Configure Maintenance Tasks

Agenda:

- Maintenance tasks
- Maintenance windows
- Workflow purge task

Maintaining the repository size

Node structures can increase the repository size massively—if not maintained

- For example, Adobe Experience Manager (AEM):
  - Creates a new page version when it is published
  - Creates a new asset version when it is modified or published
  - Maintains a robust audit log of content changes that include edits, moves, publication, and deletions
  - Stores a history of every workflow that is executed
  - Never removes ad-hoc tasks and projects completely from the system

These actions can add significant size to the repository if no mechanism is in place to remove the old nodes over a period

AEM maintenance tasks

Maintenance tasks are:

- OSGi Configuration stored in **/apps**
- Added to a single Maintenance window

Maintenance windows:

- Are context aware config stored in **/apps**
- Have a start and end date and time
  - If a task is not complete within the defined time, it continues in the next window
- Can be set daily, weekly, or monthly

Control repository size with maintenance tasks

Tasks owned by Adobe:

- Datastore Garbage Collection

Tasks that have shared ownership:

- Version Purge
- Audit Log Purge

Task owned by the customer:

- Ad-hoc Task Purge
- Workflow Purge
- Project Purge

Adobe owned and shared tasks

Adobe Owned

- Adobe owns the maintenance window and OSGi Configuration of:
  - Datastore garbage collection

Shared

- Adobe owns the maintenance window
- Customer owns the OSGi Configuration:
  - `com.day.cq.wcm.core.impl.VersionPurgeTask`
    - ❑ Purge path, max or min number versions and max version age
  - `com.adobe.cq.audit.purge.Pages` (factory)
    - ❑ Content path, min age, and page events to audit
- Adobe owns the right to override these configurations to mitigate performance degradation

Customer owned tasks

Adjust maintenance window configurations

- Add or remove tasks in each window
- Configure running windows

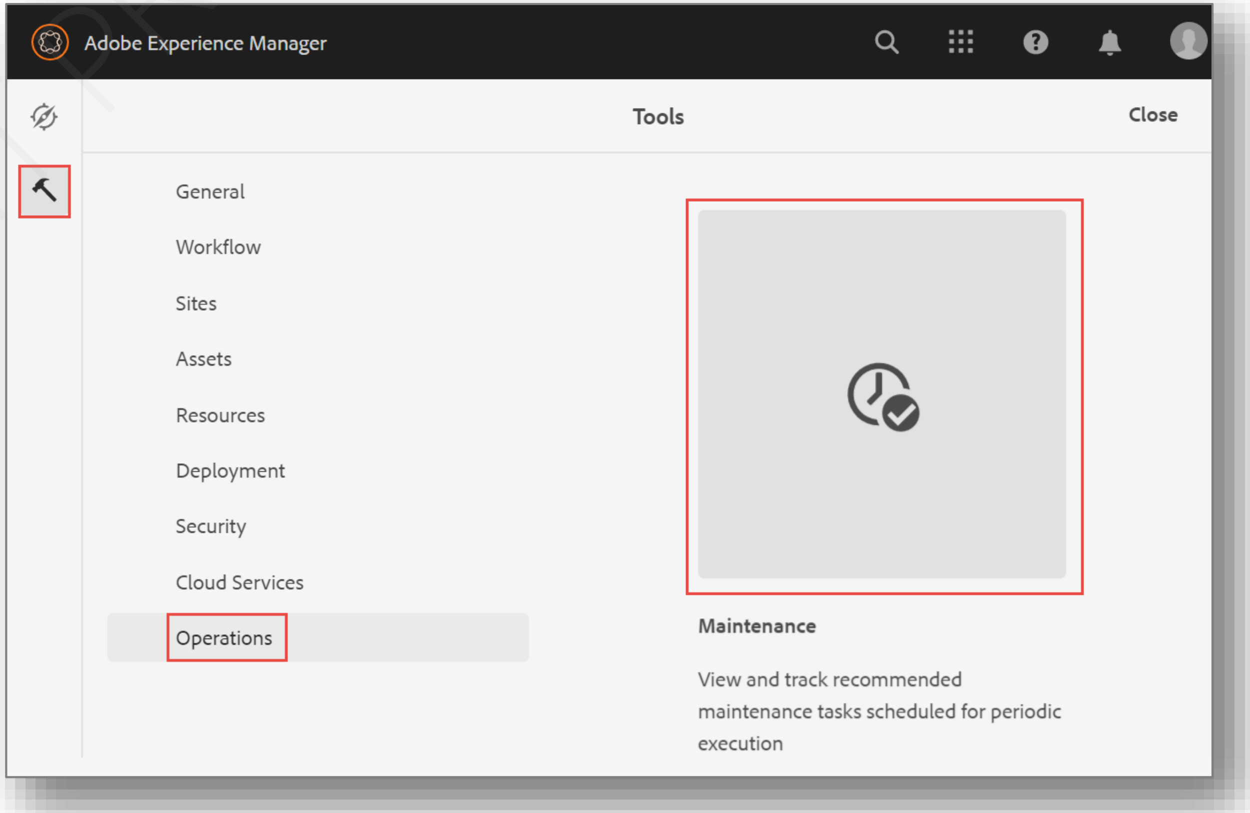
Can Modify OSGi configurations of:

