

adaptTo()

APACHE SLING & FRIENDS TECH MEETUP
BERLIN, 26-28 SEPTEMBER 2016

Sling Pipes introduction

N. Peltier, Adobe





The problem



Widespread content

```
/jcr:root/content/geometrixx/en//element(*,nt:unstructured)[@sling:resourceType='foundation/components/text']
```

60  /content/geometrixx/en/services/certification/jcr:content/par/text_1

61  /content/geometrixx/en/services/certification/jcr:content/par/text

Execution Info: 65 results (3msec)

4	jcr:lastModifiedBy	String	admin
5	jcr:primaryType	Name	nt:unstructured
6	sling:resourceType	String	foundation/components/text
7	text	String	<p>The classic for the discerning tria
8	textIsRich	String	true



- New text component, not compatible with content,
- Text component bug fix not compatible with content
- Content Migrations





POST.jsp



stuff that imports

```
<%@page session="false"
contentType="text/html"
pageEncoding="utf-8"
import="java.io.PrintWriter,
    java.util.Iterator,
    java.util.ArrayList,
    javax.jcr.Node,
    javax.jcr.RepositoryException,
    javax.jcr.Session,
    javax.jcr.Value,
    javax.jcr.util.TraversingItemVisitor,
    org.apache.jackrabbit.util.Text,
    org.apache.sling.api.resource.Resource,
    org.apache.sling.api.resource.ResourceResolver,
    org.apache.sling.api.scripting.SlingScriptHelper,
    org.slf4j.Logger,
    org.apache.jackrabbit.value.BooleanValue,
    org.apache.jackrabbit.value.StringValue,
    org.slf4j.LoggerFactory,
    com.day.cq.commons.LabeledResource,
    org.apache.sling.commons.json.JSONArray,
    org.apache.sling.commons.json.JSONException,
    org.apache.sling.commons.json.JSONObject,
    com.day.cq.dam.api.Asset,
    com.acme.cq.wcm.commons.category.CategoryService,
    com.acme.cq.wcm.commons.models.TraversingCategoryVisitor,
    com.day.cq.wcm.api.Page" %><%
%><%@taglib prefix="sling" uri="http://sling.apache.org/taglibs/sling/1.0" %><%
%><%@taglib prefix="cq" uri="http://www.day.com/taglibs/cq/1.0" %><%
%><cq:defineObjects /><%
```



stuff that defines

```
final String PARAM_PATH = "rootpath";
final String PARAM_COMMIT = "commit";
final String TYPE_BOOLEAN = "Boolean";
final String TYPE_DATE = "Date";
final String TYPE_ARRAY = "[]";
final String PARAM_UPDATE = "update";
final String PARAM_NODE_TYPE = "nodetype";
final String PARAM_RESOURCE_TYPE = "resourceType";
final String SLING_RESOURCE_TYPE = "sling:resourceType";
final String JSON_PROP_NAME = "propertyname";
final String JSON_PROP_TYPE = "propertytype";
final String JSON_PROP_VALUE = "propertyvalue";
final String JSON_COND_NAME = "namecondition";
final String JSON_COND_VALUE = "valuecondition";

ResourceResolver resolver = slingRequest.getResourceResolver();
Session session = resolver.adaptTo(Session.class);
String path = request.getParameter(PARAM_PATH);
final String nodeType = request.getParameter(PARAM_NODE_TYPE);
final String resourceType = request.getParameter(PARAM_RESOURCE_TYPE);
final NodeSavingCounter counterSave = new NodeSavingCounter();
final boolean commit = "true".equals(request.getParameter(PARAM_COMMIT));
final JSONArray updateArray = new JSONArray(request.getParameter(PARAM_UPDATE));
```



stuff that designs & monitors

```
%><html><head>
<style type="text/css">
  div {
    font-family:arial,tahoma,Helvetica,sans-serif;
    font-size:11px;
    white-space:nowrap;
  }
  .action {
    display: inline;
    width: 120px;
    float: left;
  }
  .error {
    color: red;
    font-weight: bold;
  }
  .title {
    display: inline;
    width: 150px;
    float: left;
    margin: 0 8px 0 0;
    overflow: hidden;
  }
  .activate {
  }
  .ignore {
    color: #888888;
  }
  .cf {
    color: #888888;
  }
  .path {
    display: inline;
    width: 100%;
  }
```

```
j|jspOut.printf("<div class=\"error\">" + e.getLocalizedMessage() + "</div>");
```

```
j|jspOut.printf("Setting " + node.getPath() + "." + propertyName + " = " + value);
```



