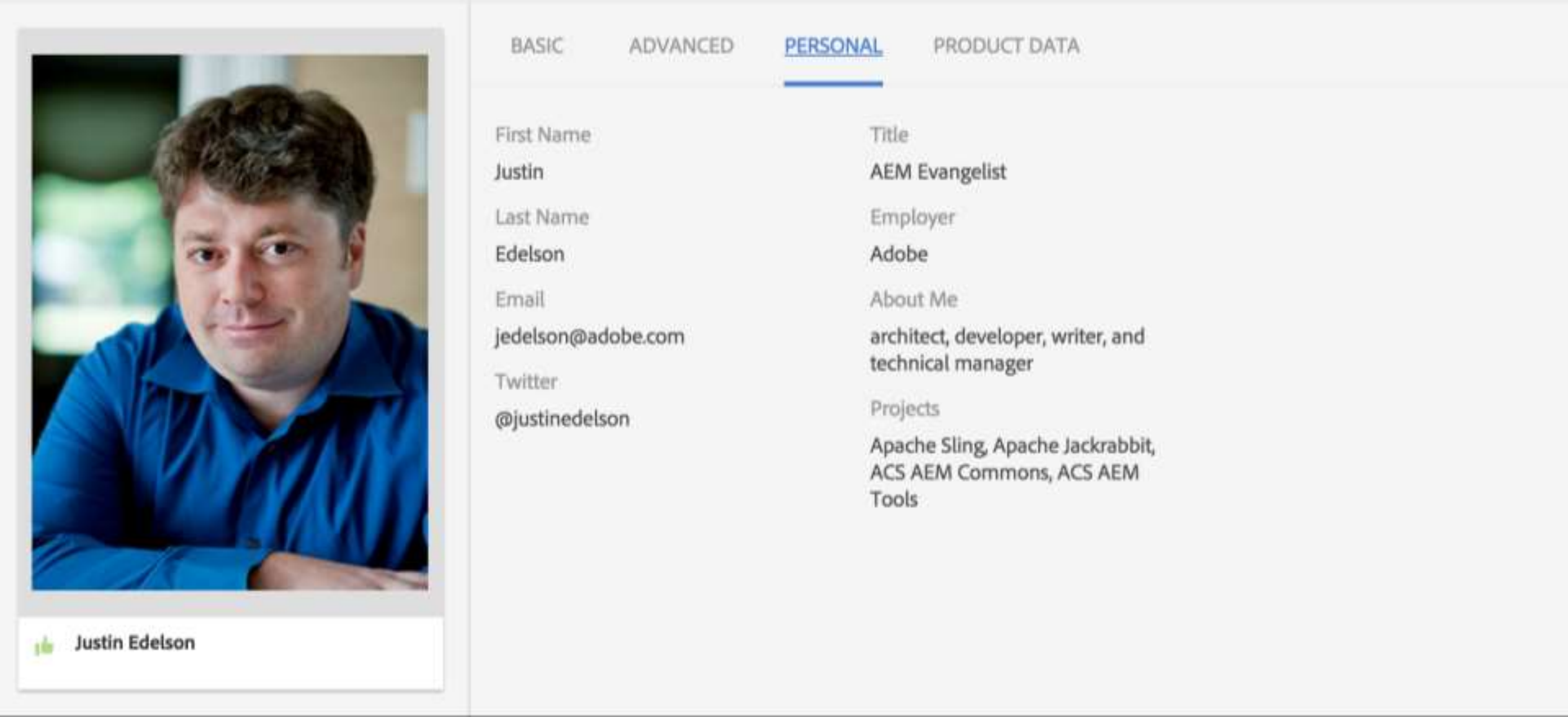
Mastering the Sling Rewriter

Justin Edelson
AEM Evangelist



BASIC ADVANCED PERSONAL PRODUCT DATA

First Name	Title
Justin	AEM Evangelist
Last Name	Employer
Edelson	Adobe
Email	About Me
jedelson@adobe.com	architect, developer, writer, and technical manager
Twitter	Projects
@justinedelson	Apache Sling, Apache Jackrabbit, ACS AEM Commons, ACS AEM Tools

Justin Edelson

What is the Sling Rewriter?

