

Adobe CQ Help / Example Implementation of a Workflow Process Step

Problem

You search for an example of a Workflow Process Step Implementation.

Resolution

A Workflow Process Step implementation in Java is a Class which implements the Interface `JavaProcessExt` (or `JavaProcess` if no external attributes are used).

This Class must be a OSGI Service and implement the method `execute(WorkItem item, WorkflowSession session, String[] args)`. In this method the provided objects `item` (`WorkItem`) and `session` (`WorkflowSession`) can be used to implement the business logic of the process step.

As soon as this Service is available in the system (for example when you include it in a bundle and install it) you can select it in the Workflow GUI (workflow model - process step - Implementation)...

Examples

Note: The following examples are just very simple examples not to be used in production as they are.

You have to include the example Class into one of your bundles and mark the package `com.day.cq.examples.wf` external to make the service available in the system...

Simple Workflow Process Step Example:

The following Workflow Process Step just checks the available routes to the next step(s) and process the route marked as default or the first route if no route was marked as default and writes some logging.

```
package com.day.cq.examples.wf;

import com.day.cq.workflow.WorkflowException;
import com.day.cq.workflow.WorkflowSession;
import com.day.cq.workflow.exec.JavaProcessExt;
import com.day.cq.workflow.exec.Route;
import com.day.cq.workflow.exec.WorkItem;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import java.util.List;
import java.util.Arrays;

/**
 * <code>WorkflowProcessStep</code> simply shows how to advance to the next step
 *
 * @scr.component metatype="false"
 * @scr.service
 */
public class WorkflowProcessStep implements JavaProcessExt {

    /** Default log. */
    protected final Logger log = LoggerFactory.getLogger(WorkflowProcessStep.class);

    public void execute(WorkItem item, WorkflowSession session)
        throws Exception {
        try {
            log.info("executing: " + session.toString());
            boolean advanced = false;
            List<Route> routes = session.getRoutes(item);
            for (Route route : routes) {
                if (route.hasDefault()) {
                    String fromTitle = item.getNode().getTitle();
                    String toTitle = route.getDestinations().get(0).getTo().getTitle();
                    session.complete(item, route);
                    log.info(item.getId() + " advanced from " + fromTitle + " to " + toTitle);
                    advanced = true;
                }
            }
            // fallback if no route was marked as default
            if (!advanced) {
                session.complete(item, routes.get(0));
                String fromTitle = item.getNode().getTitle();
            }
        }
    }
}
```

```

        String fromTitle = item.getWorkflowData().getTitle();
        String toTitle = routes.get(0).getDestinations().get(0).getTo().getTitle();
        log.info(item.getId() + " advanced from " + fromTitle + " to " + toTitle);
    }
} catch (WorkflowException e) {
    log.error("Could not advance workflow.", e);
}
}

public void execute(WorkItem item, WorkflowSession session, String args[]) throws Exception {
    log.info("Process Arguments" + Arrays.toString(args));
    execute(item, session);
}
}

```

Translation Workflow Process Step Example:

With this it's possible to define parallel XOR steps for the translation job for every language. Every step can belong to another user or group.

As `Process Arguments` you can set the default language (for example de) and the language hierarchy level (default is 3)

Note: As the Workflow GUI is not able to display more than two XOR (or AND) steps you have to manage workflows with more parallel XOR steps (so more than two languages) directly in CRX, please see [How to add more than two "AND" Steps to a Workflow](#).

Attached you will find an example translation workflow ([ExampleTranslationWorkflow-1.1-2.zip](#)) with seven parallel XOR steps for en, de, fr, es, it, ja and zh. Please note that the user for all steps is `admin`, so this has to be adapted with the corresponding users or groups the translations should be executed for the corresponding language...

```

package com.day.cq.examples.wf;

import com.day.cq.workflow.WorkflowException;
import com.day.cq.workflow.WorkflowSession;

import com.day.cq.workflow.exec.Route;
import com.day.cq.workflow.exec.WorkItem;
import com.day.cq.workflow.exec.JavaProcessExt;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import java.util.List;

/**
 * <code>TranslationProcessStep</code> checks the language label the current page is rendered in
 * and selects the corresponding route to select the correct translation step
 *
 * @scr.component metatype="false"
 * @scr.service
 */
public class TranslationProcessStep implements JavaProcessExt {

    /**
     * Default log.
     */
    protected final Logger log = LoggerFactory.getLogger(TranslationProcessStep.class);

    private String defaultLanguageLabel = "en";
    private static int languageHierarchyLevel = 3;

    public void execute(WorkItem item, WorkflowSession session) throws Exception {
        try {
            // get language label of current page
            String handle = item.getWorkflowData().getPayload().toString();
            String languageLabel = (handle != null && handle.split("/").length > languageHierarchyLevel) ? handle.split("/")[languageHierarchyLevel] : defaultLanguageLabel;
            log.info("executing for handle '" + handle + "' and actual language '" + languageLabel + "'");
            boolean advanced = false;
            List<Route> routes = session.getRoutes(item);
            Route defaultRoute = null;
            for (Route route : routes) {
                if (route.hasDefault()) defaultRoute = route;
                String toTitle = route.getDestinations().get(0).getTo().getTitle();
                if (languageLabel.equals(toTitle)) {
                    session.complete(item, route);
                    log.info("found matching wf route: " + languageLabel);
                    advanced = true;
                }
            }
            if (defaultRoute != null) {
                session.complete(item, defaultRoute);
            }
        } catch (Exception e) {
            log.error("Error executing TranslationProcessStep", e);
        }
    }
}

```

```

    }
    // fallback if no matching route was found
    if (!advanced) {
        if (defaultRoute != null) {
            session.complete(item, defaultRoute);
            log.info("no matching wf route found -> take default route!");
        } else {
            session.complete(item, routes.get(0));
            log.info("no matching wf route found and no route set " +
                "as default -> take first route!");
        }
    }
} catch (WorkflowException e) {
    log.error("Could not advance workflow.", e);
}
}


public void execute(WorkItem item, WorkflowSession session, String args[])
    throws Exception {
    // default language label to be used if actual page language label
    // isn't available (i.e. pages above language page)
    if (args != null && args.length > 0) defaultLanguageLabel = args[0];
    // the language page's hierarchy level
    if (args != null && args.length > 1)
        languageHierarchyLevel = Integer.getInteger(args[1]);
    log.info("default language label: " + defaultLanguageLabel +
        ", language hierarchy level: " + languageHierarchyLevel);
    execute(item, session);
}
}

```

Applies to

CQ5.2+

[ExampleTranslationWorkflow-1.1-2.zip](#)

 Twitter™ and Facebook posts are not covered under the terms of Creative Commons.

[Legal Notices](#) | [Online Privacy Policy](#)