- `com.adobe.granite.taskmanagement.impl.purge.TaskPurgeMaintenanceTask`
- `com.adobe.granite.workflow.purge.Scheduler` (factory)
  - 0 to many configurations
- `com.adobe.cq.projects.purge.Scheduler` (factory)
  - 0 to many configurations

OSGi configurations must be stored in /apps/*/config folders

Maintenance console

- Tools > Operations > Maintenance
  - Only available for local development
  - Changes must be checked into git
- Console is used to:
  - Adjust maintenance window
  - Add or remove tasks in window
  - Find OSGi configurations quickly
- Console is a convenience for creating and configuring tasks
  - All new configurations must be added to git under /apps/settings



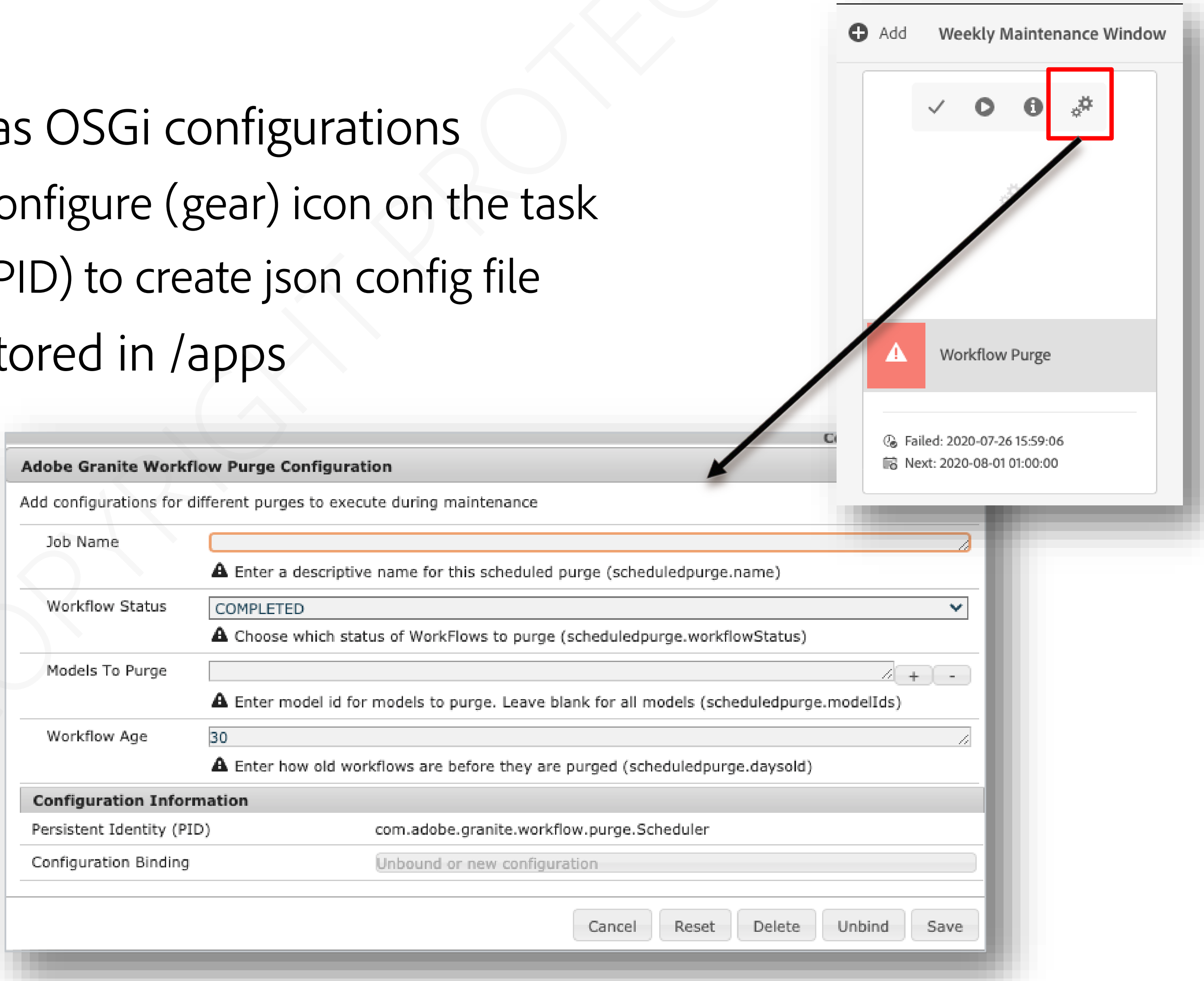
OSGi task configurations

Task configurations are stored as OSGi configurations

- Find them easily from the Configure (gear) icon on the task