- Sling Rewriter an Apache Sling module included in AEM.
- Performs transformations of rendered content (typically HTML)
  - Doesn’t know anything about your components
- Pipeline-orientated
- Based on Simple API for XML (SAX)

© 2015 Adobe Systems Incorporated. All Rights Reserved.

Is Sling Rewriter related to mod_rewrite?

No.

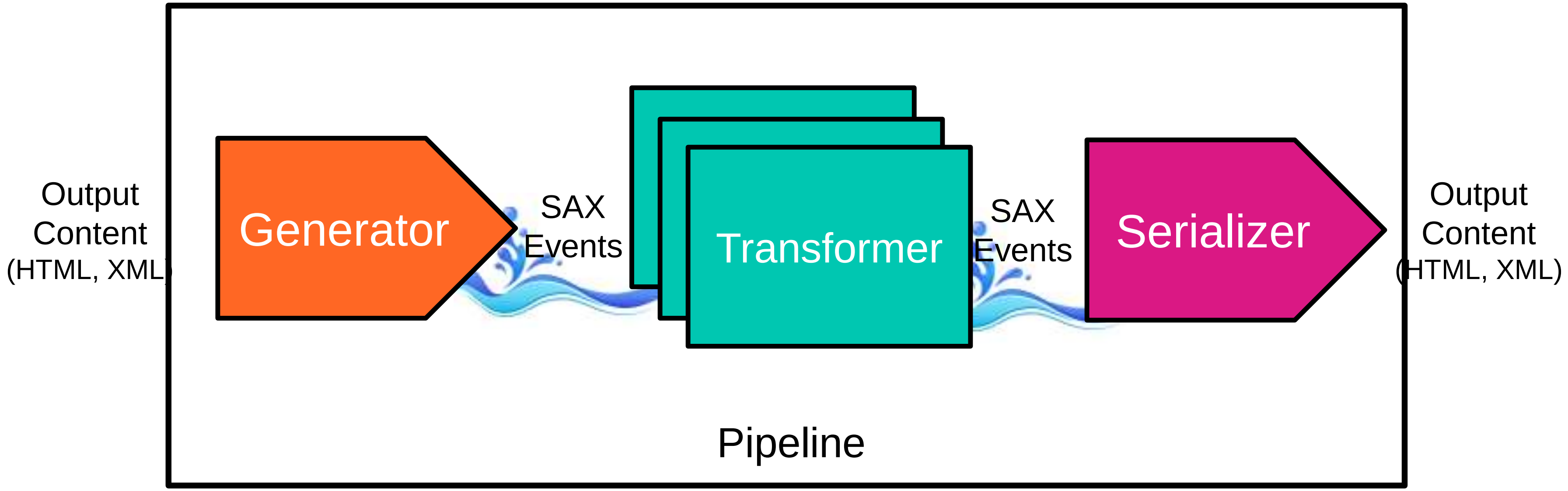
Sling Rewriter is about rewriting the (HTML) output.

mod_rewrite is about rewriting the requesting URL.

There is an Apache module, mod_ext_filter, which can be used for output rewriting.

© 2015 Adobe Systems Incorporated. All Rights Reserved.

Rewriter Pipeline



© 2015 Adobe Systems Incorporated. All Rights Reserved.

