

Task-13 CODE

QUESTION

1. Create a scheduler. When I provide any cron expression for 3 minutes, after 3 minutes, whatever page has the `Expirydate` property with the current date and time should be published, and those with previous dates and times should be unpublished.

Answer :

BY USING SCHEDULER PUBLISHING AND UNPUBLISHING THE PAGE

```
package com.adobe.aem.guides.demo.core.schedulers;

import org.osgi.service.metatype.annotations.AttributeDefinition;
import org.osgi.service.metatype.annotations.AttributeType;
import org.osgi.service.metatype.annotations.ObjectClassDefinition;

@ObjectClassDefinition(name = "PreviousDatePublish",description = "scheduler is
created for publish")
public @interface PresentDatePublish {

    @AttributeDefinition(
        name="service name",
        type = AttributeType.STRING,
        description = "enter the service name"
    )
    public String getservice_name() default "practise";
    @AttributeDefinition
    (
        name="can run concurrently",
        type=AttributeType.BOOLEAN,
        description="can run concurrently"
    )
```

```
public boolean canrunconcurrently() default false;
```

```
@AttributeDefinition
```

```
(
```

```
    name="Enabled scheduler",
```

```
    type=AttributeType.BOOLEAN,
```

```
    description="Enable the scheduler"
```

```
)
```

```
public boolean Enabledscheduler() default true;
```

```
@AttributeDefinition(
```

```
    name="Expression",
```

```
    type = AttributeType.STRING,
```

```
    description = "enter the Expression"
```

```
)
```

```
public String getExpressi();
```

```
}
```

SECONDSCHEDULER

```
package com.adobe.aem.guides.demo.core.schedulers;
```

```
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.HashMap;  
import java.util.Map;
```

```
import javax.jcr.Node;  
import javax.jcr.RepositoryException;  
import javax.jcr.Session;  
import javax.jcr.query.Query;  
import javax.jcr.query.QueryManager;  
import javax.jcr.query.QueryResult;
```

```
import org.apache.sling.api.resource.LoginException;  
import org.apache.sling.api.resource.ResourceResolver;  
import org.apache.sling.api.resource.ResourceResolverFactory;  
import org.apache.sling.commons.scheduler.ScheduleOptions;  
import org.apache.sling.commons.scheduler.Scheduler;  
import org.osgi.service.component.annotations.Activate;  
import org.osgi.service.component.annotations.Component;  
import org.osgi.service.component.annotations.Deactivate;  
import org.osgi.service.component.annotations.Modified;  
import org.osgi.service.component.annotations.Reference;  
import org.osgi.service.metatype.annotations.Designate;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;
```

```
import com.day.cq.replication.ReplicationActionType;  
import com.day.cq.replication.ReplicationException;  
import com.day.cq.replication.Replicator;  
import com.day.cq.wcm.api.Page;  
import com.day.cq.wcm.api.PageManager;
```

```
@Designate(ocd = PresentDatePublish.class)
```

```
@Component(service = Runnable.class, immediate = true)
```

```
public class PresentDatePublishsecond implements Runnable {
```

```
    private static final Logger log = LoggerFactory.getLogger(PresentDatePublishsecond.class);
```

```
    @Reference
```

```
    private Scheduler scheduler;
```

```
    @Reference
```

```

private ResourceResolverFactory resourceResolverFactory;

@Reference
private Replicator replicator;

private static final String SERVICE_USER = "hemanth";

private String cronExpression;

@Modified
public void modify(PresentDatePublish sch) {
    this.cronExpression = sch.getExpressi();
    scheduleJob(sch); // Re-schedule with the new cron expression
}

@Activate
public void activate(PresentDatePublish sch) {
    this.cronExpression = sch.getExpressi();
    scheduleJob(sch);
}

private void scheduleJob(PresentDatePublish sch) {
    log.info("Scheduler is created");
    if (sch.Enabledscheduler()) {
        ScheduleOptions scheduleOptions = scheduler.EXPR(cronExpression);
        scheduleOptions.canRunConcurrently(sch.canrunconcurrently());
        scheduleOptions.name(sch.getservice_name());
        scheduler.schedule(this, scheduleOptions);
    }
}

@Deactivate
public void deactivate(PresentDatePublish sch) {
    unscheduleJob(sch);
}

private void unscheduleJob(PresentDatePublish sch) {
    log.info("Job is unscheduled");
    scheduler.unschedule(sch.getservice_name());
}

@Override
public void run() {
    log.info("Scheduler is running in present date");
    log.info("My cron expression in present date: " + cronExpression);
}