Stuff that searches

<%

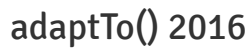
```
final PrintWriter jspOut = new PrintWriter(out);
final TimedFlush timedFlush = new TimedFlush(jspOut);
try {
    Node rootNode = (Node)session.getItem(path);
    if (rootNode != null) {
        rootNode.accept(new TraversingItemVisitor.Default(){
            protected void entering(Node node, int level) {
                try {
                    if (node.isNodeType(nodeType)) {
                        if (node.hasProperty(SLING_RESOURCE_TYPE) &&
                            node.getProperty(SLING_RESOURCE_TYPE).getString().compareTo(resourceType) == 0) {
```



Stuff that writes

```
    objValue = update;
}
if (objValue != null) {
    counterSave.count();
    String propertyType = update.optString(JSON_PROP_TYPE);
    String value = objValue.getString(JSON_PROP_VALUE);
    jspOut.printf("Setting " + node.getPath() + "." + propertyName + " = " + value);
    Value returnValue = null;
    if (propertyType == null) {
        returnValue = new StringValue(value);
    } else if (propertyType.compareTo(TYPE_BOOLEAN) == 0 ) {
        returnValue = new BooleanValue(Boolean.valueOf(value));
    }
    node.setProperty(propertyName, returnValue);
    if (commit && counterSave.needSaving()) {
        jspOut.printf("saving 1000 nodes");
        timedFlush.flush();
        node.getSession().save();
    }
} else {
    jspOut.printf("no condition found for the node %s",node.getPath());
    timedFlush.flush();
}
}
```