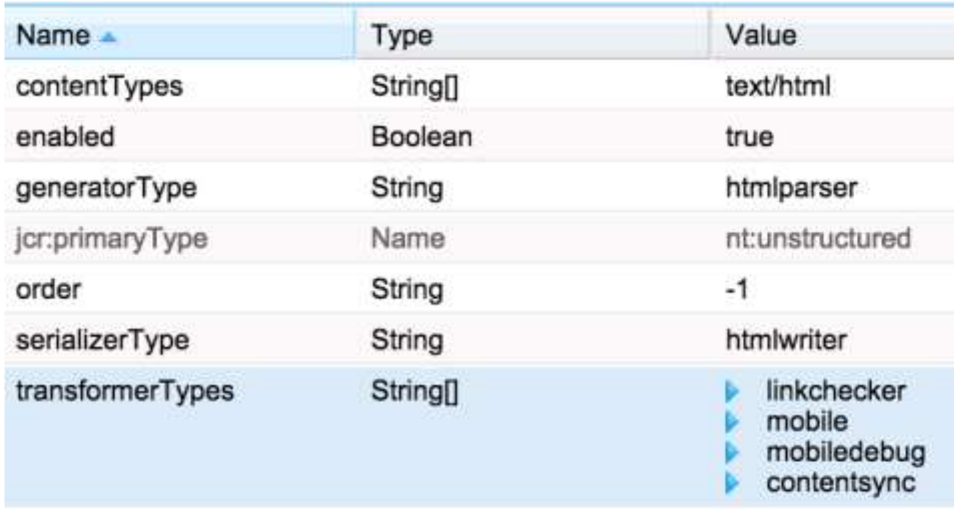
Rewriter Pipeline Configurations

Stored in the repository.

Always /apps/SOMETHING/config/rewriter

Defines **when** the pipeline is executed

Defines **how** the pipeline is composed



Name	Type	Value
contentTypes	String[]	text/html
enabled	Boolean	true
generatorType	String	htmlparser
jcr:primaryType	Name	nt:unstructured
order	String	-1
serializerType	String	htmlwriter
transformerTypes	String[]	linkchecker mobile mobiledebug contentsync

Enablement Options

All are optional

contentTypes – Response Content Type (e.g. text/html)

extensions – Request Extension (e.g. html)

resourceTypes – Resolved Resource Type

selectors – Request selectors (soon; not in AEM 6.1)

paths – Path prefixes

enabled – true/false

order – highest wins

© 2015 Adobe Systems Incorporated. All Rights Reserved.

Pipeline Options

generatorType

transformerTypes

serializerType

```
@Component @Service
@Property(name = "pipeline.type”,
    value = "xml-generator")
public class XMLParserGeneratorFactory
    implements GeneratorFactory { }

@Component @Service
@Property(name = "pipeline.type",
    value = "versioned-clientlibs")
public class VersionedClientlibsTransformerFactory
    implements TransformerFactory { }

@Component @Service
@Property(name = "pipeline.type”,
    value = "xml-serializer")
public class PlainXMLSerializerFactory
    implements SerializerFactory { }
```

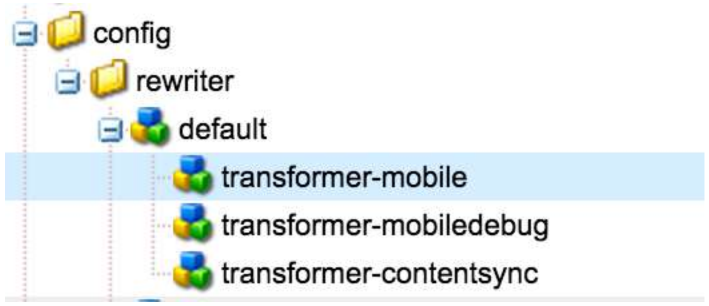
© 2015 Adobe Systems Incorporated. All Rights Reserved.

