# Event Handling

5 ways of event handling are detailed here :

- At the JCR lev­el with obser­va­tion – Event Listeners

- At the Sling lev­el with event han­dlers and jobs –Application level events like Replication - Event Handlers

- At the CQ lev­el with work­flows & launch­ers

- Par­tic­u­lar case of sched­uled events

- Par­tic­u­lar case of POST to the repos­i­to­ry

Why workflows when event handlers & event listeners are available???

* if of course there’s a human step at one stage of the pro­cess­ing.
* if the event han­dling process should be fre­quent­ly cus­tomized / para­me­ter­ized by non-admin users.

# Event listeners – JCR level Events

These are the used for lowest level of event handling by JCR observer. Below are the event lists

Event.NODE\_ADDED  
       Event.NODE\_MOVED  
       Event.NODE\_REMOVED  
       Event.PERSIST  
       Event.PROPERTY\_ADDED  
       Event.PROPERTY\_CHANGED  
       Event.PROPERTY\_REMOVED

To listen to an event at JCR level, a live session is needed. A handler is to be implemented to process the event after filtering

adminSession = repository.loginAdministrative(null);

   adminSession.getWorkspace().getObservationManager().addEventListener(

      this, //handler

      Event.PROPERTY\_ADDED|Event.NODE\_ADDED, //binary combination of event types

      "/apps/example", //path

      true, //is Deep?

      null, //uuids filter

      null, //nodetypes filter

      false);

public void onEvent(EventIterator eventIterator) {

  try {

    while (eventIterator.hasNext()){

      log.info("something has been added : {}", eventIterator.nextEvent().getPath());

    }

Event listeners listens to all the events be it your application or framework raised event. We will manually have to filter the events raised

# Event Handlers – Sling level Events

Sling offers with­in CQ an appli­ca­tion lev­el mod­el for han­dling events, which is called “Sling event­ing sup­port”. The events are sent by applications with come along with application related information.

@Component

@Service

@Property(name="event.topics",value= {**ReplicationAction.EVENT\_TOPIC, Example.EVENT\_TOPIC**})

public class ExampleEventHandler implements EventHandler {

 Logger log = LoggerFactory.getLogger(this.getClass());

 public void handleEvent(Event event) {

  if (event.getTopic().equals(Example.EVENT\_TOPIC)){

   log.info("Example {}, with description {} has been created...",event.getProperty(Example.PN\_NAME),event.getProperty(Example.PN\_DESC));

  } else if (event.getTopic().equals(ReplicationAction.EVENT\_TOPIC)){

   ReplicationAction action = ReplicationAction.fromEvent(event);

   log.info("User {} has tried to replicate {}",action.getUserId(),action.getPath());

  }

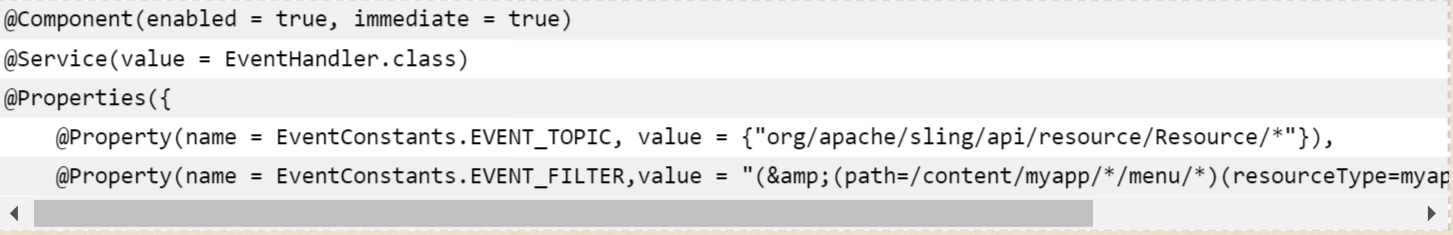
 }

}

-----------------------------------------------------------

Event listeners listens to all the events be it your application or framework raised event. We will manually have to filter the events raised.

Sling event has a property "event.filter" using which you can handle only those events which matches specific path and resourceType.



\* is the wild-card character and will match any sub paths below /content/myapp/ , "&" is AND operation saying filter also needs to match the resourceType to be fit for processing by event handler.

Eg for above ‘***event.filter***’ is available at C:\projects\dcloud\_feature\osgi\AuthoringEventListeners\src\main\java\com\cisco\wem\author\event\handler\ImageRenditionGenerationHandler.java

# Sling Jobs

Event Mechanism is a ‘***Fire event and forget it***’. There is no way is the triggered event was processed or not.

* In some use cases, importance is not given to ensure if the event has been processed atleast once
* In some, the guarantee of processing is a must

A job is a special event that has to be processed exactly once.

A job consists of two parts, the job topic describing the nature of the job and the payload which is a key value map of serializable objects.

Apache Sling supports the execution of jobs with the guarantee of processing the job at least once.