- Use the Persistent Identity (PID) to create json config file

OSGi Configurations must be stored in /apps

- Checked into git



Failed: 2020-07-26 15:59:06
Next: 2020-08-01 01:00:00

Adobe Granite Workflow Purge Configuration

Add configurations for different purges to execute during maintenance

Job Name:
Enter a descriptive name for this scheduled purge (scheduledpurge.name)

Workflow Status: COMPLETED
Choose which status of WorkFlows to purge (scheduledpurge.workflowStatus)

Models To Purge:
Enter model id for models to purge. Leave blank for all models (scheduledpurge.modelIds)

Workflow Age: 30
Enter how old workflows are before they are purged (scheduledpurge.daysold)

Configuration Information

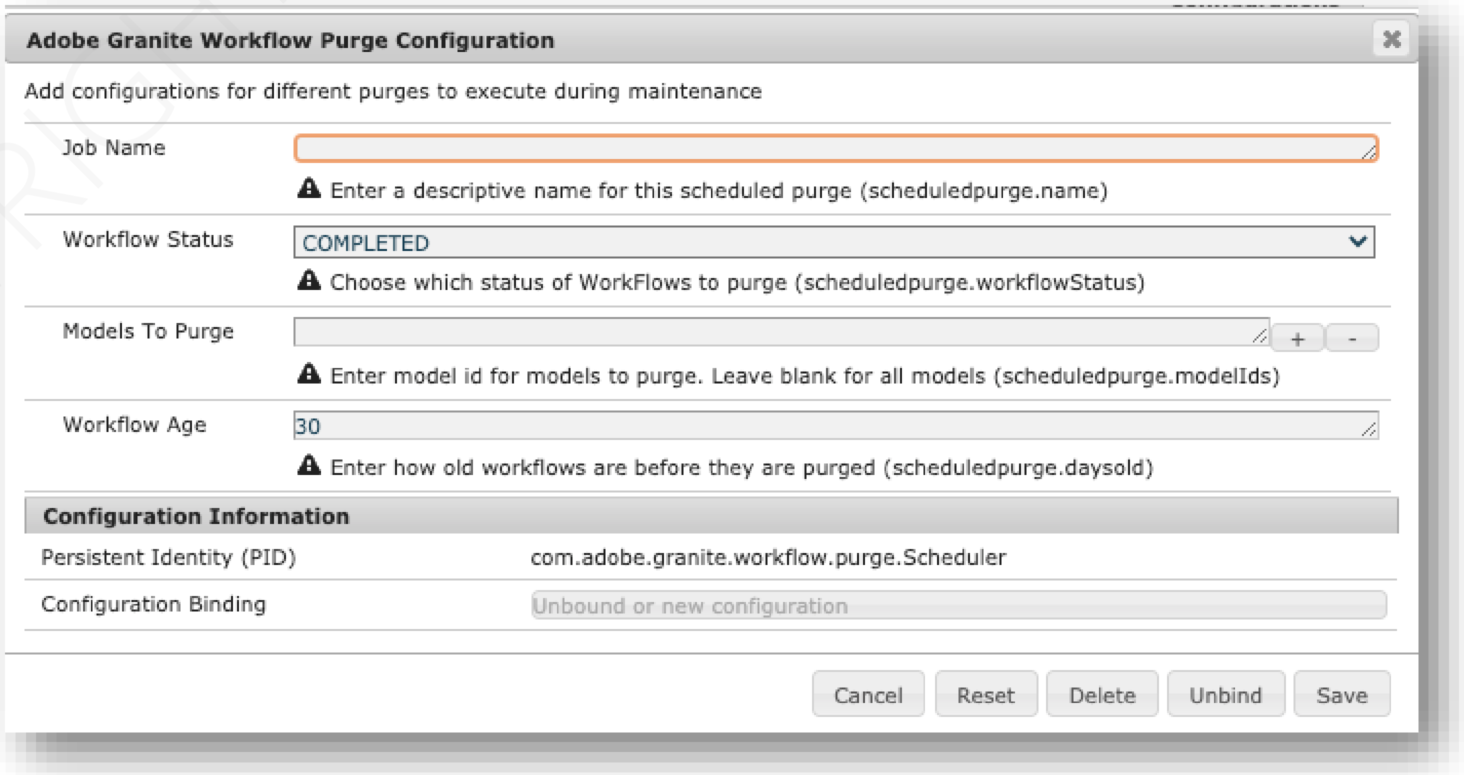
Persistent Identity (PID): com.adobe.granite.workflow.purge.Scheduler