Pipeline Element Configuration

Nodes named <component-type>-<component-name>

Only one defined property - component-optional

Everything else is up to the particular component to define



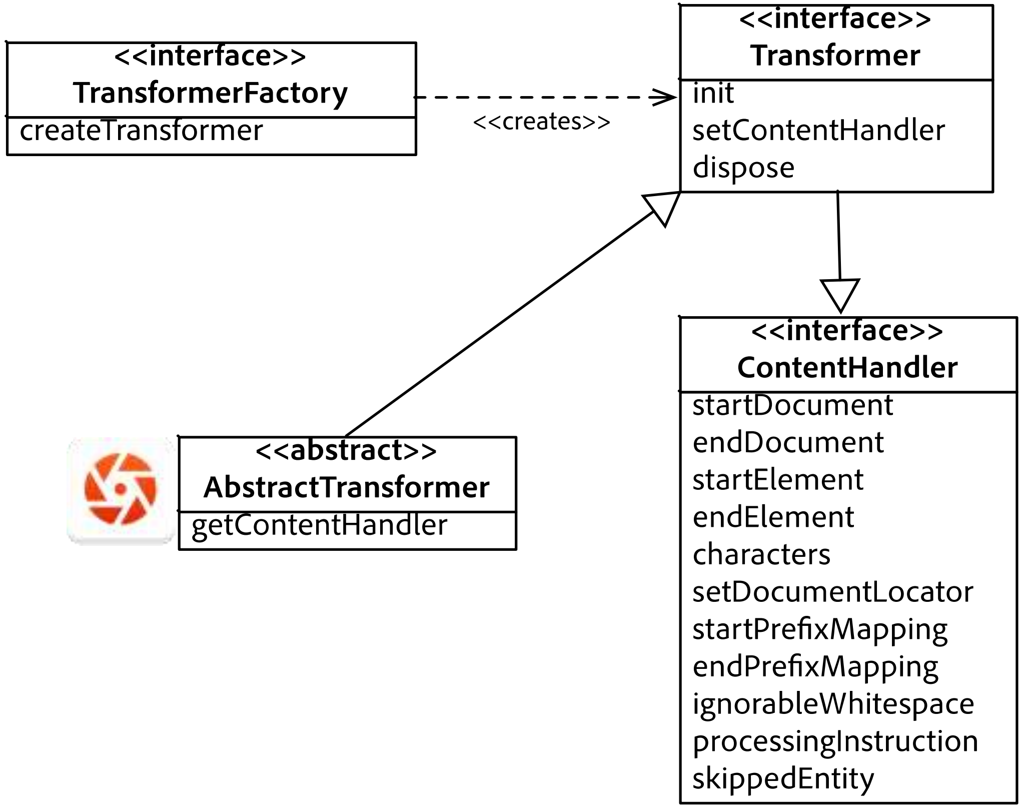
Creating your own Transformer

Implement two interfaces

```
@Component @Service
@Property(name = "pipeline.type",
        value = ”mytrans")
public class MyTransformerFactory
        implements TransformerFactory {

    public Transformer createTransformer() {
        return new MyTransformer();
    }

    private class MyTransformer
         implements Transformer {
         …
    }
}
```



© 2015 Adobe Systems Incorporated. All Rights Reserved.

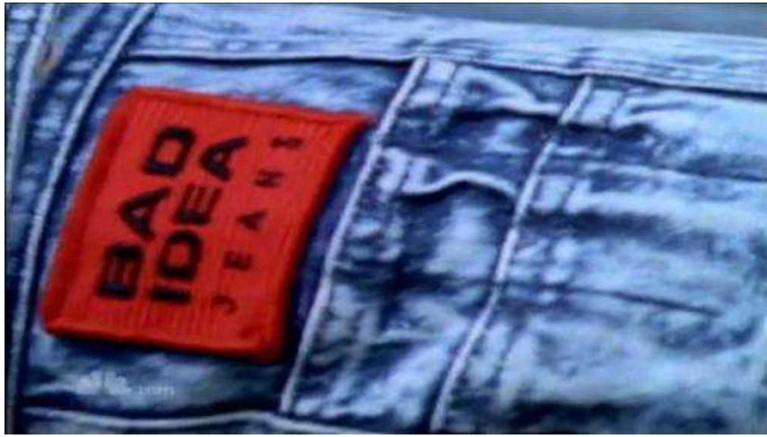
“global” Transformers

Transformers can be declared as global.

```
@Component @Service
@Property(name="pipeline.mode",
     value="global")
public class MyEvilTransformer
    implements TransformerFactory {}
```

This is a bad idea.

Don’t do it.