Sling jobs are persisted in repository under *‘/var/eventing/jobs’*

Reference : C:\projects\dcloud\_feature\osgi\XMLAuthoring\src\main\java\com\cisco\wem\xml\author\core\job\RenditionJobConsumer.java

Sling jobs work on publisher & consumer model. In which publisher publishes the event with properties where as consumer consumes the job, processes it and finally returns **JobResult.OK (**in case of success**), JobResult.Failed(**in case of retry**) & JobResult.CANCEL**

**Various Sling events available**

##### [com](eclipse-javadoc:%E2%98%82=authoringeventlisteners/C:%5C/Users%5C/hkesani%5C/.m2%5C/repository%5C/com%5C/adobe%5C/aem%5C/uber-jar%5C/6.3.0%5C/uber-jar-6.3.0-apis.jar%3Ccom).[day](eclipse-javadoc:%E2%98%82=authoringeventlisteners/C:%5C/Users%5C/hkesani%5C/.m2%5C/repository%5C/com%5C/adobe%5C/aem%5C/uber-jar%5C/6.3.0%5C/uber-jar-6.3.0-apis.jar%3Ccom.day).[cq](eclipse-javadoc:%E2%98%82=authoringeventlisteners/C:%5C/Users%5C/hkesani%5C/.m2%5C/repository%5C/com%5C/adobe%5C/aem%5C/uber-jar%5C/6.3.0%5C/uber-jar-6.3.0-apis.jar%3Ccom.day.cq).[replication](eclipse-javadoc:%E2%98%82=authoringeventlisteners/C:%5C/Users%5C/hkesani%5C/.m2%5C/repository%5C/com%5C/adobe%5C/aem%5C/uber-jar%5C/6.3.0%5C/uber-jar-6.3.0-apis.jar%3Ccom.day.cq.replication).ReplicationAction

##### [com](eclipse-javadoc:%E2%98%82=authoringeventlisteners/C:%5C/Users%5C/hkesani%5C/.m2%5C/repository%5C/com%5C/adobe%5C/aem%5C/uber-jar%5C/6.3.0%5C/uber-jar-6.3.0-apis.jar%3Ccom).[day](eclipse-javadoc:%E2%98%82=authoringeventlisteners/C:%5C/Users%5C/hkesani%5C/.m2%5C/repository%5C/com%5C/adobe%5C/aem%5C/uber-jar%5C/6.3.0%5C/uber-jar-6.3.0-apis.jar%3Ccom.day).[cq](eclipse-javadoc:%E2%98%82=authoringeventlisteners/C:%5C/Users%5C/hkesani%5C/.m2%5C/repository%5C/com%5C/adobe%5C/aem%5C/uber-jar%5C/6.3.0%5C/uber-jar-6.3.0-apis.jar%3Ccom.day.cq).[wcm](eclipse-javadoc:%E2%98%82=authoringeventlisteners/C:%5C/Users%5C/hkesani%5C/.m2%5C/repository%5C/com%5C/adobe%5C/aem%5C/uber-jar%5C/6.3.0%5C/uber-jar-6.3.0-apis.jar%3Ccom.day.cq.wcm).[api](eclipse-javadoc:%E2%98%82=authoringeventlisteners/C:%5C/Users%5C/hkesani%5C/.m2%5C/repository%5C/com%5C/adobe%5C/aem%5C/uber-jar%5C/6.3.0%5C/uber-jar-6.3.0-apis.jar%3Ccom.day.cq.wcm.api).PageEvent

[com](eclipse-javadoc:%E2%98%82=authoringeventlisteners/C:%5C/Users%5C/hkesani%5C/.m2%5C/repository%5C/com%5C/adobe%5C/aem%5C/uber-jar%5C/6.3.0%5C/uber-jar-6.3.0-apis.jar%3Ccom).[day](eclipse-javadoc:%E2%98%82=authoringeventlisteners/C:%5C/Users%5C/hkesani%5C/.m2%5C/repository%5C/com%5C/adobe%5C/aem%5C/uber-jar%5C/6.3.0%5C/uber-jar-6.3.0-apis.jar%3Ccom.day).[cq](eclipse-javadoc:%E2%98%82=authoringeventlisteners/C:%5C/Users%5C/hkesani%5C/.m2%5C/repository%5C/com%5C/adobe%5C/aem%5C/uber-jar%5C/6.3.0%5C/uber-jar-6.3.0-apis.jar%3Ccom.day.cq).[dam](eclipse-javadoc:%E2%98%82=authoringeventlisteners/C:%5C/Users%5C/hkesani%5C/.m2%5C/repository%5C/com%5C/adobe%5C/aem%5C/uber-jar%5C/6.3.0%5C/uber-jar-6.3.0-apis.jar%3Ccom.day.cq.dam).[api](eclipse-javadoc:%E2%98%82=authoringeventlisteners/C:%5C/Users%5C/hkesani%5C/.m2%5C/repository%5C/com%5C/adobe%5C/aem%5C/uber-jar%5C/6.3.0%5C/uber-jar-6.3.0-apis.jar%3Ccom.day.cq.dam.api).DamEvent

# Useful Resources

<https://www.linkedin.com/pulse/aem-how-write-sling-jobs-aem-veena-vikraman>

<https://blogs.adobe.com/experiencedelivers/experience-management/event_handling_incq/>

<http://techagyan.blogspot.com/2015/02/aem-event-handling-and-filtering.html>