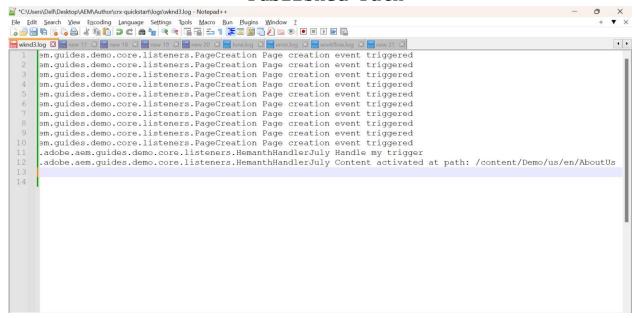# TASK-7 CODE

```java
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);
}
}
```

Published Path

Published Page