveryDate   |            |        1 |             |
|  3 | default  | handleAllFinishOfARole                            | NULL      | NULL   | finished  |
editor_Workflow_Notification | notifyAllFinishOfARole    |            |        2 |             |
|  4 | default  | handleUnfinish                                    | NULL      | reviewer | NULL      |
editor_Workflow_Actions      | segmentsSetInitialState   |            |        0 |             |
|  5 | default  | handleBeforeImport                                | NULL      | NULL   | NULL      |
editor_Workflow_Actions      | autoAssociateTaskPm       |            |        0 |             |
|  6 | default  | handleImport                                      | NULL      | NULL   | NULL      |
editor_Workflow_Notification | notifyNewTaskForPm        |            |        0 |             |
|  7 | default  | handleImport                                      | NULL      | NULL   | NULL      |
editor_Workflow_Actions      | autoAssociateEditorUsers  |            |        1 |             |
|  8 | default  | handleUserAssociationAdded                        | NULL      | NULL   | NULL      |
editor_Workflow_Notification | notifyNewTaskAssigned     |            |        0 |             |
| 45 | default  | handleDirect::notifyAllUsersAboutTaskAssociation | NULL      | NULL   | NULL      |
editor_Workflow_Notification | notifyAllAssociatedUsers  |            |        0 |             |
| 51 | default  | doCronDaily                                       | NULL      | NULL   | NULL      |
editor_Workflow_Actions      | deleteOldEndedTasks       | NULL       |        0 |             |
+----+----------+---------------------------------------------------+-----------+--------+-----------
+-----------------------------+---------------------------+------------+----------+-------------+
```

Meanings of the fields in the action table

An action is either just a function (called action) which does something or a notification which sends a notification (by default an email) to some users.

For examples of usage see the classe: editor_Workflow_Notification and editor_Workflow_Actions.

Field	Description / idea behind
id	DB auto incremented ID
workflow	defines to which workflow the action belongs. So only the actions for the workflow configured in the task are triggered. Currently the class inheritance hierarchy is reflected here. So for a Workflow class "foo" extending "default" all actions with both values are considered. **This behavior will change in the future** to enable the possibilty to disable extended default actions by child classes.
trigger	The named event trigger to react on. See list of available triggers below.
inStep	In step filter: the action is executed only if the tasks workflow is in the configured step. NULL means no filter.
byRole	By role filter: the action is executed only if the initiator of the trigger is in the configured role. NULL means no filter.
userState	userState filter: the action is executed only if the initiator of the trigger has the configured job status. NULL means no filter.
actionClass	the class where the to be called action is contained. Must inherit from "editor_Workflow_Actions_Abstract" class.
action	the action to be called
parameter	optional additional parameters, which are passed to the executed action. For example Mail notifications can be configured. See details below!
position	Order of execution for entries with the same trigger configuration.
description	Contains a human readable description for the row

External Workflow Trigger

The TaskController provides a callable action to trigger workflow events via URL - if configured in the workflow action table.

The value for the field trigger must contain the prefix: "handleDirect::" the rest of the string is the name of the trigger passed as POST parameter.

See also Task REST Api and the examples.

Example:

By default the notifyAllAssociatedUsers action is configured:

DirectTrigger configuration

```
| 45 | default  | handleDirect::notifyAllUsersAboutTaskAssociation | NULL      | NULL   | NULL      |
editor_Workflow_Notification | notifyAllAssociatedUsers  |            |        0 |
```

To trigger that action the following POST request must be done. This can be either done via frontend or via external API call. The only precondition is, that the calling user is a PM user.

Calling the workflow trigger via HTTP

What	Value
Method	POST
URL	"/editor/task/ID/workflow" where the ID is the tasks DB id.
Parameter "trigger"	the trigger to be triggered, in the example above "notifyAllUsersAboutTaskAssociation"

Remote callback when all users finish there jobs

In the example bellow it is shown how workflow trigger with remote callback URL can be configured after all users did finish there jobs in a task. The parameters field must be json string and all invalid content there will be ignored.

Remote callback configuration

```
# id, workflow, trigger, inStep, byRole, userState, actionClass, action, parameters, position, description
'37', 'default', 'handleAllFinishOfARole', NULL, NULL, NULL, 'editor_Workflow_Actions',
'triggerCallbackAction', '{"url":"http://mytestroute.de/test","params":{"username":"myusername","password":"
mypass"        }}', '0', 'Send a request to the configured url parameter with the task and task user assoc
data. If "params" field is provided in the parameters field, this will be applied to in the request json.'
```

Example for parameters configuration:

Parameters field example

