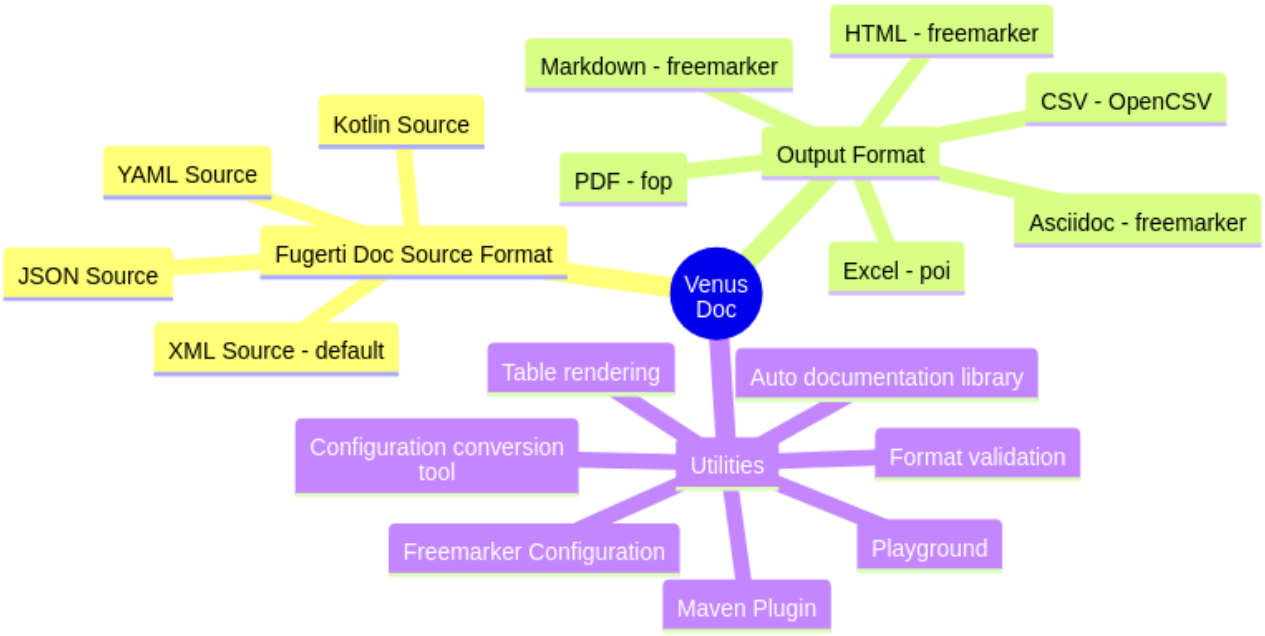
~~Fugerit DOC~~

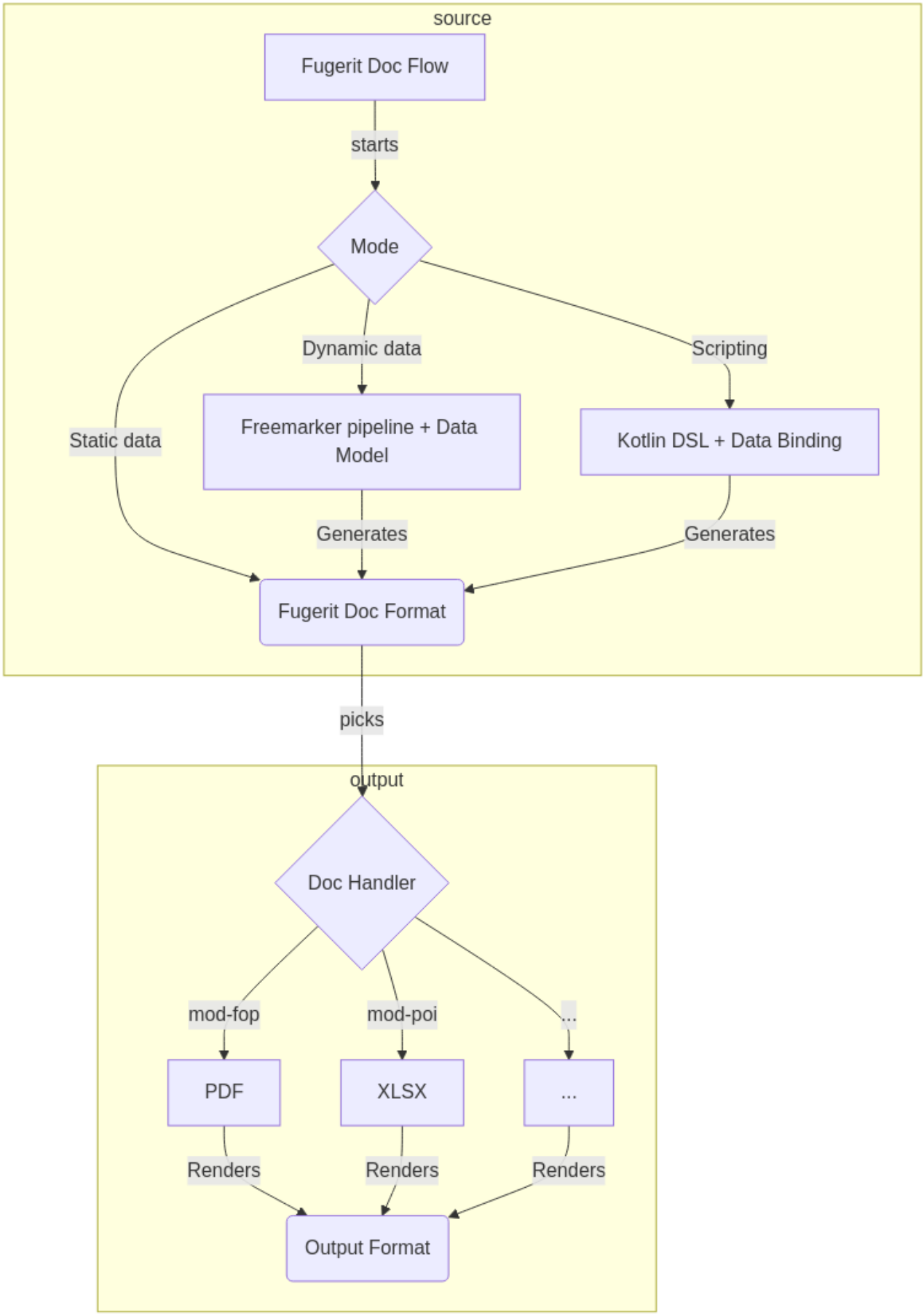
Venus Guide

Table of Contents

1. Introduction



Fugerit Venus Doc Mind Map (fj-doc)



Fugerit Venus Doc Typical Flow

Copyright

2. Quickstart

This section cover the creation of a new maven project, configured for Venus Doc.