Creating your own Transformer

```
public void startElement(String namespaceURI, String localName,
         String qName, Attributes atts) {
    nextHandler.startElement(namespaceURI, localName,
       qName, rebuildAttributes(localName, atts));
}

private Attributes rebuildAttributes(String elementName, Attributes attrs) {
    if (“a”.equals(elementName)) {
        AttributesImpl newAttrs = new AttributesImpl(attrs);
        …
        return newAttrs;
    } else {
        return attrs;
    }
}
```

© 2015 Adobe Systems Incorporated. All Rights Reserved.

Transformer Use Case
Wrapping YouTube Embedded IFrames

```
<iframe width="420" height="315"
    src="https://www.youtube.com/embed/QfvFWSQQ_0M"
    frameborder="0" allowfullscreen>
</iframe>
```

```
<div class="youtube-container">
  <iframe width="420" height="315"
    src="https://www.youtube.com/embed/QfvFWSQQ_0M"
    frameborder="0" allowfullscreen class="youtube-wrapped">
  </iframe>
</div>
```

© 2015 Adobe Systems Incorporated. All Rights Reserved.

Adding YouTube Wrapper

```
public void startElement(String uri, String localName, String qName, Attributes atts) {
    if ("iframe".equals(localName) && needsWrapping(atts)) {
        startWrapper();
        needToUnwrap = true;
        AttributesImpl newAtts = new AttributesImpl();
        newAtts.setAttributes(atts);
        newAtts.addAttribute("", "class", "class", "CDATA", "youtube-wrapped");
        nextHandler.startElement(uri, localName, qName, newAtts);
    } else {
        nextHandler.startElement(uri, localName, qName, atts);
    }
}

public void endElement(String uri, String localName, String qName) {
    nextHandler.endElement(uri, localName, qName);
    if (needToUnwrap && localName.equals("iframe")) {
        endWrapper();
        needToUnwrap = false;
    }
}
```

© 2015 Adobe Systems Incorporated. All Rights Reserved.

Adding YouTube Wrapper

```
boolean needsWrapping(Attributes atts) {
    final String src = atts.getValue("src");
    if (src == null) {
        return false;
    }
    if (src.contains("youtube")) {
        final String classAttr = atts.getValue("class");
        if (classAttr == null) {
            return true;
        } else if (classAttr.indexOf("youtube-wrapped") > -1) {
           return false;
        } else {
            return true;
        }
    }
    return false;
}
```

© 2015 Adobe Systems Incorporated. All Rights Reserved.

Adding YouTube Wrapper

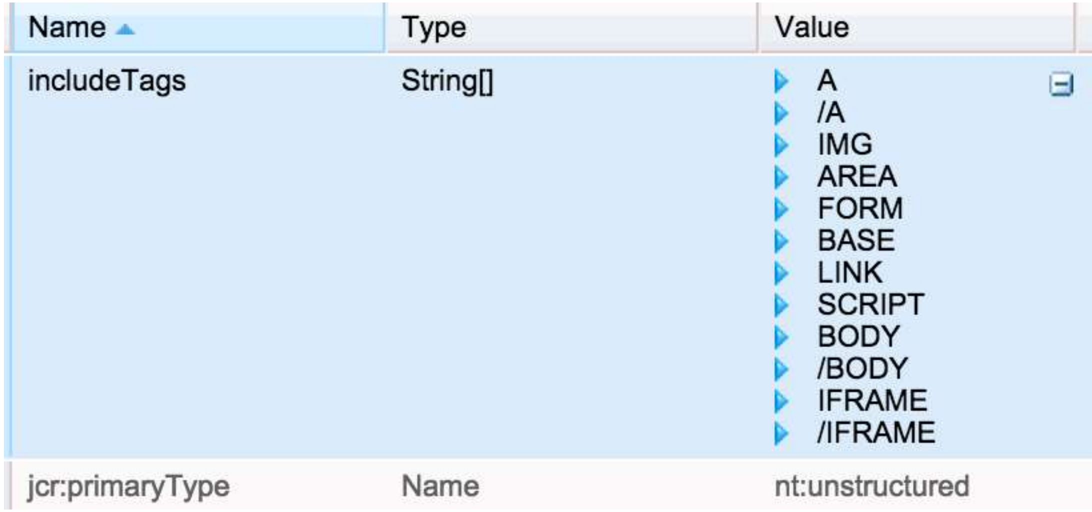
```
void endWrapper(){
    nextHandler.endElement("", "div", "div");
}

void startWrapper(){
    AttributesImpl newAtts = new AttributesImpl();
    newAtts.addAttribute("", "class", "class", "CDATA”,
        "youtube-container");
    nextHandler.startElement("", "div", "div", newAtts);
}
```

