**TASK-8 CODE**

**package** com.adobe.aem.guides.demo.core.listeners;

**import** java.util.HashMap;

**import** java.util.Map;

**import** javax.jcr.Session;

**import** org.apache.sling.api.resource.LoginException;

**import** org.apache.sling.api.resource.ModifiableValueMap;

**import** org.apache.sling.api.resource.PersistenceException;

**import** org.apache.sling.api.resource.Resource;

**import** org.apache.sling.api.resource.ResourceResolver;

**import** org.apache.sling.api.resource.ResourceResolverFactory;

**import** org.osgi.service.component.annotations.Component;

**import** org.osgi.service.component.annotations.Reference;

**import** org.osgi.service.event.Event;

**import** org.osgi.service.event.EventConstants;

**import** org.osgi.service.event.EventHandler;

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

**import** com.day.cq.replication.ReplicationAction;

@Component(service = EventHandler.**class**, immediate = **true**,

property = {

EventConstants.***EVENT\_TOPIC*** + "=" + ReplicationAction.***EVENT\_TOPIC***

}

)

**public** **class** HemanthHandlerJuly **implements** EventHandler {

**private** **static** **final** Logger ***log*** = LoggerFactory.*getLogger*(HemanthHandlerJuly.**class**);

@Reference

**private** WorkflowService workflowService;

@Reference

**private** ResourceResolverFactory resolverFactory;

@Override

**public** **void** handleEvent(Event event) {

***log***.info("Handle my trigger");

handleReplicationEvent(event);

}

**private** **void** handleReplicationEvent(Event event) {

ReplicationAction action = ReplicationAction.*fromEvent*(event);

String path = action.getPath();

***log***.info("Content activated at path: {}", path);

// startWorkflow(path);

addPropertyToPage(path);

}

**private** **void** addPropertyToPage(String pagePath) {

Map<String, Object> param = **new** HashMap<>();

param.put(ResourceResolverFactory.***SUBSERVICE***, "hemanth");

**try** (ResourceResolver resolver = resolverFactory.getServiceResourceResolver(param)) {

String jcrContentPath = pagePath + "/jcr:content";

Resource resource = resolver.getResource(jcrContentPath);

**if** (resource != **null**) {

ModifiableValueMap properties = resource.adaptTo(ModifiableValueMap.**class**);

**if** (properties != **null**) {

properties.put("changed", **true**);

resolver.commit();

***log***.info("Property 'changed' set to true for: {}", jcrContentPath);

}

} **else** {

***log***.warn("Resource not found at path: {}", jcrContentPath);

}

} **catch** (LoginException e) {

***log***.error("LoginException while adding property", e);

} **catch** (PersistenceException e) {

***log***.error("PersistenceException while adding property", e);

} **catch** (Exception e) {

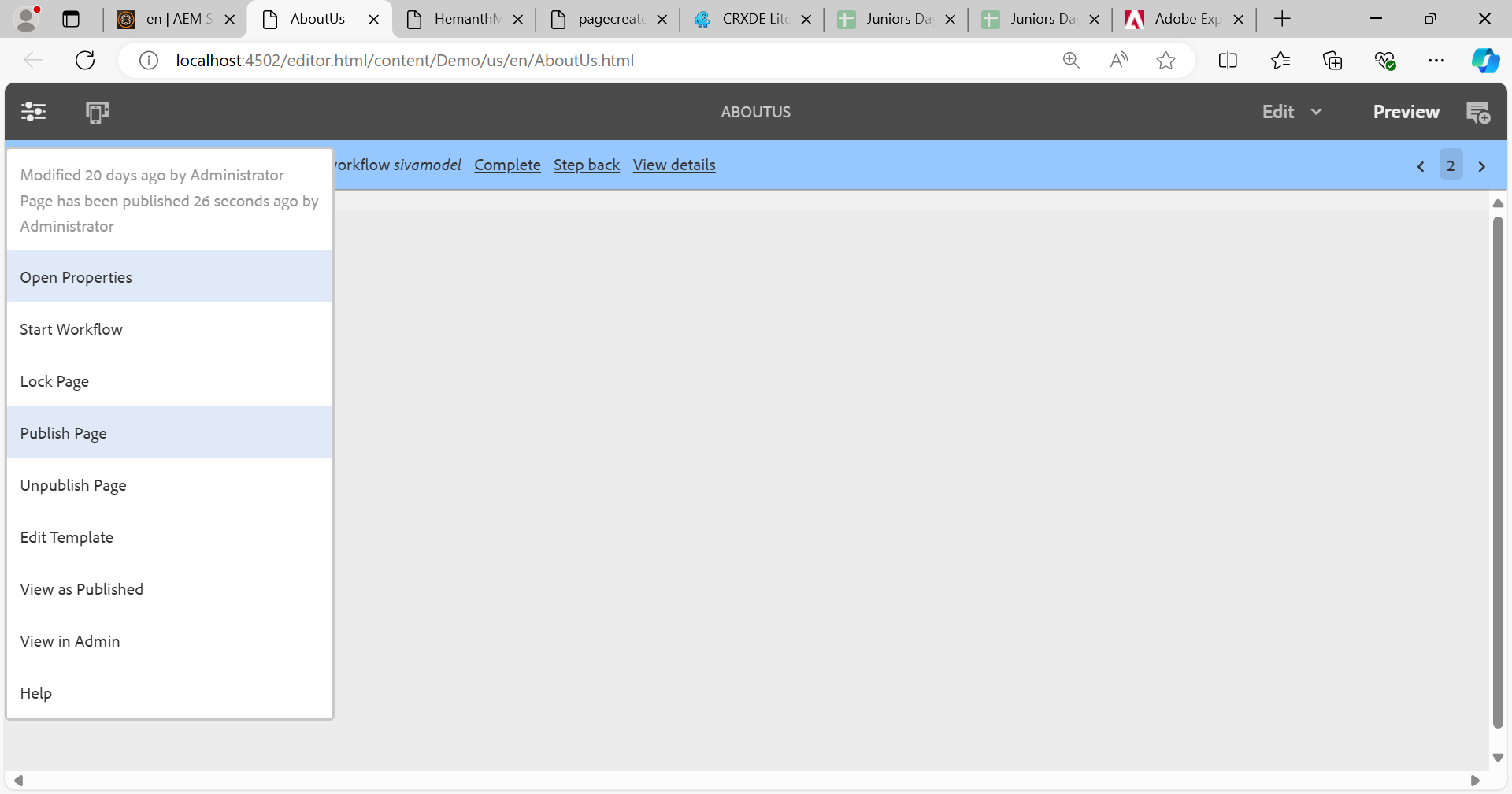
***log***.error("Exception while adding property", e);