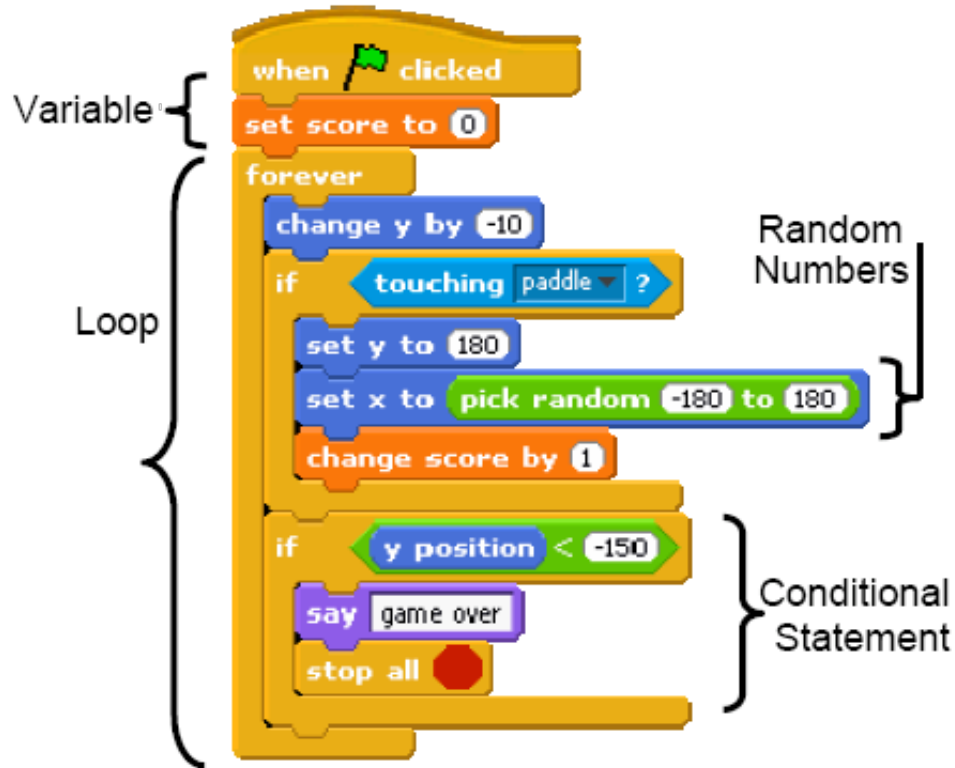


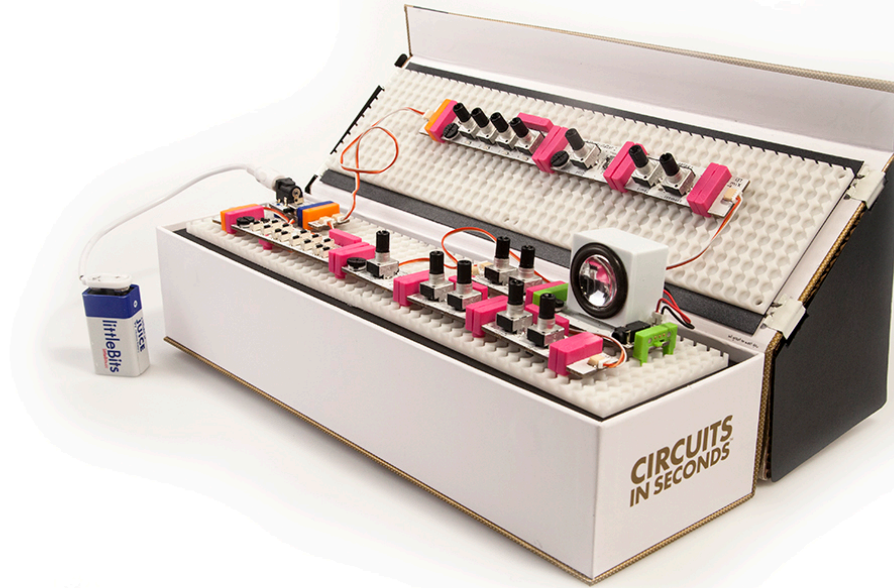
Make stuff simple & reusable



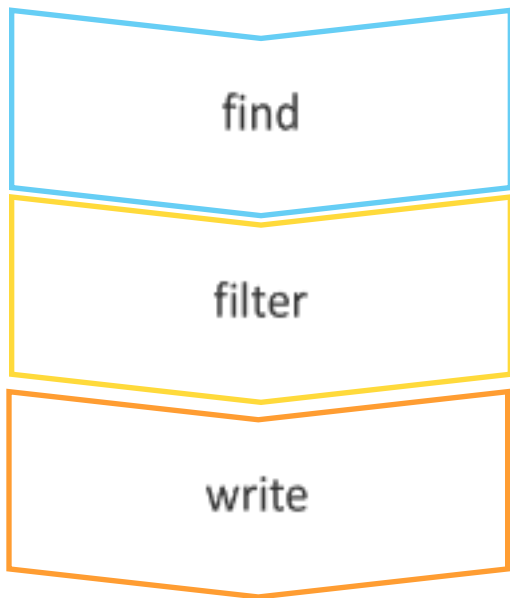
Lego mindstorms ©







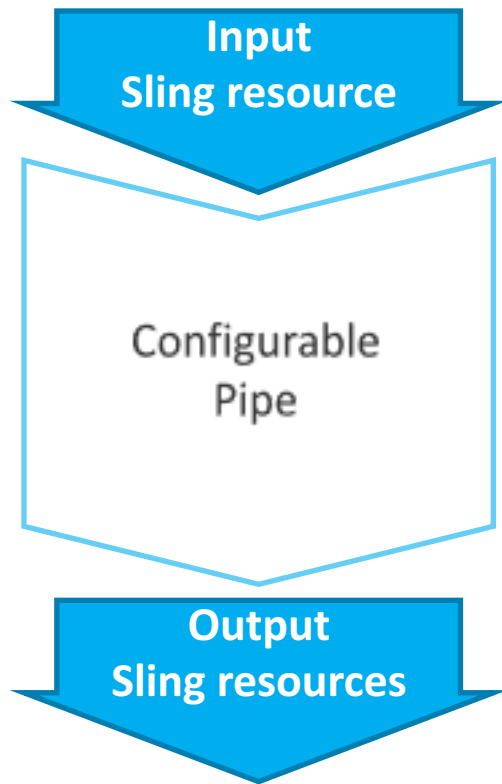
Apache sling pipes



- Reusable sling stuff
 - Proven
 - lightweight
 - configurable
 - chainable (in a clean way)
 - executable



