

Open APIs
for Open
Minds

The Art of Doing Nothing

Jason Fox
FIWARE Foundation



Passthrough Plug-In for DITA-OT

● Java 51.0%

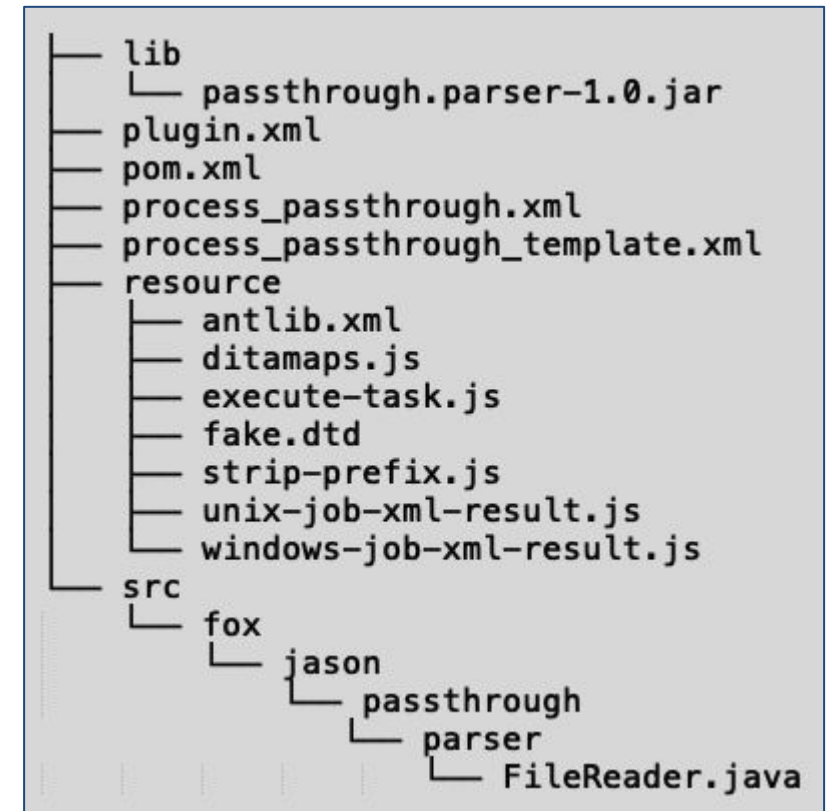
● JavaScript 49.0%

Does as little as possible

- Avoids preprocessing and adds hooks for other plug-ins

```
@Override
public void parse(InputSource input) throws IOException, SAXException {
    URL url = new URL(input.getSystemId());
    String name = url.getPath().replaceAll("_", " ");
    if(name.contains("/")) {
        name = name.substring(name.lastIndexOf("/") + 1, name.length());
        if(name.contains(".")) {
            name = name.substring(0, name.lastIndexOf("."));
        }
    }

    StringBuilder sb = new StringBuilder();
    sb.append("<topic id=\"" + name + "\" class=\"" + name + " domains=\"" + name + "\">");
    sb.append("<title class=\"" + name + "\">" + name + "</title>");
    sb.append("<body class=\"" + name + "\">");
    sb.append("</topic>");
    //Delegate to content handler.
    SAXResult result = new SAXResult(handler);
    try {
        TransformerFactory.newInstance().newTransformer().transform(new StreamSource(input), result);
    } catch (Exception e) {
        throw new IOException(e);
    }
}
```



- **GitHub:** fox.jason.passthrough

PanDoc Plug-In for DITA-OT

● Lua 92.7%

● JavaScript 3.7%

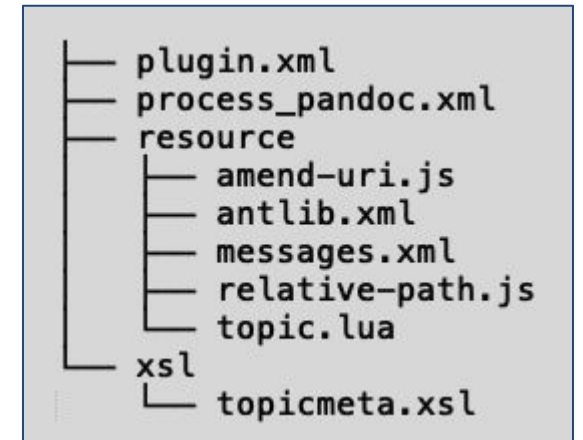
● XSLT 3.6%

An integration of the **PanDoc Haskell Library** for converting one markup format into another. Runs a **Lua template** which transforms over 30 different input formats to valid DITA

