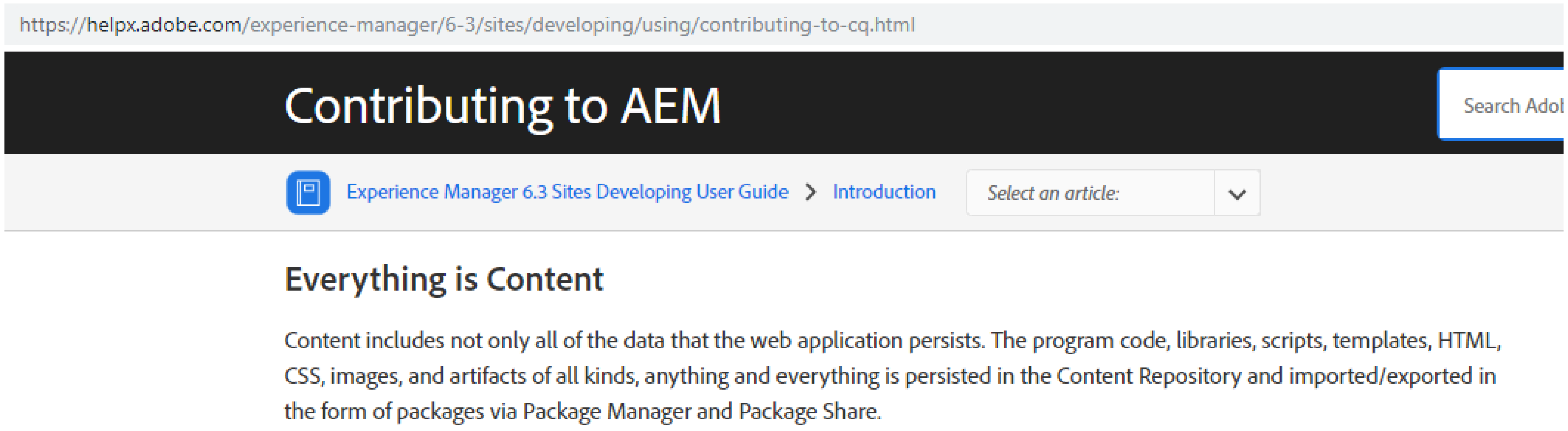
КАК ИЗБАВИТЬСЯ ОТ БАГОВ В КОНТЕНТЕ. ОБЗОР РЕШЕНИЙ

Владислав Минич
Certified AEM Developer в Axamit

What is content?


https://helpx.adobe.com/experience-manager/6-3/sites/developing/using/contributing-to-cq.html

Contributing to AEM

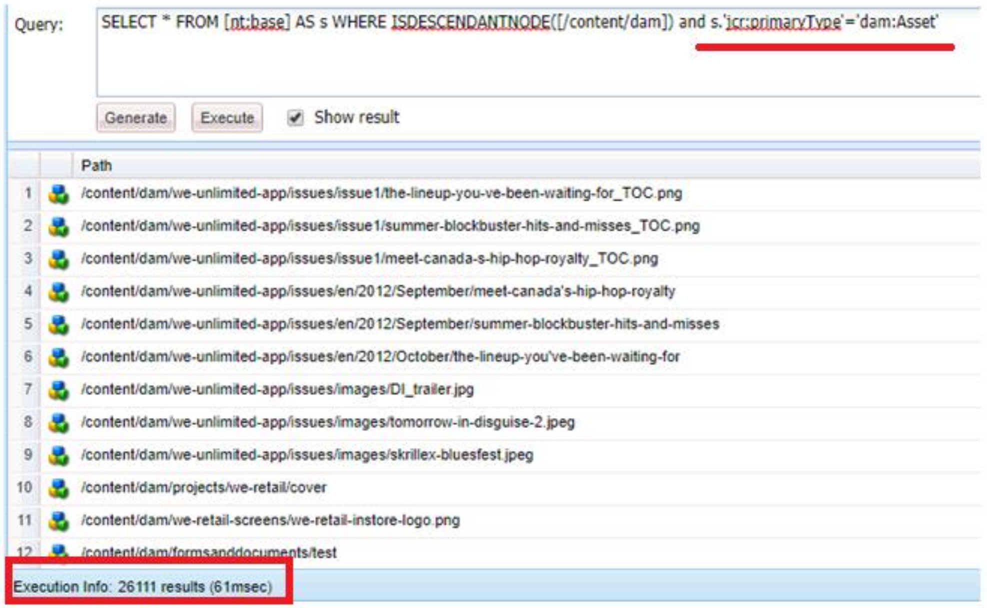
Search Adol
Experience Manager 6.3 Sites Developing User Guide > Introduction
Select an article:

Everything is Content

Content includes not only all of the data that the web application persists. The program code, libraries, scripts, templates, HTML, CSS, images, and artifacts of all kinds, anything and everything is persisted in the Content Repository and imported/exported in the form of packages via Package Manager and Package Share.

Problem

- Change content structure
- Modify/fix content
- Analyze content
- Prepare and package content



Query:

```
SELECT * FROM [nt:base] AS s WHERE ISDESCENDANTNODE([/content/dam]) and s.'jcr:primaryType'='dam:Asset'
```

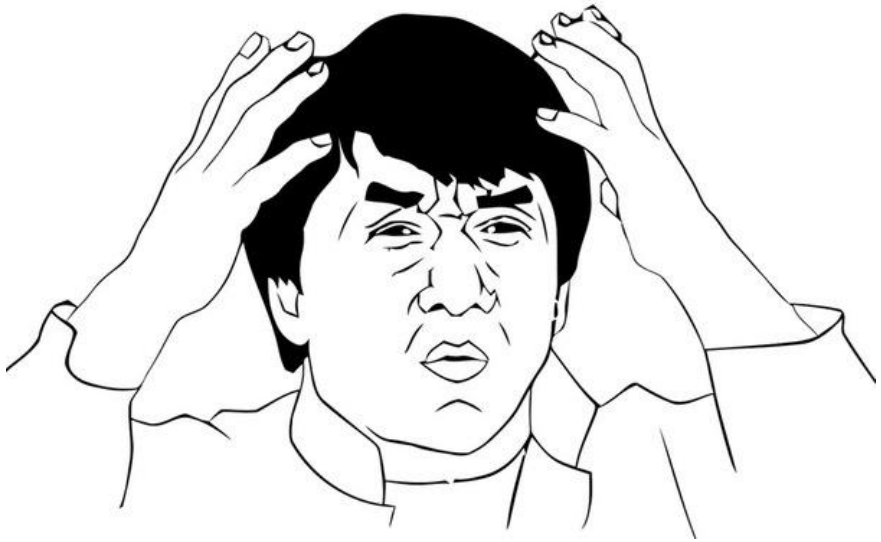
	Path
1	/content/dam/we-unlimited-app/issues/issue1/the-lineup-you-ve-been-waiting-for_TOC.png
2	/content/dam/we-unlimited-app/issues/issue1/summer-blockbuster-hits-and-misses_TOC.png
3	/content/dam/we-unlimited-app/issues/issue1/meet-canada-s-hip-hop-royalty_TOC.png
4	/content/dam/we-unlimited-app/issues/en/2012/September/meet-canada's-hip-hop-royalty
5	/content/dam/we-unlimited-app/issues/en/2012/September/summer-blockbuster-hits-and-misses
6	/content/dam/we-unlimited-app/issues/en/2012/October/the-lineup-you've-been-waiting-for
7	/content/dam/we-unlimited-app/issues/images/DI_trailer.jpg
8	/content/dam/we-unlimited-app/issues/images/tomorrow-in-disguise-2.jpeg
9	/content/dam/we-unlimited-app/issues/images/skrillex-bluesfest.jpeg
10	/content/dam/projects/we-retail/cover
11	/content/dam/we-retail-screens/we-retail-instore-logo.png
12	/content/dam/formsanddocuments/test

Execution Info: 26111 results (61msec)