2.1. New maven project

Just run the org.fugerit.java:fj-doc-maven-plugin:init plugin :

```
mvn org.fugerit.java:fj-doc-maven-plugin:init \
-DgroupId=org.example.doc \
-DartifactId=fugerit-demo \
-Dflavour=quarkus-3 \
-Dextensions=base,freemarker,mod-fop
```

And you will get a project folder named after the artifactId.
See the README.md in the project folder for further infos.

In case of 'quarkus-3' flavour, for instance, simply :

```
cd fugerit-demo
mvn quarkus:dev
```

Open the Open swagger UI to test available services.

See the Maven Plugin Goal init for further options.

2.2. Existing maven project

Just run the org.fugerit.java:fj-doc-maven-plugin:add plugin inside the folder containing your pom :

```
mvn org.fugerit.java:fj-doc-maven-plugin:add \
-Dextensions=base,freemarker,mod-fop
```

Dependencies, project configuration and an example main will be added to the project.

See the Maven Plugin Goal Add for further options.

3. Maven Plugin

Here is the Fugerit Doc Maven Plugin Documentation.

Currently, three goals are available :

goal	since	description
add	8.6.0	add Venus Doc Configuration to an existing project.
init	8.7.2	create a new project already configured (actually an extension to 'add' goal); a UI is available on the playground too.
verify	8.7.0	verify the templates in a FreeMarker configuration (folder), note: it can be used on any Apache FreeMarker configuration, not only Fugerit Venus Doc.

3.1. Goal 'add'

This goal add Fugerit Venus Doc configuration to an existing maven project.

Simply run the plugin in the folder containing your project POM :

```
mvn org.fugerit.java:fj-doc-maven-plugin:add \
-Dextensions=base,freemarker,mod-fop
```

3.1.1. Goal 'add' available parameters

parameter	required	default	description
version	true	latest stable	fj-doc version to add to the project (i.e. '8.7.5'), **advice**: **keep the default** unless a specific version is strictly needed.
extensions	true	base,freemarker	List of fj-doc core modules to add (*)
projectFolder	true	.	Maven project base folder
addDocFacade	true	true	If true, a stub doc configuration helper will be created

parameter	required	default	description
force	false	false	Will force project setup even if fj-doc already configured (warning: can overwrite configuration)
excludeXmlApis	false	false	It will exclude dependency xml-apis:xml-apis
addExclusions	false		Add comma separated exclusion, for instance : xml-apis:xml-apis,${groupId}:${artifiCatId}
addVerifyPlugin	true	true	If set to true, it will configure the 'verify' goal on the project
addJunit5	true	true	If set to true, it will add junit5 (test scope) and basic test
addLombok	true	true	If set to true, it will add lombok (provided scope) and slf4j-simple (test scope)
addDependencyOnTop	true	false	If set to true, added dependencies will be added before existing ones
freemarkerVersion	true	2.3.32	Freemarker compatibility version (max 2.3.33)

3.1.2. Available extensions.

short name	full name	type handler	description
base	fj-doc-base	md	library base, xml as format for document template

short name	full name	type handler	description
freemarker	fj-doc-freemarker	html	Template and configuration functionalities based on [Apache FreeMarker](https://freemarker.apache.org/)
mod-fop	fj-doc-mod-fop	fo, pdf	Type handler based on [Apache FOP](https://xmlgraphics.apache.org/fop/)
mod-poi	fj-doc-mod-poi	xls, xlsx	Type handler based on [Apache POI](https://poi.apache.org/)
mod-opencsv	fj-doc-mod-opencsv	opencsv	Type handler based on [OpenCSV](https://opencsv.sourceforge.net/)
mod-openpdf-ext	fj-doc-mod-openpdf-ext	pdf	Type handler based on [OpenPDF](https://github.com/LibrePDF/OpenPDF)
mod-openrtf-ext	fj-doc-mod-openrtf-ext	rtf	Type handler based on [OpenRTF](https://github.com/LibrePDF/OpenRTF)
base-json	fj-doc-base-json		add support to use json documents as format for document template
base-yaml	fj-doc-base-yaml		add support to use yaml documents as format for document template

3.2. Goal 'init'

Create a new project and add Venus Doc Configuration to it.

```
mvn org.fugerit.java:fj-doc-maven-plugin:init \
-DgroupId=org.example.doc \
-DartifactId=fugerit-demo \
```

```
-Dextensions=base,freemarker,mod-fop
```

Project folder will be `./${artifactId}/`.

3.2.1. Goal 'init' available parameters

parameter	required	default	description
groupId	true		new project group id
artifactId	true		new project artifact id
projectVersion	true	1.0.0-SNAPSHOT	new project version
javaRelease	true	21	java release version
flavour	true	vanilla	the flavour for the new project (see below for options)
flavourVersion	false	see below	override default framework version if supported (recommended : leave default or blank)

it is possible to set any property from 'add' goal, except 'projectFolder' which is set to `./${artifactId}`.

3.3. Goal 'verify'

verify the templates in a FreeMarker configuration (folder), note: it can be used on any Apache FreeMarker configuration, not only Fugerit Venus Doc.

3.3.1. Verify at command line

```
mvn org.fugerit.java:fj-doc-maven-plugin:verify -DtemplateBasePath
=./src/test/resources/fj_doc_test/template-fail
```

3.3.2. Verify at maven build time

```
  <plugin>
    <groupId>org.fugerit.java</groupId>
    <artifactId>fj-doc-maven-plugin</artifactId>
    <version>${fj-doc-version}</version>
    <executions>
      <execution>
```

```
        <id>freemarker-verify</id>
        <phase>compile</phase>
        <goals>
          <goal>verify</goal>
        </goals>
      </execution>
    </executions>
    <configuration>
      <!-- Where the FreeMarker templates are located -->
      <templateBasePath>${project.basedir}/src/main/resources/fugerit-
blank/template</templateBasePath>
      <!-- WARNING: if set to 'true', build will fail when at least one syntax error
is found -->
      <failOnErrors>true</failOnErrors>
      <!-- If 'true' a report will be generated (when 'true', param
reportOutputFolder is required) -->
      <generateReport>true</generateReport>
      <!-- Template syntax verify report output folder -->
      <reportOutputFolder>${project.build.directory}/freemarker-syntax-verify-
report</reportOutputFolder>
    </configuration>
  </plugin>
```

3.3.3. Goal 'verify' available parameters

parameter	required	default	description
templateBasePath	true		Path to base folder containing FreeMarker templates
freemarkerVersion	false	latest stable	FreeMarker configuration (i.e. 2.3.33)
templateFilePattern	false		Filter on templates to be checked, regex on filename, i.e. ".{0,}[.]ftl"
failOnErrors	true	true	If set to true the build will fail when template syntax errors will be found, otherwise errors will be only logged

parameter	required	default	description
generateReport	false	false	If set to true a report will be generated (and property 'reportOutputFolder' will be olso required).
reportOutputFolder	false		Output folder for the generated report.
reportOutputFormat	false	'html'	Output format for the generated report, supported : html (default), pdf, csv, xlsx, md.