- **Extensions:**
 - passthrough.pre, passthrough.process
- Use **<exec>** to invoke PanDoc executable

```
<exec executable="pandoc" dir="${dita.temp.dir}"
  osfamily="unix" taskname="pandoc" errorproperty="pandoc.error"
  resultproperty="pandoc.result" >
  <arg line="&quot;@{src}&quot;; -t
  ${dita.plugin.fox.jason.passthrough.pandoc.dir}/resource/topic.lua -o
  &quot;@{dest}&quot;; -M title=&quot;@{title}&quot;; ${pandoc.metadata}"/>
</exec>
```

- **GitHub:** fox.jason.passthrough.pandoc



Swagger-to-DITA Plug-In for DITA-OT

● XSLT 66.3%

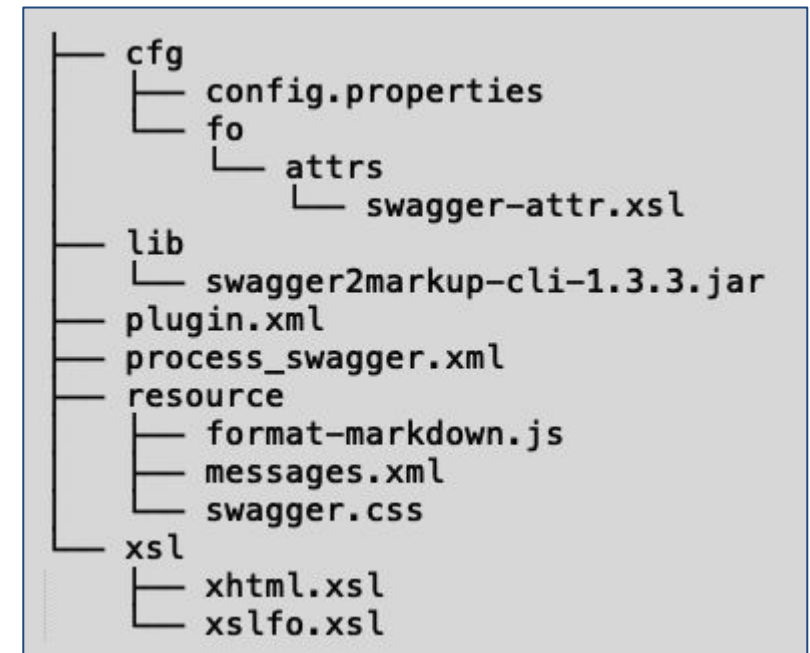
● JavaScript 20.2%

● CSS 13.5%

Creates valid DITA from a Swagger Specification

- **Extensions:**
 - passthrough.pre, passthrough.process
 - extend.css.process to inject CSS
- Invokes pre-existing Swagger-to-Markdown JAR

```
<java taskname="swagger-gen" outputproperty="swagger.output"
  resultproperty="swagger.result" errorproperty="swagger.error"
  jar="${dita.plugin.fox.jason.passthrough.swagger.dir}/lib/
  swagger2markup-cli-1.3.3.jar"
  fork="true" failonerror="false" maxmemory="128m">
  ... etc
</java>
<format-swagger-markdown file="${swagger.temp.file}.md"/>
<add-pandoc-file src="${swagger.temp.file}.md" dest="@{dest}"
  title="@{title}" metadata="@{metadata}"/>
```



- **GitHub:** [fox.jason.passthrough.swagger](https://github.com/fox-jason/passthrough.swagger)

Postman-Collection-to-DITA Plug-In for DITA-OT

● JavaScript 100.0%

Creates valid DITA from a Postman Collection JSON

- **Extensions:**
 - passthrough.pre
 - passthrough.process
- **JavaScript:** Generates the Markdown

```
<postman-to-markdown src="@{src}" dest="${postman.temp.file}"/>  
<add-pandoc-file src="${postman.temp.file}" dest="@{dest}"  
  title="@{title}" metadata="@{metadata}"/>
```

```
├─ plugin.xml  
├─ process_postman.xml  
└─ resource  
    └─ postman-to-markdown.js
```

- **GitHub:** fox.jason.passthrough.postman

Thank you!

<http://fiware.org>

Follow @FIWARE on Twitter