Solutions

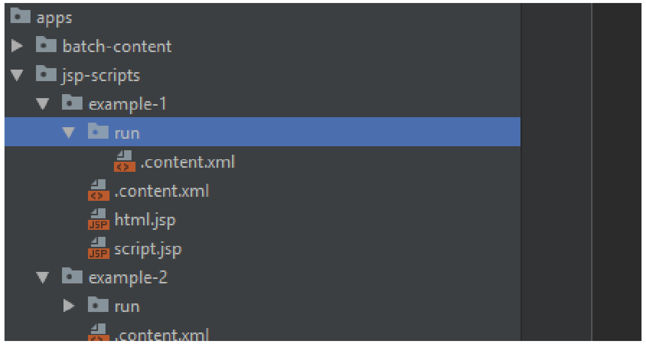
- JSP Scriptlets
- Sling Pipes
- Groovy Console

JSP Scriplets

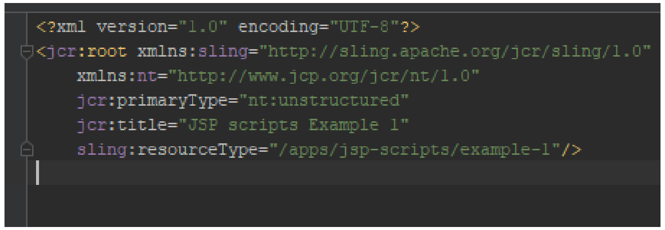


JSP Scriplets

- Store under apps folder with resource type for running
- Run via POST/GET request



```
apps
▶ batch-content
▼ jsp-scripts
    ▼ example-1
        ▼ run
            .content.xml
        .content.xml
        html.jsp
        script.jsp
    ▼ example-2
        ▶ run
        .content.xml
```



```xml
<?xml version="1.0" encoding="UTF-8"?>
<jcr:root xmlns:sling="http://sling.apache.org/jcr/sling/1.0"
    xmlns:nt="http://www.jcp.org/jcr/nt/1.0"
    jcr:primaryType="nt:unstructured"
    jcr:title="JSP scripts Example 1"
    sling:resourceType="/apps/jsp-scripts/example-1"/>
```

JSP Scriplets

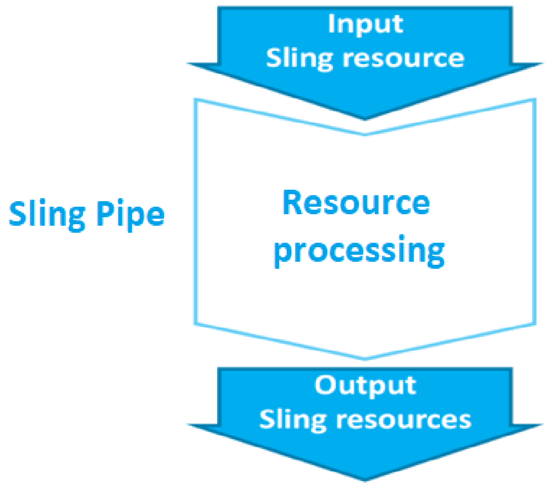
Bindings:

- slingResponse – SlingHttpServletResponse Object
- request – HttpServletRequest Object
- response – HttpServletResponse Object
- resourceResolver – Current requests ResourceResolver
- sling – SlingScriptHelper Object which provides methods for scripts. Most commonly used to fetch an OSGI service reference
- resource – The current resource same as slingRequest.getResource()
- log – SLF4J logger

DEMO

Sling Pipes

- It is a tool where you can load content tree nodes, perform some operations and retrieve an output or modify the nodes
- It provides reusable blocks called pipes which can be configured for any possible operation on content



Sling Pipes

Compatibilities:

- java 8
- slingQuery (3.0.0+)
- jackrabbit api (2.7.5+)

Sling Pipes

A Pipe is basically a jcr node which has several properties :

- **sling:resourceType** = slingPipes/container
- **name** = not to be used as an id and could be a key for output bindings
- **path** = defines pipe’s input. If it isn’t present previous pipes output, it will be used as input for this pipe
- **expr** = expression through which the pipe will execute
- **additionalBinding** = it is a node you can add to set “global” bindings (property=value) in pipe execution
- **additionalScripts** = scripts which can be used as expressions
- **conf** = optional child node, provided to add additional configurations