}

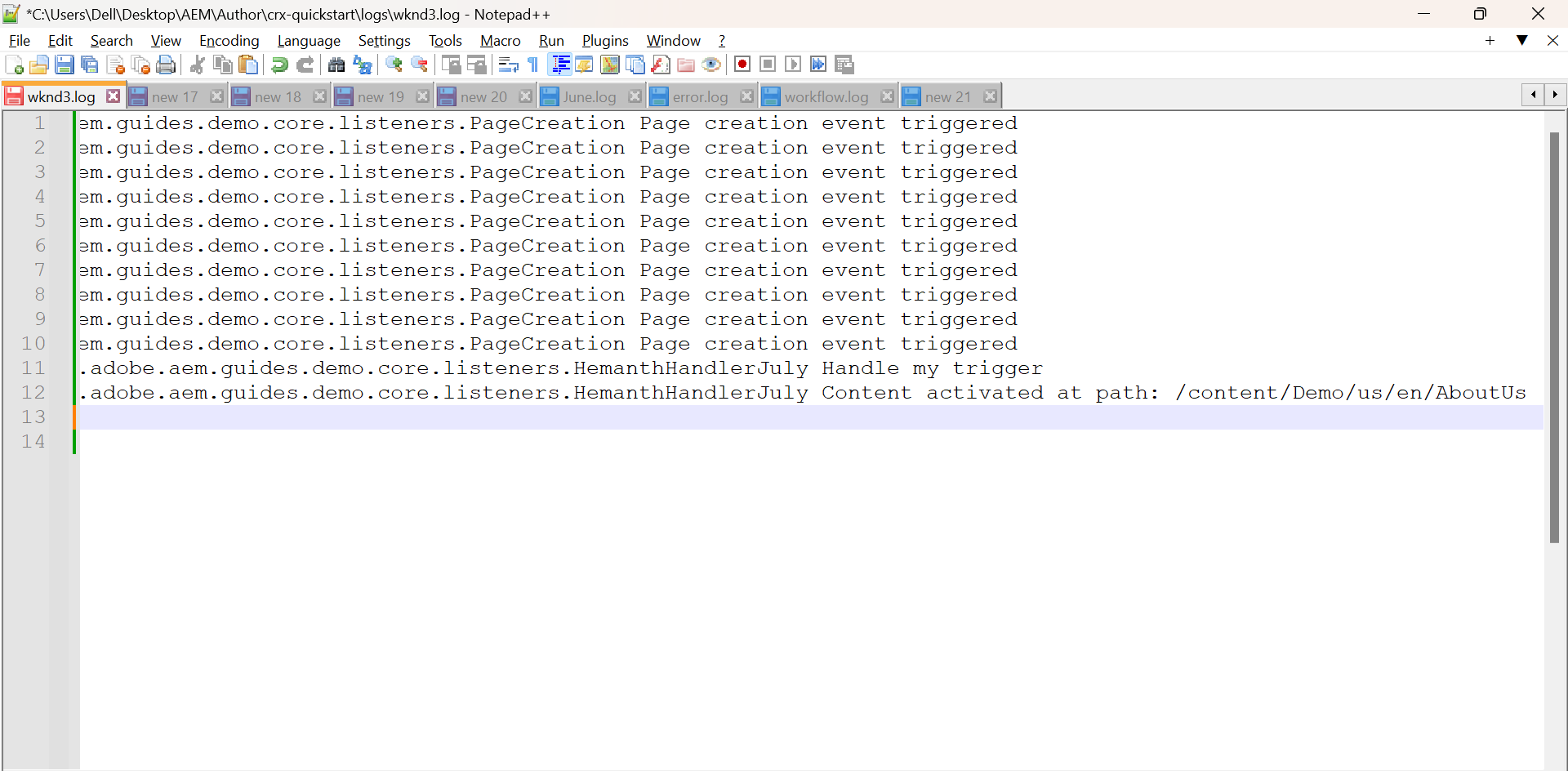
}

}

**PUBLSIHED PAGE**



**PUBLSIHED PATH**



**PROPERTY ADDED**