© 2015 Adobe Systems Incorporated. All Rights Reserved.

BUT…

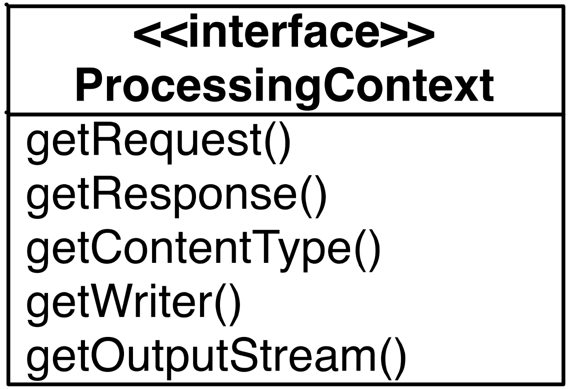
- AEM’s HTML parser ignores <iframe> by default.
- Need to adjust the configuration
  - With a node named generator-htmlparser

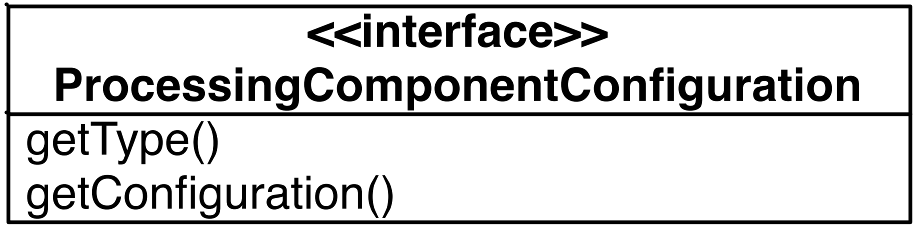


Name	Type	Value
includeTags	String[]	A /A IMG AREA FORM BASE LINK SCRIPT BODY /BODY IFRAME /IFRAME
jcr:primaryType	Name	nt:unstructured

© 2015 Adobe Systems Incorporated. All Rights Reserved.

The init() method





© 2015 Adobe Systems Incorporated. All Rights Reserved.

The characters() method

The characters() method is hard…

```
<div>foo</div>

characters(['f','o','o'], 0, 2);

characters(['f'], 0, 1);
characters(['o'], 0, 1);
characters(['o'], 0, 1);

characters(['>','f'], 1, 2);
characters(['o', 'o'], 0, 2);
```

“SAX parsers may return all contiguous character data in a single chunk, or they may split it into several chunks.“

© 2015 Adobe Systems Incorporated. All Rights Reserved.

The characters() method

For simple manipulations:

```
public void characters(char ch[],int start, int length) {
    String str = new String(ch, start, length);
    str = str.toUpperCase();
    nextHandler.characters(str.toCharArray(), 0, length);
}
```

© 2015 Adobe Systems Incorporated. All Rights Reserved.

The characters() method

For complex manipulations:

```
public void startElement(String uri, String localName, String qName,
    Attributes atts) {
    if (isSpecialElement(localName, atts) {
        collectingCharacters = true;
        buffer = new StringBuilder();
    }
    nextHandler.startElement(uri, localName, qName, atts);
}

public void characters(char[] ch, int start, int length) {
    if (collectingCharacters) {
        buffer.append(ch, start, length);
    } else {
        nextHandler.characters(ch, start, length);
    }
}
```

© 2015 Adobe Systems Incorporated. All Rights Reserved.