Configuration Binding: Unbound or new configuration

Workflow Purge Configuration – An example

Workflow information can be purged based on:

- Specific model IDs to purge:
  - Example ID: /var/workflow/models/myWF
  - 1 to many model IDs
  - Blank – All workflows
- Status: COMPLETED or RUNNING
- Age of workflow (in days)



Adobe Granite Workflow Purge Configuration

Add configurations for different purges to execute during maintenance

Job Name:

⚠ Enter a descriptive name for this scheduled purge (scheduledpurge.name)

Workflow Status: COMPLETED

⚠ Choose which status of WorkFlows to purge (scheduledpurge.workflowStatus)

Models To Purge:

⚠ Enter model id for models to purge. Leave blank for all models (scheduledpurge.modelIds)

Workflow Age: 30

⚠ Enter how old workflows are before they are purged (scheduledpurge.daysold)

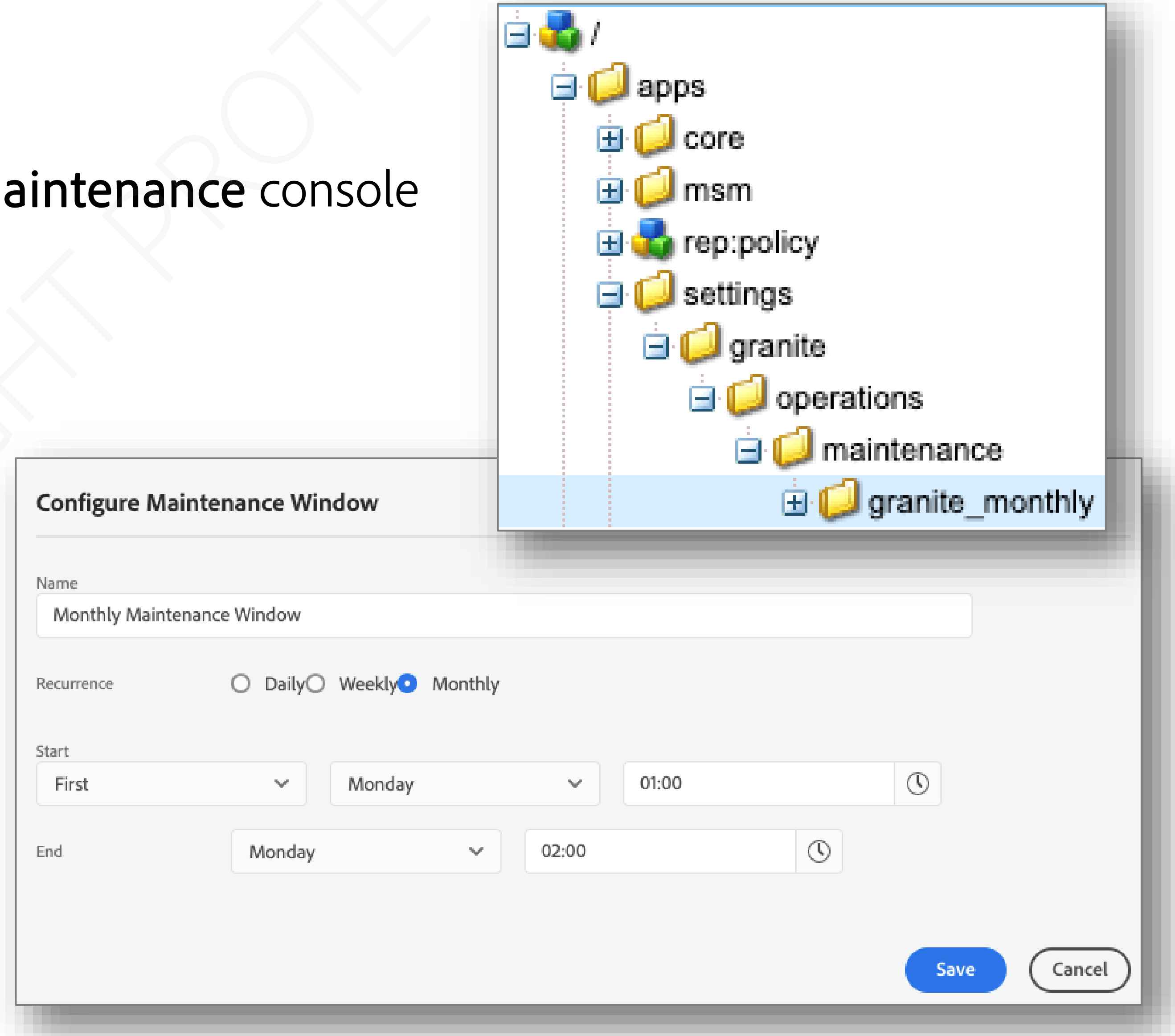