Apache sling pipes



Basic Usage



How to configure a pipe

xpath

```
sling:resourceType: slingPipes/xpath
```

```
expr: /jcr:root/content/geometrixx//element(*,cq:Page)
```



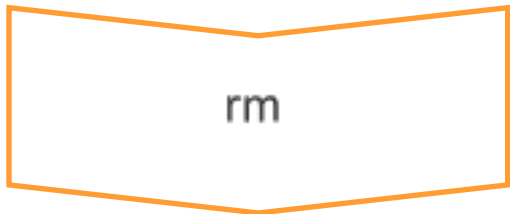
Configurations can be more complex



```
sling:resourceType: slingPipes/filter
conf: {
  jcr:content:{
    sling:resourceType:geometrixx/event
    par: {
      slingPipesFilter_noChildren:true
    }
  }
}
```

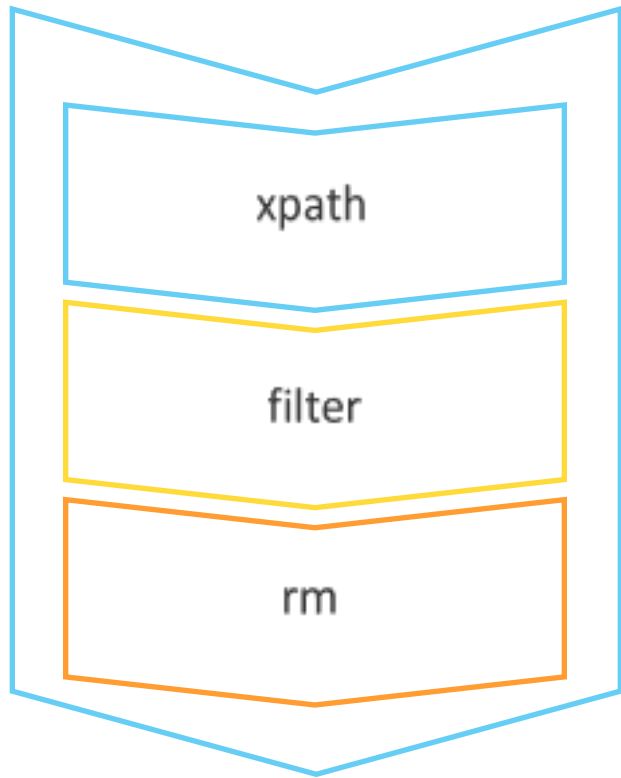


Configurations can be simpler



`sling:resourceType: slingPipes/rm`

One specific pipe : container



```
sling:resourceType: slingPipes/container  
conf: {  
  jcr:primaryType: sling:OrderedFolder  
  xpath: {...},  
  filter: {...},  
  rm: {...}  
}
```

How to execute a pipe

```
GET /some/readonly/containerpipe.json  
POST /some/write/containerpipe.json
```

```
@Reference
```

```
Plumber plumber;
```

```
...
```

```
Pipe pipe = plumber.getPipe(containerResource);
```

```
Iterator<Resource> children = pipe.getOutput();
```

Demo

Expressions and Bindings

- ECMA like expressions can be used in most configurations

```
This ${'prop' + 'erty'}: is ${2 > 1 ? 'use' : 'pain' }less
```

- Expressions become useful with pipe bindings

```
//assuming we have a « user » and a « profile » pipe above  
displayName:${ (profile.gender ? 'Ms' : 'Mr') + profile.first + ' ' +  
profile.last}  
path:${path.user}/profile
```

Demo

Conclusions

- Think about it for other usages:
 - Json/JCR based reports,
 - List based components
- Give it a try and contribute
 - To make it simpler,
 - To make it more powerful,
 - Or to replace it 😊

Thank you!

- <https://github.com/npeltier/sling-pipes-client> (using `./init.sh` will install pipes, dependencies, sample for free!)
- <https://sling.apache.org/documentation/bundles/sling-pipes.html>
- <https://github.com/apache/sling/tree/trunk/contrib/extensions/sling-pipes>
- <https://camel.apache.org/pipes-and-filters.html>

Nicolas Peltier (@npeltier)