4. The Doc Source Format

The Doc Source Format is the basic of Fugerit Venus Doc library.

4.1. XML Source Format

When writing a sample Venus Document, it is possible to draw from some online resources :

1. Venus DOC XML Schema Definition - Current version of the Venus DOC XSD, the main source for writing valid Venus document meta model.
2. Venus DOC XML Reference - the informations contained in the previous XSD in HTML format for convenience.
3. Venus DOC meta informations reference - documentations for existing 'info' properties for 'metadata' section.
4. Online Playground - To test how the XML elements are rendered to documents by DocHandlers.

```
<?xml version="1.0" encoding="utf-8"?>
<doc
    xmlns="http://javacoredoc.fugerit.org"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://javacoredoc.fugerit.org
https://www.fugerit.org/data/java/doc/xsd/doc-2-1.xsd" >

  <!--
    This is a Venus Fugerit Doc (https://github.com/fugerit-org/fj-doc) XML Source
Document.
    For documentation on how to write a valid Venus Doc XML Meta Model refer to :
    https://venusdocs.fugerit.org/guide/#doc-format-entry-point
  -->

  <metadata>
    <!-- Margin for document : left;right;top;bottom -->
    <info name="margins">10;10;10;30</info>
    <!-- documenta meta information -->
    <info name="doc-title">Hello World</info>
    <info name="doc-author">fugerit79</info>
    <info name="doc-language">en</info>
  </metadata>
  <body>
    <para>Hello World!</para>
  </body>

</doc>
```

4.2. JSON and YAML Source Format

It is possible to use JSON or YAML as source. It is just needed to follow some conversion rules.

The conversion rules from xml to json/yaml are :

1. xml meta information are translated as top level property (xmlns etc.)
2. Every JSON/YAML object contains the information of a XML tag
3. "_t" special property contains the tag name (for example "_t" : "metadata")
4. "_v" special property contains the text content of an element
5. "_e" special property contains a list of child elements
6. any other xml attribute is mapped as a JSON/YAML property (for example "name" : "margins")

As it is possible to directly convert JSON and YAML, rules for YAML are the same as for JSON format.

All XML comments are ignored

For XML/JSON/YAML conversion a Online Playground is available.

Here is a simple JSON source document :

```
{
    "xmlns" : "http://javacoredoc.fugerit.org",
    "xsi:schemaLocation" : "http://javacoredoc.fugerit.org
https://www.fugerit.org/data/java/doc/xsd/doc-2-0.xsd",
    "xmlns:xsi" : "http://www.w3.org/2001/XMLSchema-instance",
    "_t" : "doc",
    "_e" : [ {
        "_t" : "metadata",
        "_e" : [ {
            "name" : "margins",
            "_t" : "info",
            "_v" : "10;10;10;30"
        }, {
            "name" : "doc-title",
            "_t" : "info",
            "_v" : "Hello World"
        }, {
            "name" : "doc-author",
            "_t" : "info",
            "_v" : "fugerit79"
        }, {
            "name" : "doc-language",
            "_t" : "info",
```

```
            "_v" : "en"
        } ]
    }, {
        "_t" : "body",
        "_e" : [ {
            "_t" : "para",
            "_v" : "Hello World!"
        } ]
    } ]
}
```

And the equivalent YAML :

```
---
xmlns: "http://javacoredoc.fugerit.org"
xsi:schemaLocation: "http://javacoredoc.fugerit.org
https://www.fugerit.org/data/java/doc/xsd/doc-2-1.xsd"
xmlns:xsi: "http://www.w3.org/2001/XMLSchema-instance"
_t: "doc"
_e:
- _t: "metadata"
  _e:
  - name: "margins"
    _t: "info"
    _v: "10;10;10;30"
  - name: "doc-title"
    _t: "info"
    _v: "Basic example"
  - name: "doc-subject"
    _t: "info"
    _v: "fj doc venus sample source xml"
  - name: "doc-author"
    _t: "info"
    _v: "fugerit79"
  - name: "doc-language"
    _t: "info"
    _v: "en"
  - name: "default-font-name"
    _t: "info"
    _v: "TitilliumWeb"
  - name: "excel-table-id"
    _t: "info"
    _v: "data-table=print"
  - name: "csv-table-id"
    _t: "info"
    _v: "data-table"
  - _t: "footer-ext"
    _e:
```

```
    - align: "right"
      _t: "para"
      _v: "${currentPage} / ${pageCount}"
- _t: "body"
  _e:
  - _t: "para"
    _v: "My sample title"
  - padding: "2"
    columns: "3"
    width: "100"
    id: "data-table"
    colwidths: "30;30;40"
    _t: "table"
    _e:
    - _t: "row"
      _e:
      - border-color: "#000000"
        border-width: "1"
        align: "center"
        _t: "cell"
        _e:
        - style: "bold"
          _t: "para"
          _v: "Name"
      - align: "center"
        _t: "cell"
        _e:
        - style: "bold"
          _t: "para"
          _v: "Surname"
      - align: "center"
        _t: "cell"
        _e:
        - style: "bold"
          _t: "para"
          _v: "Title"
    - _t: "row"
      _e:
      - _t: "cell"
        _e:
        - _t: "para"
          _v: "Luthien"
      - _t: "cell"
        _e:
        - _t: "para"
          _v: "Tinuviel"
      - _t: "cell"
        _e:
        - _t: "para"
```

```
          _v: "Queen"
    - _t: "row"
      _e:
      - _t: "cell"
        _e:
        - _t: "para"
          _v: "Thorin"
      - _t: "cell"
        _e:
        - _t: "para"
          _v: "Oakshield"
      - _t: "cell"
        _e:
        - _t: "para"
          _v: "King"
```

4.3. XML Source Format

When writing a sample Venus Document, it is possible to draw from some online resources :

1. Venus DOC XML Schema Definition - Current version of the Venus DOC XSD, the main source for writing valid Venus document meta model.
2. Venus DOC XML Reference - the informations contained in the previous XSD in HTML format for convenience.
3. Venus DOC meta informations reference - documentations for existing 'info' properties for 'metadata' section.
4. Online Playground - To test how the XML elements are rendered to documents by DocHandlers.