Configuration Information

Persistent Identity (PID): com.adobe.granite.workflow.purge.Scheduler

Configuration Binding: Unbound or new configuration

Monthly windows

- Must be manually created in JCR
- Can be modified or configured from the **Maintenance** console
- Can be scheduled:
  - On first or last day of month
  - On the day's start and end time



Configure Maintenance Window

Name: Monthly Maintenance Window

Recurrence: ○ Daily ○ Weekly ● Monthly

Start: First | Monday | 01:00

End: Monday | 02:00

Add maintenance tasks to source control

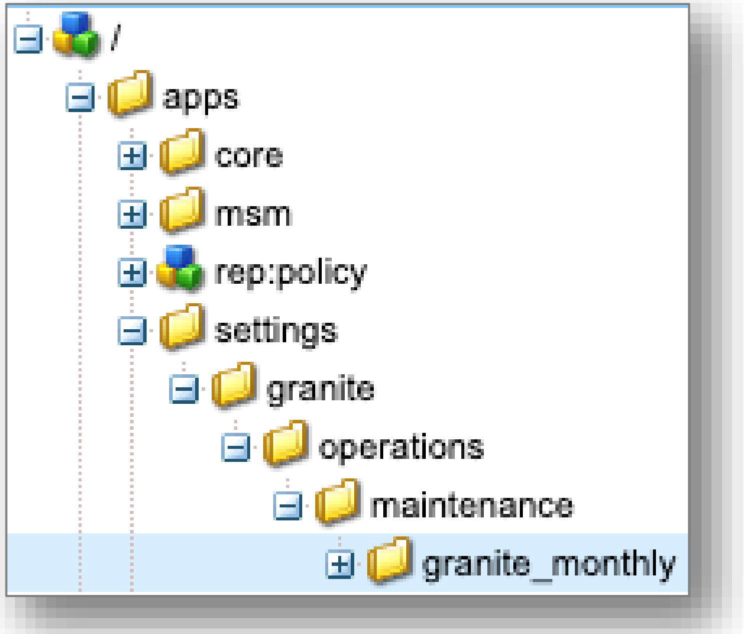
Maintenance UI adds maintenance windows and tasks under:

- **`/conf/global`**`/settings/granite/operations/maintenance`

After configuring, they must be checked into source control under:

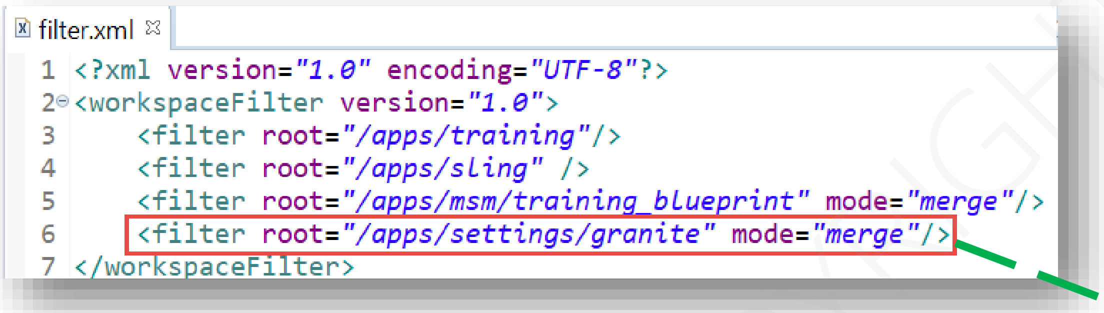
- **`/apps`**`/settings/granite/operations/maintenance`