```

```

    try (ResourceResolver resourceResolver = getServiceResourceResolver()) {
        if (resourceResolver != null) {
            handlePageReplication(resourceResolver);
        }
    } catch (LoginException e) {
        log.error("Failed to get service resource resolver", e);
    } catch (ReplicationException e) {
        log.error("ReplicationException occurred", e);
    } catch (Exception e) {
        log.error("An unexpected error occurred", e);
    }
}

private ResourceResolver getServiceResourceResolver() throws LoginException {
    Map<String, Object> param = new HashMap<>();
    param.put(ResourceResolverFactory.SUBSERVICE, SERVICE_USER);
    return resourceResolverFactory.getServiceResourceResolver(param);
}

private void handlePageReplication(ResourceResolver resourceResolver) throws
ReplicationException {
    try {
        Session session = resourceResolver.adaptTo(Session.class);
        if (session == null) {
            log.error("Could not obtain a JCR session.");
            return;
        }

        String queryString = "SELECT * FROM [cq:PageContent] AS content WHERE
ISDESCENDANTNODE(content, '/content/') AND content.[expirydate] IS NOT NULL";
        QueryManager queryManager = session.getWorkspace().getQueryManager();
        Query query = queryManager.createQuery(queryString, Query.JCR_SQL2);
        QueryResult result = query.execute();

        javax.jcr.NodeIterator nodes = result.getNodes();
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-
dd'T'HH:mm:ss.SSSXXX");
        String currentDateStr = sdf.format(new Date());

        while (nodes.hasNext()) {
            Node node = nodes.nextNode();
            if (node.hasProperty("expirydate")) {
                String expiryDate = node.getProperty("expirydate").getString();
                PageManager pageManager = resourceResolver.adaptTo(PageManager.class);
                Page page = pageManager.getContainingPage(node.getPath());
            }
        }
    }
}

```

```

        if (page != null) {
            if (expiryDate.startsWith(currentDateStr.substring(0, 10))) {
                // Publish the page
                replicator.replicate(session, ReplicationActionType.ACTIVATE,
page.getPath());
                log.info("Published page: " + page.getPath());
            } else if (expiryDate.compareTo(currentDateStr) < 0) {
                // Unpublish the page
                replicator.replicate(session, ReplicationActionType.DEACTIVATE,
page.getPath());
                log.info("Unpublished page: " + page.getPath());
            }
        }
    }
}
}
} catch (RepositoryException e) {
    log.error("Error while querying and replicating pages", e);
}
}
}

```

CORN EXPRESSION

The screenshot shows the Adobe AEM console configuration page for a scheduler named "practice". The page is titled "PreviousDatePublish" and includes a "scheduler is created for publish" message. The configuration fields are as follows:

- service name:** practice (with a hint "enter the service name (getservice.name)")
- can run concurrently:** ☐ (with a hint "can run concurrently (canrunconcurrently)")
- Enabled scheduler:** ☒ (with a hint "Enable the scheduler (Enabledscheduler)")
- Expression:** 0 0/2 * 1/1 * ? * (with a hint "enter the Expression (getExpressi)")

The "Configuration Information" section shows:

- Persistent Identity (PID):** com.adobe.aem.guides.demo.core.schedulers.PresentDatePublishsecond
- Configuration Binding:** Unbound: or new configuration

At the bottom of the configuration form are buttons: Cancel, Reset, Delete, Unbind, and Save. Below the configuration form is a table listing other services:

Service Name	Configuration	Actions
Reporting Services API proxy servlet	-	edit, delete
Reporting Services settings provider	-	edit, delete
Routing Config	-	edit, delete, add
RTE Filter Servlet Amendment	-	add

PREVIOUS DATES UNPUBLISHED PRESENT DATES PUBLISHED

```
C:\Users\De\l\Desktop\AEM\Author\crx-quickstart\logs\June.log - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
wknd3.log new 17 new 18 new 19 new 20 June.log error.log workflow.log
91 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Unpublished page: /content/Demo/us/en/hemujuilylauncher/shiva
92 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Published page: /content/Demo/us/en/siddaih0
93 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Unpublished page: /content/Demo/us/en/siddaih
94 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Scheduler is running in present date
95 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond My cron expression in present date: 0 0/2 * 1/1 * ? *
96 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Unpublished page: /content/we-retail/language-masters/nagendra
97 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Unpublished page: /content/Demo/us/en/article3/lohitanna
98 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Unpublished page: /content/Demo/us/en/workflowpage/hemu
99 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Unpublished page: /content/Demo/us/en/workflowpage/shiva
100 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Published page: /content/Demo/us/en/AboutUs/Login
101 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Unpublished page: /content/Demo/us/en/hemujuilylauncher/shiva
102 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Published page: /content/Demo/us/en/siddaih0
103 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Unpublished page: /content/Demo/us/en/siddaih
104 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Scheduler is running in present date
105 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond My cron expression in present date: 0 0/2 * 1/1 * ? *
106 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Unpublished page: /content/we-retail/language-masters/nagendra
107 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Unpublished page: /content/Demo/us/en/article3/lohitanna
108 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Unpublished page: /content/Demo/us/en/workflowpage/hemu
109 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Unpublished page: /content/Demo/us/en/workflowpage/shiva
110 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Published page: /content/Demo/us/en/AboutUs/Login
111 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Unpublished page: /content/Demo/us/en/hemujuilylauncher/shiva
112 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Published page: /content/Demo/us/en/siddaih0
113 |be.aem.guides.demo.core.schedulers.PresentDatePublishsecond Unpublished page: /content/Demo/us/en/siddaih
114
```