The characters() method

```
public void endElement(String uri, String localName, String qName) {
    if (collectingCharacters) {
        String output = manipulate(buffer);
        nextHandler.characters(output.toCharArray(), 0, output.length);
        collectingCharacters = false;
        buffer = null;
    }
    nextHandler.endElement(uri, localName, qName);
}
```

© 2015 Adobe Systems Incorporated. All Rights Reserved.

Rewriting Other Things

- Rewriting XML?
  - Check out xml-generator and xml-serializer in ACS AEM Commons
- Rewriting JSON?
  - Well…

© 2015 Adobe Systems Incorporated. All Rights Reserved.

JSON Array Generator

```
public class JsonArrayGenerator implements Generator {
  public void finished() throws IOException, SAXException {
    try {
        JSONArray array = new JSONArray(writer.toString());
        contentHandler.startDocument();
        for (int i = 0; i < array.length(); i++) {
            final JSONObject obj = array.getJSONObject(i);
            contentHandler.startElement(null, obj.toString(), null, null);
        }
        contentHandler.endDocument();
    } catch (JSONException e) {
        throw new SAXException("Unable to parse JSON Array", e);
    }
  }
}
```

SAX Smuggling

JSON Array Content Handler

```
public abstract class JsonArrayContentHandler implements ContentHandler {
    protected abstract void endArray();
    protected abstract void startArray()
    protected abstract void handleObject(JSONObject obj);

    public void startDocument() { startArray();  }
    public void startElement(String arg0, String arg1, String arg2, Attributes arg3)
            throws SAXException {
        try {
            JSONObject obj = new JSONObject(arg1);
            handleObject(obj);
        } catch (JSONException e) {
            throw new SAXException("Unable to parse JSON string", e);
        }
    }
    public void endDocument() { endArray(); }
}
```

© 2015 Adobe Systems Incorporated. All Rights Reserved.

JSON Object Rewriting

```
protected void handleObject(JSONObject obj) {
    obj.put("text", obj.get("name"));
    contentHandler.startElement(null,
       obj.toString(), null, null);
}
```