Allows the maintenance windows/tasks to be immutable



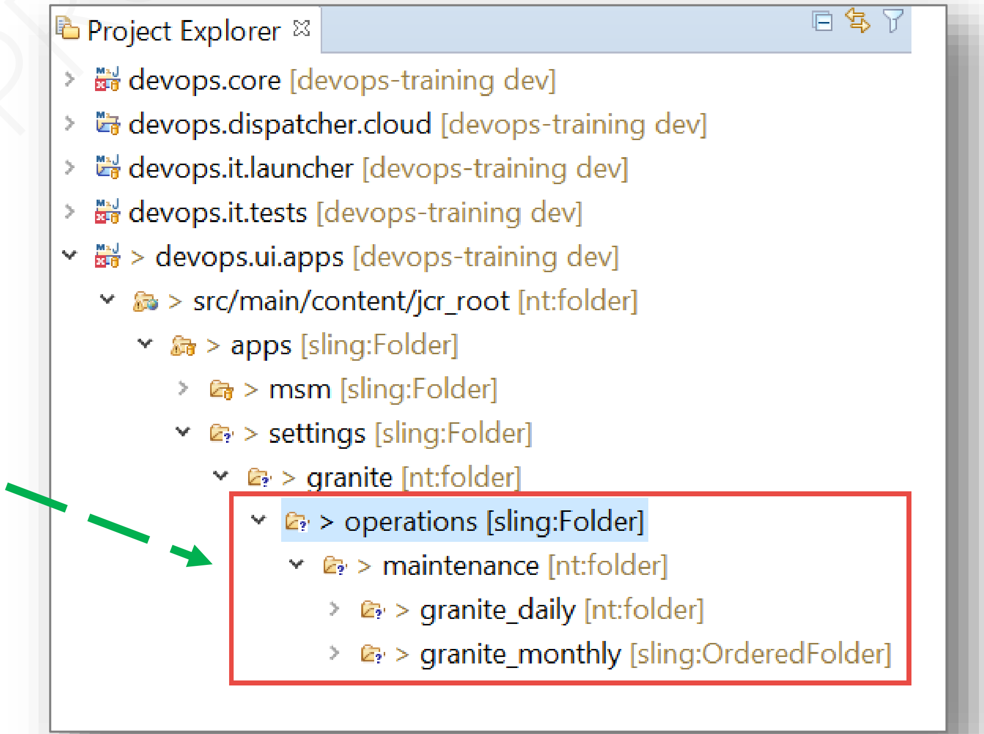
Sync the configurations with your Maven project

- Update **ui.apps** > **/src/main/content/META-INF > filter.xml** with the configurations as needed:



filter.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<workspaceFilter version="1.0">
    <filter root="/apps/training"/>
    <filter root="/apps/sling" />
    <filter root="/apps/msm/training_blueprint" mode="merge"/>
    <filter root="/apps/settings/granite" mode="merge"/>
</workspaceFilter>
```



Project Explorer

```
> devops.core [devops-training dev]
> devops.dispatcher.cloud [devops-training dev]
> devops.it.launcher [devops-training dev]
> devops.it.tests [devops-training dev]
v > devops.ui.apps [devops-training dev]
    v > src/main/content/jcr_root [nt:folder]
        v > apps [sling:Folder]
            > > msm [sling:Folder]
            v > settings [sling:Folder]
                v > granite [nt:folder]
                    v > operations [sling:Folder]
                        v > maintenance [nt:folder]
                            > > granite_daily [nt:folder]
                            > > granite_monthly [sling:OrderedFolder]
```

- Right-click the **ui.apps** > **src/main/content/jcr_root** > **apps** and select **Import from server**

Key takeaways

- Maintenance tasks are processes that run on a schedule to optimize the repository
- Use OSGi configurations to configure maintenance tasks specific to business requirements for purging content
- Schedule the maintenance tasks to be executed during the daily, weekly, or monthly maintenance windows
- Use the **Maintenance** console to configure the windows on a local AEM instance, synchronize them back into the AEM project, and check them into source control
- Configure **Adobe Granite Workflow Purge Configuration** to purge workflow instances according their age and status