

Workflow Events



You need to [set up your project and environment](#) go be able to handle Workflows events

You can have your plugin to *listen* to workflow events using Confluence's [Event Listeners](#).

Events

Event	Description
StateChangeEvent	Fired when the state of page or blog post is changed
StateExpireEvent	Fired when the state of a page of blog post has expired
ApprovalAssignedEvent	Fired when the approval is assigned
ApprovalUnassignedEvent	Fired when the approval is unassigned
ContentApproveEvent	Fired when the page or blog post is approved
ContentRejectEvent	Fired when the page or blog post is rejected
TaskCreateEvent	Fired when a task is created
TaskUpdatedEvent	Fired when a task is updated
TaskCompleteEvent	Fired when a task is completed
TaskCloseEvent	Fired when a task is closed (usually when the state changed and there are still open tasks)
TaskAssignEvent	Fired when a task is assigned
TaskDueDateExpiredEvent	Fired when a task due date is expired

Check Workflows API's `com.atlassian.workflow.event` package and subpackages for more event types.

Usage

Workflow events are handled the same way as Confluence events

```
public class WorkflowEventLogger implements EventListener {
    private static final Logger log = Logger.getLogger(WorkflowEventLogger.class);

    public void handleEvent(Event event) {
        if (log.isDebugEnabled()) {
            log.debug(event.toString());
        }
    }

    public Class[] getHandledEventClasses() {
        return new Class[] {
            // State Events
            StateChangeEvent.class,
            StateExpireEvent.class,
            ContentAssignEvent.class,
            // Approval event
            ContentApproveEvent.class,
            ContentRejectEvent.class,
            // Tasks Events
            TaskCreateEvent.class,
            TaskCompleteEvent.class,
            TaskCloseEvent.class,
            TaskAssignEvent.class,
        };
    }
}
```