```
{
"url":"http://mytestroute.de/test",
"params":{
        "username":"myusername",
        "password":"mypass"
        }
}
```

url url which will be called by the trigger

params additional key-value parameters which will be submitted as row post data. This can be used for authentication for example.

Alongside with the custom "**params"** also the task data and the task-user association will be present in the submitted post data.

task translate5 task info

tua all associated users to the provided task

Post request content example

```
{
    "username": "myusername",
    "password": "mypass",
    "task": {
        "id": 1919,
        "entityVersion": "124",
        "modified": "2022-06-08 11:35:10",
        "taskGuid": "{876c2173-58aa-4220-865b-8454a5470fd6}",
        "taskNr": "",
        "foreignId": "",
        "taskName": "Task-de-en.html - de \/ en",
        "foreignName": "",
        "sourceLang": "4",
        "targetLang": "5",
        "relaisLang": "0",
        "locked": null,
        "lockingUser": null,
        "state": "open",
        "workflow": "default",
        "workflowStep": "14",
        "workflowStepName": "reviewing",
        "pmGuid": "{e6828cdf-2ee0-4a25-af0a-92e6f060e9eb}",
        "pmName": "Manager, Project (manager)",
        "wordCount": "11",
        "userCount": "1",
        "orderdate": "2022-06-07 00:00:00",
        "enddate": null,
        "referenceFiles": "0",
        "terminologie": "0",
        "enableSourceEditing": "0",
        "edit100PercentMatch": "0",
        "lockLocked": "1",
        "qmSubsegmentFlags": "{\"qmSubsegmentFlags\":[{\"text\":\"Accuracy\",\"id\":1,\"children\":[{\"text\":\"
Mistranslation\",\"id\":2,\"children\":[{\"text\":\"Terminology\",\"id\":3}]},{\"text\":\"Omission\",\"id\":4},
{\"text\":\"Addition\",\"id\":5},{\"text\":\"Untranslated\",\"id\":6}]},{\"text\":\"Fluency\",\"id\":7,\"
children\":[{\"text\":\"Register\",\"id\":8},{\"text\":\"Style\",\"id\":9},{\"text\":\"Inconsistency\",\"id\":
10},{\"text\":\"Spelling\",\"id\":11},{\"text\":\"Typography\",\"id\":12},{\"text\":\"Grammar\",\"id\":13},{\"
text\":\"Locale violation\",\"id\":14},{\"text\":\"Unintelligible\",\"id\":15}]},{\"text\":\"Verity\",\"id\":16,
\"children\":[{\"text\":\"Completeness\",\"id\":17},{\"text\":\"Legal requirements\",\"id\":18},{\"text\":\"
Locale applicability\",\"id\":19}]}],\"severities\":{\"critical\":\"Critical\",\"major\":\"Major\",\"minor\":\"
Minor\"}}",
        "emptyTargets": "0",
        "importAppVersion": "development",
        "customerId": "1",
        "usageMode": "simultaneous",
        "segmentCount": "3",
        "segmentFinishCount": "3",
        "taskType": "projectTask",
        "projectId": "1918",
        "diffExportUsable": "0",
        "description": "",
        "created": "2022-06-07 14:23:02"
    },
    "tua": {
        "id": "1492",
        "taskGuid": "{876c2173-58aa-4220-865b-8454a5470fd6}",
        "userGuid": "{e6828cdf-2ee0-4a25-af0a-92e6f060e9eb}",
        "state": "finished",
        "role": "reviewer",
        "workflowStepName": "reviewing",
        "workflow": "default",
        "segmentrange": "",
        "usedState": null,
        "isPmOverride": "0",
        "deadlineDate": "2022-06-10 19:55:12",
        "assignmentDate": "2022-06-07 14:24:07",
        "finishedDate": "2022-06-07 17:12:29",
        "trackchangesShow": "1",
        "trackchangesShowAll": "1",
        "trackchangesAcceptReject": "1"
    }
}
```

Workflow Trigger in the application

- TODO List all in the core available workflow trigger
- TODO Explain that more Actions can be added in the core code

Workflow Actions and Notifications

- TODO List all in the core available workflow actions and notifications and explain them (from editor_Workflow_Actions)
- TODO Explain that more Actions can be added by custom Plugins or in the core code

Workflow Mail parameters

Each notification which sends an e-mail can be configured with CC and BCC receivers. They have to be added into the parameters field as JSON.

The JSON may look like:

