Snippets

Service:

@Component (name = "…", configurationFactory = true, immediate = true, metatype = true, label = "…")

@Service (….class)

@Property (label = "…", name = "…", description = "…", unbounded = PropertyUnbounded.ARRAY),

@Property (label = "…", name = "service.id", description = "…", value = "…"),

@Property(name = "webconsole.configurationFactory.nameHint", value = {"… : {service.id}" })

@Activate

protected final void activate(final [org](eclipse-javadoc:%E2%98%82=aem-foundation-core/H:%5C/maven%5C/repo%5C/org%5C/osgi%5C/org.osgi.compendium%5C/4.2.0%5C/org.osgi.compendium-4.2.0.jar%3Corg).[osgi](eclipse-javadoc:%E2%98%82=aem-foundation-core/H:%5C/maven%5C/repo%5C/org%5C/osgi%5C/org.osgi.compendium%5C/4.2.0%5C/org.osgi.compendium-4.2.0.jar%3Corg.osgi).[service](eclipse-javadoc:%E2%98%82=aem-foundation-core/H:%5C/maven%5C/repo%5C/org%5C/osgi%5C/org.osgi.compendium%5C/4.2.0%5C/org.osgi.compendium-4.2.0.jar%3Corg.osgi.service).[component](eclipse-javadoc:%E2%98%82=aem-foundation-core/H:%5C/maven%5C/repo%5C/org%5C/osgi%5C/org.osgi.compendium%5C/4.2.0%5C/org.osgi.compendium-4.2.0.jar%3Corg.osgi.service.component).ComponentContext con ) {

final java.util.Dictionary<?, ?> properties = context.getProperties();

Object sender = properties.get("redirectmapChangeReview.sender");

org.apache.sling.commons.osgi.PropertiesUtil.toString(object, StringUtils.EMPTY);

org.apache.sling.commons.osgi.PropertiesUtil.toStringArray(object);

}

Use in JSP: SampleService service=sling.getService(SampleService.class);

Use via Java:

BundleContext bundleContext = FrameworkUtil.getBundle(MyClass.class).getBundleContext();

ServiceReference factoryRef = bundleContext.getServiceReference(ResourceResolverFactory.class.getName());

ResourceResolverFactory resolverFactory = (ResourceResolverFactory) bundleContext.getService(factoryRef);

***@Reference(target = "(mailservice.label=InternetA)")***

# User Mapper Map:

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

paramMap.put(ResourceResolverFactory.***SUBSERVICE***, "readWriteService");

ResourceResolver resolver = **this**.resourceResolverFactory.getServiceResourceResolver(paramMap);

### OSGi Config node

org.apache.sling.serviceusermapping.impl.ServiceUserMapperImpl.amended-ratereadwriteservice  
 **user.mapping**="[au.com.suncorp.aem.commons-core:readWriteService=writeuser,au.com.suncorp.aem.commons-core-63:readWriteService=writeuser,au.com.suncorp.aem.suncorp-bank-core:readWriteService=writeuser,au.com.suncorp.aem.suncorp-bank-core-63:readWriteService=writeuser]"/>

Version Play:

**private** Node getVersionNode(**final** WorkItem item, Session session)

{

VersionManager versionManager;

Node versionNode = **null**;

**try** {

versionManager = session.getWorkspace().getVersionManager();

VersionHistory versionHistory = versionManager

.getVersionHistory(item.getWorkflow().getWorkflowData().getPayload().toString() +

"/jcr:content");

Node workspaceNode = session.getNode(item.getWorkflow().getWorkflowData().getPayload().toString() +

"/jcr:content");

String reference = **null**;

**if** (workspaceNode.hasProperty("jcr:predecessors") && workspaceNode.getProperty("jcr:predecessors").getValues().length > 0) {

reference = workspaceNode.getProperty("jcr:predecessors").getValues()[0].getString();

}

NodeIterator itr = versionHistory.getAllLinearFrozenNodes();

Node lastVersionNode = **null**;

lastVersionNode = extractActualBaseVersion(reference, itr);

versionNode = session.getNode(lastVersionNode.getPath() + "/jcr:frozenNode/redirectMap.txt/jcr:content");

} **catch** (RepositoryException e) {

***LOGGER***.error(e.getMessage(), e);

}

**return** versionNode;

}

**private** Node extractActualBaseVersion(String reference, NodeIterator itr) **throws** RepositoryException, ValueFormatException,

PathNotFoundException

{

Node lastVersionNode = **null**;

**while** (itr.hasNext()) {

Node frozenNode = (Node) itr.next();

lastVersionNode = frozenNode;

**if** (frozenNode.hasProperty("jcr:uuid") && StringUtils.*isNotEmpty*(reference) &&

frozenNode.getProperty("jcr:uuid").getString().equalsIgnoreCase(reference))

{**break**;}

}

**return** lastVersionNode;

}

# Custom Logger:

### OSGi Config node

org.apache.sling.commons.log.LogManager.factory.config-sc-commons

jcr:primaryType="sling:OsgiConfig"

org.apache.sling.commons.log.file="logs/suncorp-aem-commons.log"

org.apache.sling.commons.log.level="debug"

org.apache.sling.commons.log.names="[au.com.suncorp.insurance.commons]"

org.apache.sling.commons.log.pattern="\{0,date,yyyy-MM-dd HH:mm:ss.SSS} {4} [{3}] {5}"/>

# Rest Services Call

