

# Adobe CQ Help / How to Catch and Process Workflow Events

## Problem

You search for an example how to catch and process workflow events.

## Resolution

A Class to catch and handle Workflow Events must implement the Interfaces [EventHandler](#) and [JobProcessor](#).

This Class must be a OSGI Service and implement the method `process(Event event)`. In this method the provided object event ([Event](#)) can be used to access all event related data.

As soon as this Service is available in the system (for example when you include it in a bundle and install it) the workflow events will be caught and handled with this Class...

## Example

*Note:* The following example is just very simple and not to be used in production as it is. 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...

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

import javax.jcr.RepositoryException;
import javax.jcr.Session;

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

import com.day.cq.workflow.WorkflowService;
import com.day.cq.workflow.WorkflowSession;
import com.day.cq.workflow.WorkflowException;
import com.day.cq.workflow.exec.Workflow;
import com.day.cq.workflow.exec.WorkflowData;
import com.day.cq.workflow.event.WorkflowEvent;

import org.osgi.service.event.Event;
import org.osgi.service.event.EventHandler;
import org.apache.sling.event.EventUtil;
import org.apache.sling.event.JobProcessor;
import org.apache.sling.jcr.api.SlingRepository;

/**
 * <code>WorkFlowTest</code>...
 *
 * @scr.component metatype="false"
 * @scr.service interface="org.osgi.service.event.EventHandler"
 * @scr.property name="event.topics" refValues.a="com.day.cq.workflow.event.WorkflowEv
 */

public class WorkflowEventCatcher implements EventHandler, JobProcessor {

    protected final Logger log = LoggerFactory.getLogger(WorkflowEventCatcher.class);

    /**
     * @scr.reference policy="static"
     */
    private WorkflowService workflowService;

    /**
     * @scr.reference policy="static"
     */
    protected SlingRepository repository;
```

```

public void handleEvent(Event event) {
    EventUtil.processJob(event, this);
}

/**
 * @return boolean
 * @see org.apache.sling.event.JobProcessor#process(org.osgi.service.event.Event)
 */
public boolean process(Event event) {
    log.info("Received event of topic: " + event.getTopic());

    if (EventUtil.isLocal(event)) {
        WorkflowSession wfSession = null;
        try {
            wfSession = getWorkflowSession();
            String topic = event.getTopic();
            if (topic.equals(WorkflowEvent.EVENT_TOPIC)) {
                Object eventType = event.getProperty(WorkflowEvent.EVENT_TYPE);
                if (eventType.equals(WorkflowEvent.WORKFLOW_STARTED_EVENT)) {
                    log.info("Workflow has started");
                    String instanceId = (String) event.getProperty(WorkflowEvent.W
                try {
                    Workflow instance = wfSession.getWorkflow(instanceId);
                    WorkflowData data = instance.getWorkflowData();
                    // additional code comes here
                    // ...
                } catch (WorkflowException we) {
                    log.warn("Unable to load workflow instance", we);
                }
                } else if (eventType.equals(WorkflowEvent.WORKFLOW_COMPLETED_EVENT)
                    log.info("Workflow has ended");
                } else if (eventType.equals(WorkflowEvent.NODE_TRANSITION_EVENT))
                    log.info("Workflow in transition");
            }
        }
        catch (Exception e) {
            log.error(e.getMessage());
        }
        finally {
            if (wfSession != null) {
                wfSession.getSession().logout();
            }
        }
        return true;
    }
}

/**
 * helper to get WorkflowSession
 *
 * @return WorkflowSession
 */
private WorkflowSession getWorkflowSession() {
    try {
        Session adminSession = repository.loginAdministrative(null);
        return workflowService.getWorkflowSession(adminSession);
    } catch (RepositoryException re) {
        log.error(re.getMessage());
    }
    return null;
}
}

```

---

**Applies to**



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

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