```
{
  "cc": {
    "*": ["visiting"],
    "reviewing": ["translation", "reviewing"]
  },
  "bcc": {
    "byUserLogin": ["testlector", "testapiuser"]
  }
}
```

The structure under CC and BCC is basically equal, it is always a string key, pointing to an array.

Basically the key is either the target workflow step name, * for matching all workflow step names, or the special keys "byUserLogin" to provide directly some user logins as additional mail receivers.

The array is containing either the steps which should receive the e-mail, or in the case of byUserLogin just a user login of single users.

Default activated Actions

- TODO To be done

Daily and periodical called (Cronjob) Actions

How to set up cron jobs for translate5 please look in the installation manual of translate5.

In order to use the periodically triggered actions (cronjobs) via URL the following configuration has to be done:

- Setup a cronjob or scheduled task which calls the URL https://YOURINSTALLATION/editor/cron/daily/ (under linux for example with wget)
- The IP Adress of the caller must configured in the Zf_configuration table for security reasons
    - Set the caller IP Adress in the configuration runtimeOptions.cronIP

Alternatively the Cron Jobs can be triggered via CLI command: "t5 cron" triggers the periodical entries, "t5 cron --daily" triggers the daily actions.

After setting up the cron job, the workflow trigger "doCronDaily" is fired on each call of the above URL / CLI command call.

Useful actions for this trigger are:

trigger	actionClass	action	parameters	description
doCronDaily	editor_Workflow_Actions	finishOverduedTaskUserAssoc	/	Checks the deadine dates of a task assoc, if it is overdued, it'll be finished for all lectors, triggers normal workflow handlers if needed.
doCronPeriodical	editor_Workflow_Actions	deleteOldEndedTasks	Optional: {"limit": 5, "workflowSteps": ["no workflow", "workflowEnded"]}	Delete all tasks where the task status is 'end', and the last modified date for this task is older than x days (where x is Zf_configuration property taskLifetimeDays) Only X tasks are deleted at once. X defaults to 5 and can be set as parameter. Additionally "workflowSteps" can be provided. In this case except from default **end** status of a task also tasks in one of workflow steps will be handled.

doCronPeriodical	editor_Workflow_Actions	deleteOldEndedTasks	{ "filesystem": "local", "targetPath": "/backup/task-{id}-{taskNr}-{taskName}.zip", "limit": 5 (optional), "workflowSteps": ["no workflow", "workflowEnded"] (optional) }	Same as above deleteOldEndedTasks BUT makes an export (original format and XLF2 + task metadata as JSON file) to the location given in targetPath. Basically all values given in {curly braces} are replaced with the same named value from the task json, so the naming of the zip files can be controlled.
				
				SFTP is implemented as target too, example:
				
				`{`
				`    "filesystem": "sftp",`
				`    "targetPath": "/backup/task-{id}-{taskNr}-{taskName}.zip",`
				`    "host": "HOSTNAME_OR_IP",`
				`        "port": 1234,`
				`    "username": "USERNAME",`
				`    "password": "PASSWORD"`
				`}`
				
				Apart from the local filesystem and SFTP, WebDAV and several cloud services can be addressed. Get in contact with MittagQI to get more information here.
				
				Port is optional, must be given as integer.
				
				targetPath must contain the whole absolute path on the server. You get that after logging in with an ordinary sftp tool to your sftp server.
				
				Only X tasks are deleted/backuped at once. X defaults to 5 and can be set as optional parameter limit in the config JSON.

doCronDaily	editor_Workflow_Notification	notifyOverdueDeadline	{"receiverUser":"aleksandar10","daysOffset": 2,"template":"notifyOverdueTasks_MasterPM"}	Notify the users of a task when the delivery date is over the defined days in the parameters config daysOffset. Available config fields: receiverUser: the user login to which all of the reminders will be send. If not configured, email reminder to the actual task-associated user daysOffset: how many days after the deadline date a reminder email will be send template: template used for the reminder email
doCronDaily	notifyDeadlineApproaching	notifyDeadlineApproaching	{"receiverUser":"aleksandar10","daysOffset": 2,"template":"notifyOverdueTasks_MasterPM"}	Notify the the associated users when the deadlineDate is approaching. receiverUser: the user login to which all of the reminders will be send. If not configured, email reminder to the actual task-associated user daysOffset: how many days before the deadline date a reminder email will be send template: template used for the reminder email