Supported is provided by the base module dependency :

```
<dependency>
    <groupId>org.fugerit.java</groupId>
    <artifactId>fj-doc-base</artifactId>
    <version>${fj-doc-version}</version>
</dependency>
```

Here is an example xml :

```
<?xml version="1.0" encoding="utf-8"?>
<doc
    xmlns="http://javacoredoc.fugerit.org"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://javacoredoc.fugerit.org
https://www.fugerit.org/data/java/doc/xsd/doc-2-1.xsd" >
```

```
  <!--
    This is a Venus Fugerit Doc (https://github.com/fugerit-org/fj-doc) XML Source
Document.
    For documentation on how to write a valid Venus Doc XML Meta Model refer to :
    https://venusdocs.fugerit.org/guide/#doc-format-entry-point
  -->

  <metadata>
    <!-- Margin for document : left;right;top;bottom -->
    <info name="margins">10;10;10;30</info>
    <!-- documenta meta information -->
    <info name="doc-title">Hello World</info>
    <info name="doc-author">fugerit79</info>
    <info name="doc-language">en</info>
  </metadata>
  <body>
    <para>Hello World!</para>
  </body>

</doc>
```

4.4. JSON and YAML Source Format

It is possible to use JSON or YAML as source. It is just needed to follow some conversion rules.

The conversion rules from xml to json/yaml are :

1. xml meta information are translated as top level property (xmlns etc.)
2. Every JSON/YAML object contains the information of a XML tag
3. "_t" special property contains the tag name (for example "_t" : "metadata")
4. "_v" special property contains the text content of an element
5. "_e" special property contains a list of child elements
6. any other xml attribute is mapped as a JSON/YAML property (for example "name" : "margins")

As it is possible to directly convert JSON and YAML, rules for YAML are the same as for JSON format.

All XML comments are ignored

For XML/JSON/YAML conversion a Online Playground is available.

Support for *JSON* format needs the following dependency :

```
<dependency>
    <groupId>org.fugerit.java</groupId>
    <artifactId>fj-doc-base-json</artifactId>
    <version>${fj-doc-version}</version>
</dependency>
```

Here is a simple JSON source document :

```
{
    "xmlns" : "http://javacoredoc.fugerit.org",
    "xsi:schemaLocation" : "http://javacoredoc.fugerit.org
https://www.fugerit.org/data/java/doc/xsd/doc-2-0.xsd",
    "xmlns:xsi" : "http://www.w3.org/2001/XMLSchema-instance",
    "_t" : "doc",
    "_e" : [ {
        "_t" : "metadata",
        "_e" : [ {
            "name" : "margins",
            "_t" : "info",
            "_v" : "10;10;10;30"
        }, {
            "name" : "doc-title",
            "_t" : "info",
            "_v" : "Hello World"
        }, {
            "name" : "doc-author",
            "_t" : "info",
            "_v" : "fugerit79"
        }, {
            "name" : "doc-language",
            "_t" : "info",
            "_v" : "en"
        } ]
    }, {
        "_t" : "body",
        "_e" : [ {
            "_t" : "para",
            "_v" : "Hello World!"
        } ]
    } ]
}
```

Support for *YAML* format needs the following dependency :

```
<dependency>
    <groupId>org.fugerit.java</groupId>
```

```
    <artifactId>fj-doc-base-yaml</artifactId>
    <version>${fj-doc-version}</version>
</dependency>
```

Here is the equivalent YAML :

```
---
xmlns: "http://javacoredoc.fugerit.org"
xsi:schemaLocation: "http://javacoredoc.fugerit.org
https://www.fugerit.org/data/java/doc/xsd/doc-2-1.xsd"
xmlns:xsi: "http://www.w3.org/2001/XMLSchema-instance"
_t: "doc"
_e:
- _t: "metadata"
  _e:
  - name: "margins"
    _t: "info"
    _v: "10;10;10;30"
  - name: "doc-title"
    _t: "info"
    _v: "Basic example"
  - name: "doc-subject"
    _t: "info"
    _v: "fj doc venus sample source xml"
  - name: "doc-author"
    _t: "info"
    _v: "fugerit79"
  - name: "doc-language"
    _t: "info"
    _v: "en"
  - name: "default-font-name"
    _t: "info"
    _v: "TitilliumWeb"
  - name: "excel-table-id"
    _t: "info"
    _v: "data-table=print"
  - name: "csv-table-id"
    _t: "info"
    _v: "data-table"
  - _t: "footer-ext"
    _e:
    - align: "right"
      _t: "para"
      _v: "${currentPage} / ${pageCount}"
- _t: "body"
  _e:
  - _t: "para"
    _v: "My sample title"
```

```
  - padding: "2"
    columns: "3"
    width: "100"
    id: "data-table"
    colwidths: "30;30;40"
    _t: "table"
    _e:
    - _t: "row"
      _e:
      - border-color: "#000000"
        border-width: "1"
        align: "center"
        _t: "cell"
        _e:
        - style: "bold"
          _t: "para"
          _v: "Name"
      - align: "center"
        _t: "cell"
        _e:
        - style: "bold"
          _t: "para"
          _v: "Surname"
      - align: "center"
        _t: "cell"
        _e:
        - style: "bold"
          _t: "para"
          _v: "Title"
    - _t: "row"
      _e:
      - _t: "cell"
        _e:
        - _t: "para"
          _v: "Luthien"
      - _t: "cell"
        _e:
        - _t: "para"
          _v: "Tinuviel"
      - _t: "cell"
        _e:
        - _t: "para"
          _v: "Queen"
    - _t: "row"
      _e:
      - _t: "cell"
        _e:
        - _t: "para"
          _v: "Thorin"
```

```
      - _t: "cell"
        _e:
        - _t: "para"
          _v: "Oakshield"
      - _t: "cell"
        _e:
        - _t: "para"
          _v: "King"
```

4.5. Kotlin Source Format (Experimental)

Supported is provided by the base module dependency.

```
<dependency>
    <groupId>org.fugerit.java</groupId>
    <artifactId>fj-doc-ext-kotlin</artifactId>
    <version>${fj-doc-ext-kotlin-version}</version>
</dependency>
```

At the moment kotlin (KTS) support is still experimental. The Kotlin extension is maintained on a separate repository.