Adobe Experience Manager Web Console
Sling Rewriter

Troubleshooting

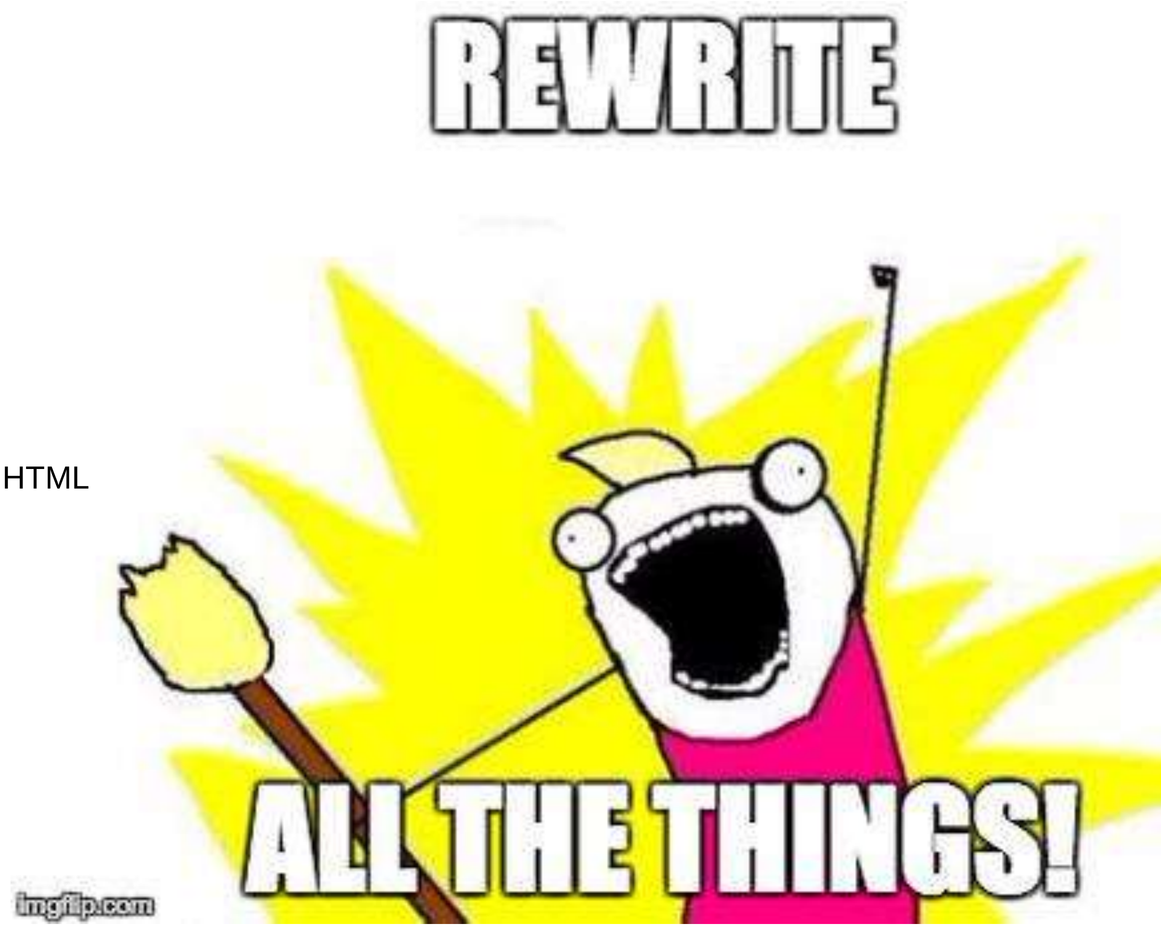
- WebConsole Configuration Printer
- Recent Requests Web Console

```
211 LOG Found processor for post processing ProcessorConfiguration:
{
contentTypes=[text/html],order=-1, active=true, valid=true, processErrorResponse=true,
pipeline=(generator=Config(type=htmlparser, config=JcrPropertyMap
[node=Node[NodeDelegate{tree=/libs/cq/config/rewriter/default/generator-htmlparser: {
jcr:primaryType = nt:unstructured, includeTags = [A, /A, IMG, AREA, FORM, BASE, LINK, SCRIPT,
BODY, /BODY, IFRAME, /IFRAME]}}], values={jcr:primaryType=nt:unstructured,
includeTags=[Ljava.lang.String;@3b9d3802}]), transformers=(Config(type=linkchecker, config={}),
Config(type=mobile, config=JcrPropertyMap
[node=Node[NodeDelegate{tree=/libs/cq/config/rewriter/default/transformer-mobile: {
jcr:primaryType = nt:unstructured, component-optional = true}}],
values={jcr:primaryType=nt:unstructured, component-optional=true}]), Config(type=mobiledebug,
config=JcrPropertyMap
[node=Node[NodeDelegate{tree=/libs/cq/config/rewriter/default/transformer-mobiledebug: {
jcr:primaryType = nt:unstructured, component-optional = true}}],
values={jcr:primaryType=nt:unstructured, component-optional=true}]), Config(type=contentsync,
config=JcrPropertyMap
[node=Node[NodeDelegate{tree=/libs/cq/config/rewriter/default/transformer-contentsync: {
jcr:primaryType = nt:unstructured, component-optional = true}}],
values={jcr:primaryType=nt:unstructured, component-optional=true}]), Config(type=youtube-iframe,
config={}), serializer=Config(type=htmlwriter, config={}))
}
```

© 2015 Adobe Systems Incorporated. All Rights Reserved.

Final Thoughts

- Sling Rewriter is awesome
- It has very little to do with mod_rewrite
- Don’t use pipeline.mode=global
- Adjust includeTags if are you rewriting non-link HTML elements.
- Be careful when implementing characters().



Adobe