Sling Pipes

Predefined pipes (some of them):

- Base Pipe – dummy pipe, output=input, sling:resourceType = slingPipes/base
- XPath Pipe – gets resources from the xpath query, sling:resourceType= slingPipes/xpath, expr = xpath query expression
- Container Pipe – used to assemble a sequence of pipe, sling:resourceType = slingPipes/container, conf node is mandatory
- Write Pipe – writes nodes and properties to the input of pipe, sling:resourceType = slingPipes/write
- Move Pipe – used to move input to the target path, sling:resourceType=slingPipes/mv,

  expr = target path
- Remove Pipe – removed the input resource (node or prop) and returns the parent, sling:resourceType = slingPipes/rm

AXAMIT

Sling Pipes

Running:

- by GET/POST request:

request path	path of a pipe configuration resource or a resource of type slingPipes/plumber with a path parameter indicating the pipe configuration resource path.
request method	GET or POST. Note that GET will not work on pipe modifying content (unless you are using a dryRun)
request extension	.json or .csv
request selectors	you can add status to get status on a currently executed pipe

and other optional parameters like dryRun, async, writer etc.

- By Pipe Builder API (from java/groovy):

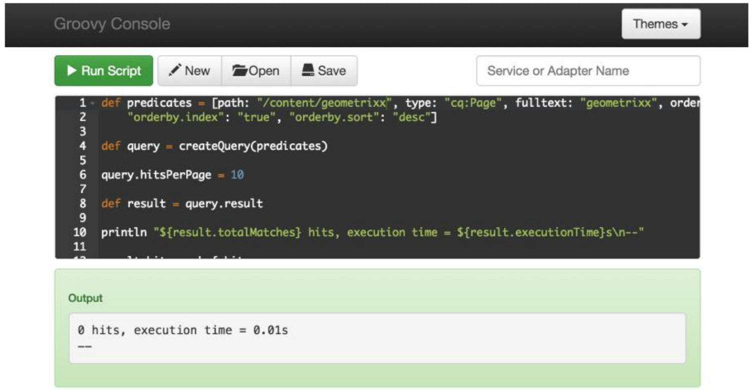
plumber.newPipe(resolver).xpath('//element(*,nt:unstructured)[@sling:resourceType='to/**delete**']").rm().run();

- By Pipe Model

DEMO

Groovy Console

- interface for running Groovy scripts in Adobe Experience Manager
- additional package that provides admin console and API
- run via admin console or POST/GET request or Groovy Console API



```
def predicates = [path: "/content/geometrixx", type: "cq:Page", fulltext: "geometrixx", order
    "orderby.index": "true", "orderby.sort": "desc"]

def query = createQuery(predicates)

query.hitsPerPage = 10

def result = query.result

println "${result.totalMatches} hits, execution time = ${result.executionTime}s\n--"

```

```
0 hits, execution time = 0.01s
--
```

Groovy Console

Requirements:

- AEM instance running
- Maven 3.x

Compatibilities :

Groovy Console Version(s)	**AEM Version**
13.x.x, 12.x.x	6.4
11.x.x	6.3
10.x.x, 9.x.x	6.2
8.x.x	6.1
7.x.x	6.0
6.x.x, 5.x.x	5.6 (CQ)
3.x.x	5.5, 5.4 (CQ)

AXAMIT

Groovy Console

The binding variables, which are listed below, are available for use in all scripts:

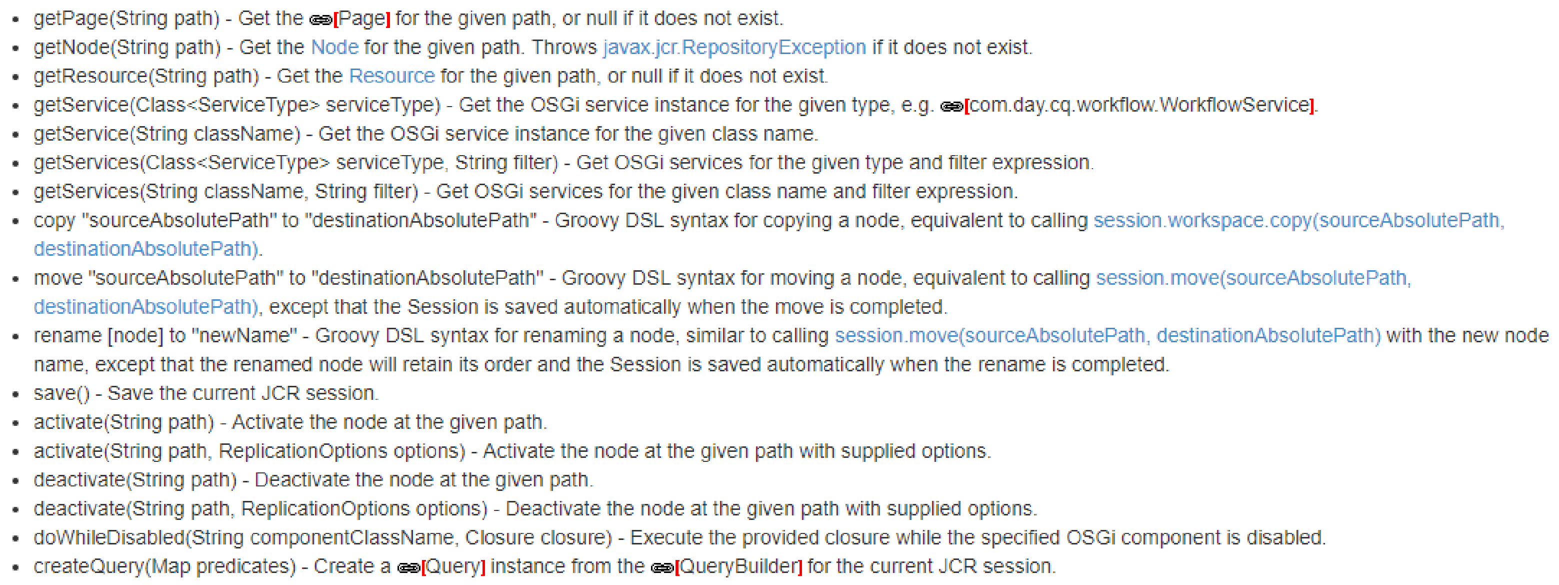
- session - javax.jcr.Session
- pageManager - [com.day.cq.wcm.api.PageManager]
- resourceResolver - org.apache.sling.api.resource.ResourceResolver
- slingRequest - org.apache.sling.api.SlingHttpServletRequest
- queryBuilder - [com.day.cq.search.QueryBuilder]
- bundleContext - org.osgi.framework.BundleContext
- log - org.slf4j.Logger

Additional packages, which are imported into all scripts:

- com.day.cq.search
- com.day.cq.tagging
- com.day.cq.wcm.api
- com.day.cq.replication
- javax.jcr
- org.apache.sling.api
- org.apache.sling.api.resource

Groovy Console

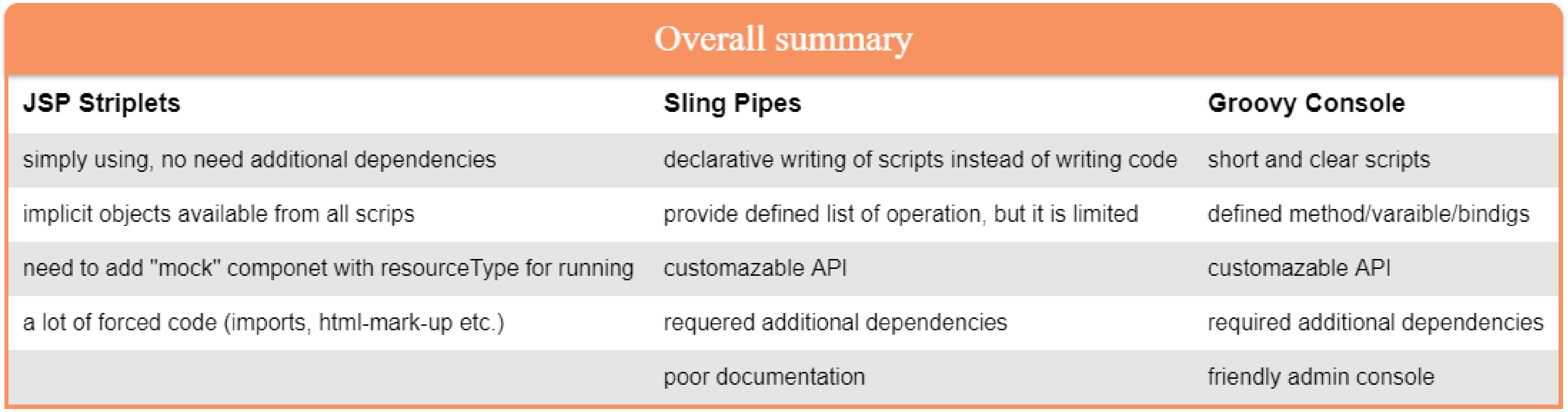
The methods, which are listed below, are available for use in all scripts:



- getPage(String path) - Get the [Page] for the given path, or null if it does not exist.
- getNode(String path) - Get the Node for the given path. Throws javax.jcr.RepositoryException if it does not exist.
- getResource(String path) - Get the Resource for the given path, or null if it does not exist.
- getService(Class<ServiceType> serviceType) - Get the OSGi service instance for the given type, e.g. [com.day.cq.workflow.WorkflowService].
- getService(String className) - Get the OSGi service instance for the given class name.
- getServices(Class<ServiceType> serviceType, String filter) - Get OSGi services for the given type and filter expression.
- getServices(String className, String filter) - Get OSGi services for the given class name and filter expression.
- copy "sourceAbsolutePath" to "destinationAbsolutePath" - Groovy DSL syntax for copying a node, equivalent to calling session.workspace.copy(sourceAbsolutePath, destinationAbsolutePath).
- move "sourceAbsolutePath" to "destinationAbsolutePath" - Groovy DSL syntax for moving a node, equivalent to calling session.move(sourceAbsolutePath, destinationAbsolutePath), except that the Session is saved automatically when the move is completed.
- rename [node] to "newName" - Groovy DSL syntax for renaming a node, similar to calling session.move(sourceAbsolutePath, destinationAbsolutePath) with the new node name, except that the renamed node will retain its order and the Session is saved automatically when the rename is completed.
- save() - Save the current JCR session.
- activate(String path) - Activate the node at the given path.
- activate(String path, ReplicationOptions options) - Activate the node at the given path with supplied options.
- deactivate(String path) - Deactivate the node at the given path.
- deactivate(String path, ReplicationOptions options) - Deactivate the node at the given path with supplied options.
- doWhileDisabled(String componentClassName, Closure closure) - Execute the provided closure while the specified OSGi component is disabled.
- createQuery(Map predicates) - Create a [Query] instance from the the [QueryBuilder] for the current JCR session.

DEMO

Summary



Overall summary

JSP Striplets	Sling Pipes	Groovy Console
simply using, no need additional dependencies	declarative writing of scripts instead of writing code	short and clear scripts
implicit objects available from all scrips	provide defined list of operation, but it is limited	defined method/varaible/bindigs
need to add "mock" componet with resourceType for running	customazable API	customazable API
a lot of forced code (imports, html-mark-up etc.)	requered additional dependencies	required additional dependencies
	poor documentation	friendly admin console

Links

Sling Pipes:

- Documentation - https://sling.apache.org/documentation/bundles/sling-pipes.html
- Articles:
  - https://hashimkhan.in/2016/09/14/sling-pipes/
  - http://blogs.adobe.com/contentmanagement/2017/05/01/sling-pipes-a-rockstar-way-to-deal-with-jcr/

Groovy Console:

- Groovy Console Package- https://github.com/OlsonDigital/aem-groovy-console
- Articles:
  - https://www.axamit.com/blog/adobe/groovyconsole
  - https://hashimkhan.in/aem-adobecq5-code-templates/groovy-script/

JSP:

- Documentation - https://docs.oracle.com/javaee/5/tutorial/doc/bnajo.html
- JSP Implicit objects - http://adobeaemtips.blogspot.com/2014/05/cq-defined-objects.html

Q & A