Kotlin source is based on a custom DSL (Domain Specific Language).
The Fugerit Doc Kotlin DSL is basically a mapping of the xsd.

Here is a sample kotlin source document :

```
import org.fugerit.java.doc.base.kotlin.dsl.dslDoc

dslDoc {
    val docTitle = attStr( data, "docTitle" )
    meta {
        info( ( docTitle ) ).name( "doc-title" )
        info( ( "10;10;10;30" ) ).name( "margins" )
        info( ( "fj doc venus sample source Kotlin Template - kts" ) ).name( "doc-
subject" )
        info( ( "fugerit79" ) ).name( "dock-author" )
        info( ( "en" ) ).name( "doc-language" )
        info( ( "TitilliumWeb" ) ).name( "default-font-name" )
        info( ( "data-table=print" ) ).name( "excel-table-id" )
        info( ( "data-table" ) ).name( "csv-table-id" )
        footerExt {
            para( '$'+"{currentPage} / "+'$'+"{pageCount}" ).align( "right" )
        }
    }
    body {
```

```
        h( docTitle ).headLevel( 1 )
        table {
            row {
                cell { para( "Name" ) }.align( "center" )
                cell { para( "Surname" ) }.align( "center" )
                cell { para( "Title" ) }.align( "center" )
            }.header( true )
            attListMap( data, "listPeople" ).forEach( { e -> row {
                cell { para( attStr( e, "name" ) ) }
                cell { para( attStr( e, "surname" ) ) }
                cell { para( attStr( e, "title" ) ) }
            } } )
        }.columns( 3 ).colwidths( "30;30;40" ).width( 100 ).id( "data-table"
).padding( 2 )
    }
}
```

4.6. Doc Format Info Element

The *info* element is available to setup some specific properties of the document.

Here is a list of possible info element.

For instance in this document we set following *info* elements :

- margin
- doc-title
- doc-author
- doc-language

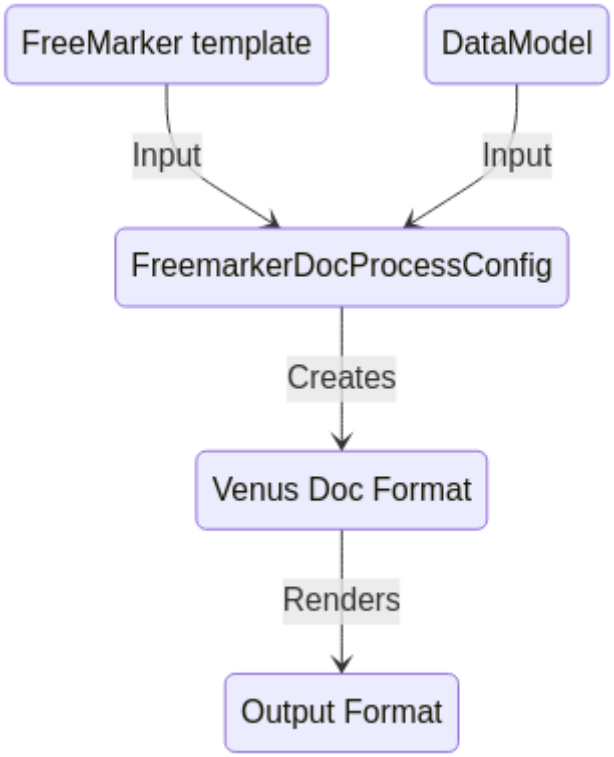
Some DocHandler will ignore some info elements.

```
<?xml version="1.0" encoding="utf-8"?>
<doc
    xmlns="http://javacoredoc.fugerit.org"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://javacoredoc.fugerit.org
https://www.fugerit.org/data/java/doc/xsd/doc-2-1.xsd" >

  <metadata>
    <!-- Margin for document : left;right;top;bottom -->
    <info name="margins">10;10;10;30</info>
    <!-- documenta meta information -->
    <info name="doc-title">Hello World</info>
    <info name="doc-author">fugerit79</info>
    <info name="doc-language">en</info>
```

```
  </metadata>
  <body>
    <para>Hello World!</para>
  </body>

</doc>
```

5. Dynamic Data : Freemarker

In a typical workflow, freemarker will be used to generate the Venus Doc Format.



Fugerit Venus Doc - Dynamic Document Generation

This is a simple tutorial for Fugerit Venus Doc full usage.

This tutorial is based on a quarkus project available at :
tutorial repository git.

The project has been created with the command :

```
mvn org.fugerit.java:fj-doc-maven-plugin:init \
-DgroupId=org.fugerit.java-tutorial \
-DartifactId=fj-doc-quarkus-tutorial \
-Dflavour=quarkus-3 \
-Dextensions=base,freemarker,mod-fop,mod-poi
```

5.1. FreemarkerDocProcessConfig configuration

The main configuration file is fm-doc-process-config.xml.

Here you can find :

- freemarker doc process config xsd

- freemarker doc process config reference

5.1.1. Root element freemarker-doc-process-config

This is the root configuration element :

```
<freemarker-doc-process-config
    xmlns="https://freemarkerdocprocess.fugerit.org"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="https://freemarkerdocprocess.fugerit.org
https://www.fugerit.org/data/java/doc/xsd/freemarker-doc-process-1-0.xsd"
    validating="true"
    failOnValidate="true"
    cleanSource="true">
```

Aside from namespace declaration, it is possible to set some general attribute for the configuration :

name	type	default	description
validating	*boolean*	*false*	If set to 'true' the FreemarkerDocProcessConfig will try to validate the source. (since 8.9.1), NOTE: if active, source reader will be buffered, potentially resulting in higher memory usage. See also FAQ.
failOnValidate	*boolean*	*false*	If set to 'true' the FreemarkerDocProcessConfig will fail in case of validation errors, if 'false' will just print the result as warning. (since 8.9.1), NOTE: 'validating' is set to true, this attribute is ignored. See also FAQ
cleanSource	*boolean*	*false*	If set to 'true' the FreemarkerDocProcessConfig will try to clean the source (i.e. DocXmlUtils.cleanXml()). (since 8.9.1), NOTE: if active, source reader will be buffered, potentially resulting in higher memory usage. See also FAQ

5.1.2. Element docHandlerConfig

The main elements are the <**docHandlerConfig**/> containing the available output format doc handlers.

```
    <docHandlerConfig registerById="true">
        <!-- Type handler for markdown format -->
        <docHandler id="md-ext" info="md"
type="org.fugerit.java.doc.base.typehandler.markdown.SimpleMarkdownExtTypeHandler"
/>
        <!-- Type henalder for xml format, generates the source xml:doc -->
        <docHandler id="xml-doc" info="xml"
```

```
type="org.fugerit.java.doc.base.config.DocTypeHandlerXMLUTF8" />
        <!-- Type handler for html using freemarker -->
        <docHandler id="html-fm" info="html"
type="org.fugerit.java.doc.freemarker.html.FreeMarkerHtmlTypeHandlerEscapeUTF8" />
        <!-- Type handler for html using freemarker (fragment version, only
generates body content no html or head part -->
        <docHandler id="html-fragment-fm" info="fhtml"
type="org.fugerit.java.doc.freemarker.html.FreeMarkerHtmlFragmentTypeHandlerEscapeUT
F8" />
        <!-- Type handler generating xls:fo style sheet -->
        <docHandler id="fo-fop" info="fo"
type="org.fugerit.java.doc.mod.fop.FreeMarkerFopTypeHandlerUTF8" />
        <!-- Type handler generating pdf -->
        <docHandler id="pdf-fop" info="pdf"
type="org.fugerit.java.doc.mod.fop.PdfFopTypeHandler">
            <docHandlerCustomConfig charset="UTF-8" fop-config-mode="classloader"
fop-config-classloader-path="fj-doc-quarkus-tutorial/fop-config.xml" fop-suppress-
events="1"/>
        </docHandler>
        <!-- Type handler generating xls -->
        <docHandler id="xls-poi" info="xls"
type="org.fugerit.java.doc.mod.poi.XlsPoiTypeHandler" />
        <!-- Type handler generating xlsx -->
        <docHandler id="xlsx-poi" info="xlsx"
type="org.fugerit.java.doc.mod.poi.XlsxPoiTypeHandler" />
    </docHandlerConfig>
```

5.1.3. Element docChain

And the <**docChain**/> elements containing the configuration for the freemarker template and data model.

Usually you will need at least one configuration step :

```
    <docChain id="shared">
        <chainStep stepType="config">
            <config
                    id="fj_doc_config_fm_fjdocquarkustutorial"
                    class="org.fugerit.java.tutorial.fjdocquarkustutorial.DocHelper"
                    exception-handler="RETHROW_HANDLER"
                    fallback-on-null-loop-variable="false"
                    log-exception="false"
                    mode="class"
                    path="/fj-doc-quarkus-tutorial/template/"
                    version="2.3.32"
                    wrap-unchecked-exceptions="true"
                    load-bundled-functions="true"
            />
```

```
        </chainStep>
    </docChain>
```

And one or more document process step :

```
    <!-- example document chain -->
    <docChain id="document" parent="shared">
        <chainStep stepType="complex" map-atts="listPeople" template-
path="${chainId}.ftl"/>
    </docChain>
```

5.1.4. Build in step types

name	type	description
config	org.fugerit.java.doc.freemarker.config.FreeMarkerConfigStep	responsable for freemarker and venus configuration
function	org.fugerit.java.doc.freemarker.config.FreeMarkerFunctionStep	add freemarker functions to the data model
complex	org.fugerit.java.doc.freemarker.config.FreeMarkerComplexProcessStep	it apply the freemarker template and render the output
map	org.fugerit.java.doc.freemarker.config.FreeMarkerMapStep	Venus data model to Freemarker data model mapping

Additional step can be added setting the fully qualified class name in the type attribute.

5.2. Venus Freemarker Chain

With a venus freemarker chain, we use an Apache FreeMarker template (ftl) :

```
<?xml version="1.0" encoding="utf-8"?>
<doc
xmlns="http://javacoredoc.fugerit.org"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://javacoredoc.fugerit.org
https://www.fugerit.org/data/java/doc/xsd/doc-2-1.xsd" >

    <#--
        This is a Venus Fugerit Doc (https://github.com/fugerit-org/fj-doc)
```

```
FreeMarker Template XML (ftl[x]).
        For consideration of Venus Fugerit Doc and Apache FreeMarker integration see
:
        https://venusguides.fugerit.org/src/docs/common/doc_format_freemarker.html
        The result will be a :
    -->
    <!--
        This is a Venus Fugerit Doc (https://github.com/fugerit-org/fj-doc) XML
Source Document.
        For documentation on how to write a valid Venus Doc XML Meta Model refer to
:
        https://venusguides.fugerit.org/src/docs/common/doc_format_summary.html
    -->

    <#assign defaultTitle="My sample title">

    <metadata>
        <!-- Margin for document : left;right;top;bottom -->
        <info name="margins">10;10;10;30</info>
        <!-- documenta meta information -->
        <info name="doc-title">${docTitle!defaultTitle}</info>
        <info name="doc-subject">fj doc venus sample source FreeMarker Template XML
- ftlx</info>
        <info name="doc-author">fugerit79</info>
        <info name="doc-language">en</info>
        <!-- property specific for xls/xlsx -->
        <info name="excel-table-id">data-table=print</info>
        <!-- property specific for csv -->
        <info name="csv-table-id">data-table</info>
        <footer-ext>
            <para align="right">${r"${currentPage}"} / ${r"${pageCount}"}</para>
        </footer-ext>
    </metadata>
    <body>
    <para>${docTitle!defaultTitle}</para>
    <table columns="3" colwidths="30;30;40"  width="100" id="data-table"
padding="2">
        <row header="true">
            <cell align="center"><para>Name</para></cell>
            <cell align="center"><para>Surname</para></cell>
            <cell align="center"><para>Title</para></cell>
        </row>
        <#if listPeople??>
            <#list listPeople as current>
                <row>
                    <cell><para>${current.name}</para></cell>
                    <cell><para>${current.surname}</para></cell>
                    <cell><para>${current.title}</para></cell>
                </row>
```

```
            </#list>
        </#if>
    </table>
    </body>

</doc>
```

Adding to it the data model, and applying the template :

```
DocHelper docHelper = new DocHelper();
// create custom data for the fremarker template 'document.ftl'
List<People> listPeople = Arrays.asList(new People("Luthien", "Tinuviel", "Queen"),
new People("Thorin", "Oakshield", "King"));
// output generation
docHelper.getDocProcessConfig().fullProcess("document", DocProcessContext.
newContext("listPeople", listPeople), handlerId, baos);
```

The Venus Freemarker Template and the data model will produce a Venus Doc Format document source :

```
<?xml version="1.0" encoding="utf-8"?>
<doc
xmlns="http://javacoredoc.fugerit.org"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://javacoredoc.fugerit.org
https://www.fugerit.org/data/java/doc/xsd/doc-2-1.xsd" >

        <!--
        This is a Venus Fugerit Doc (https://github.com/fugerit-org/fj-doc) XML
Source Document.
        For documentation on how to write a valid Venus Doc XML Meta Model refer to
:
        https://venusguides.fugerit.org/src/docs/common/doc_format_summary.html
    -->


    <metadata>
        <!-- Margin for document : left;right;top;bottom -->
        <info name="margins">10;10;10;30</info>
        <!-- documenta meta information -->
        <info name="doc-title">My sample title</info>
        <info name="doc-subject">fj doc venus sample source FreeMarker Template XML
- ftlx</info>
        <info name="doc-author">fugerit79</info>
        <info name="doc-language">en</info>
        <!-- property specific for xls/xlsx -->
        <info name="excel-table-id">data-table=print</info>
```

```
        <!-- property specific for csv -->
        <info name="csv-table-id">data-table</info>
        <footer-ext>
            <para align="right">${currentPage} / ${pageCount}</para>
        </footer-ext>
    </metadata>
    <body>
    <para>My sample title</para>
    <table columns="3" colwidths="30;30;40"  width="100" id="data-table" padding=
"2">
        <row header="true">
            <cell align="center"><para>Name</para></cell>
            <cell align="center"><para>Surname</para></cell>
            <cell align="center"><para>Title</para></cell>
        </row>
                <row>
                    <cell><para>Luthien</para></cell>
                    <cell><para>Tinuviel</para></cell>
                    <cell><para>Queen</para></cell>
                </row>
                <row>
                    <cell><para>Thorin</para></cell>
                    <cell><para>Oakshield</para></cell>
                    <cell><para>King</para></cell>
                </row>
    </table>
    </body>

</doc>
```

And pass it to the chosen DocHandler in order to have the desired output format, for instance the markdown version :

```
My sample title

| Name | Surname | Title  |
|---------------|---------------|---------------|
| Luthien | Tinuviel | Queen  |
| Thorin | Oakshield | King  |
```

5.3. Venus Freemarker Usage

Adding everything together, we need a facade to read the configuration and process the Freemarker template chains :

```
package org.fugerit.java.tutorial.fjdocquarkustutorial;
```

```
import org.fugerit.java.doc.freemarker.process.FreemarkerDocProcessConfig;
import org.fugerit.java.doc.freemarker.process.FreemarkerDocProcessConfigFacade;

/**
* DocHelper, version : auto generated on 2024-09-25 23:53:33.687
*/
public class DocHelper {

     private FreemarkerDocProcessConfig docProcessConfig =
FreemarkerDocProcessConfigFacade.loadConfigSafe( "cl://fj-doc-quarkus-tutorial/fm-
doc-process-config.xml" );

     public FreemarkerDocProcessConfig getDocProcessConfig() { return this
.docProcessConfig; }

}
```

And for instance a rest serive to expose the api :

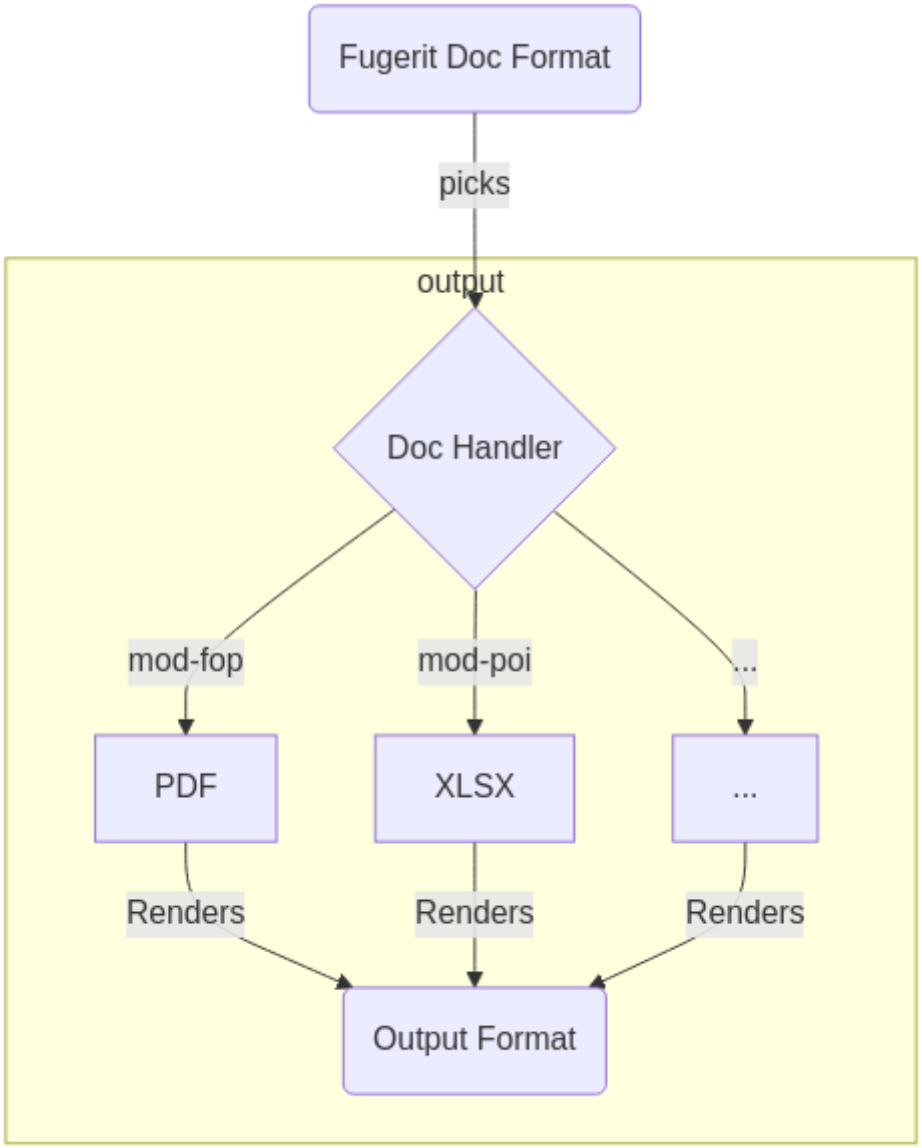
```
package org.fugerit.java.tutorial.fjdocquarkustutorial;

import jakarta.ws.rs.GET;
import jakarta.ws.rs.Path;
import jakarta.ws.rs.Produces;
import jakarta.ws.rs.WebApplicationException;
import jakarta.ws.rs.core.MediaType;
import lombok.extern.slf4j.Slf4j;
import org.fugerit.java.doc.base.config.DocConfig;
import org.fugerit.java.doc.base.process.DocProcessContext;

import java.io.ByteArrayOutputStream;
import java.util.Arrays;
import java.util.List;

import org.eclipse.microprofile.openapi.annotations.Operation;
import org.eclipse.microprofile.openapi.annotations.responses.APIResponse;
import org.eclipse.microprofile.openapi.annotations.tags.Tag;
import org.eclipse.microprofile.openapi.annotations.tags.Tags;

@Slf4j
@Path("/doc")
public class DocResource {

    byte[] processDocument(String handlerId) {
        try (ByteArrayOutputStream baos = new ByteArrayOutputStream()) {
            // creates the doc helper
            DocHelper docHelper = new DocHelper();
```

```
            // create custom data for the fremarker template 'document.ftl'
            List<People> listPeople = Arrays.asList(new People("Luthien",
"Tinuviel", "Queen"), new People("Thorin", "Oakshield", "King"));
            // output generation
            docHelper.getDocProcessConfig().fullProcess("document",
DocProcessContext.newContext("listPeople", listPeople), handlerId, baos);
            // return the output
            return baos.toByteArray();
        } catch (Exception e) {
            String message = String.format("Error processing %s, error:%s",
handlerId, e);
            log.error(message, e);
            throw new WebApplicationException(message, e);
        }
    }

    @APIResponse(responseCode = "200", description = "The Markdown document content"
)
    @APIResponse(responseCode = "500", description = "In case of an unexpected
error" )
    @Tags( { @Tag( name = "document" ), @Tag( name = "markdown" ) } )
    @Operation( operationId = "MarkdownExample", summary = "Example Markdown
generation",
        description =  "Generates an example Markdown document using Fugerit Venus
Doc handler" )
    @GET
    @Produces("text/markdown")
    @Path("/example.md")
    public byte[] markdownExample() {
        return processDocument(DocConfig.TYPE_MD);
    }

    @APIResponse(responseCode = "200", description = "The HTML document content" )
    @APIResponse(responseCode = "500", description = "In case of an unexpected
error" )
    @Tags( { @Tag( name = "document" ), @Tag( name = "html" ) } )
    @Operation( operationId = "HTMLExample", summary = "Example HTML generation",
        description =  "Generates an example HTML document using Fugerit Venus Doc
handler" )
    @GET
    @Produces("text/html")
    @Path("/example.html")
    public byte[] htmlExample() {
        return processDocument(DocConfig.TYPE_HTML);
    }

    @APIResponse(responseCode = "200", description = "The PDF document content" )
    @APIResponse(responseCode = "500", description = "In case of an unexpected
error" )
```

```
    @Tags( { @Tag( name = "document" ), @Tag( name = "pdf" ) } )
    @Operation( operationId = "PDFExample", summary = "Example PDF generation",
        description =  "Generates an example PDF document using Fugerit Venus Doc
handler" )
    @GET
    @Produces("application/pdf")
    @Path("/example.pdf")
    public byte[] pdfExample() {
        return processDocument(DocConfig.TYPE_PDF);
    }

    @APIResponse(responseCode = "200", description = "The Excel document content" )
    @APIResponse(responseCode = "500", description = "In case of an unexpected
error" )
    @Tags( { @Tag( name = "document" ), @Tag( name = "excel" ) } )
    @Operation( operationId = "ExcelExample", summary = "Example Excel generation",
        description =  "Generates an example Excel document using Fugerit Venus Doc
handler" )
    @GET
    @Produces("application/vnd.openxmlformats-officedocument.spreadsheetml.sheet")
    @Path("/example.xlsx")
    public byte[] excelExample() {
        return processDocument(DocConfig.TYPE_XLSX);
    }

}
```

6. Doc Handlers

Doc Handler are modules responsible for rendering the output format.



Fugerit Venus Doc - Doc Handlers

6.1. DocHandler Module Index

doc-handler	module	type	description
org.fugerit.java.doc.freemarker.html.FreeMarkerHtmlTypeHandlerEscapeUTF8	fj-doc-freemarker	HTML	Renders HTML documents using Apache FreeMarker template engine.

doc-handler	module	type	description
org.fugerit.java.doc.freemarker.html.FreeMarkerHtmlFragmentTypeHandlerEscapeUTF8	fj-doc-freemarker	HTML Fragment	Same as above, but an HTML fragment is just the body of the HTML document.
org.fugerit.java.doc.mod.fop.PdfFopTypeHandler	fj-doc-mod-fop	PDF (PDF/A, PDF/UA)	A PDF doc handler based on Apache FOP Project. It offers options for PDF/A and PDF/UA formats.
org.fugerit.java.doc.mod.fop.FreeMarkerFopTypeHandlerUTF8	fj-doc-mod-fop	fo	A XLS-FO doc handler based on Apache FOP Project.
org.fugerit.java.doc.mod.poi.XlsxPoiTypeHandler	fj-doc-mod-poi	XLSX	Generates an XLSX document using Apache POI Project.
org.fugerit.java.doc.mod.poi.XlsPoiTypeHandler	fj-doc-mod-poi	XLS	Generates an XLS document using Apache POI Project.
org.fugerit.java.doc.mod.opencsv.OpenCSVTypeHandler	fj-doc-mod-opencsv	CSV	Generates a CSV document using OpenCSV.
org.fugerit.java.doc.mod.openpdf.ext.PdfTypeHandler	fj-doc-mod-openpdf-ext	PDF	Generates a PDF document using OpenPDF.
org.fugerit.java.doc.mod.openpdf.ext.HtmlTypeHandler	fj-doc-mod-openpdf-ext	HTML	Generates a HTML document using OpenPDF.
org.fugerit.java.doc.mod.openrtf.ext.RtfTypeHandler	fj-doc-mod-openpdf-ext	RTF	Generates a RTF document using OpenRTF.

6.2. [fj-doc-freemarker]

To use this doc handler, you will need to add the following dependency :

```
<dependency>
    <groupId>org.fugerit.java</groupId>
    <artifactId>fj-doc-freemarker</artifactId>
    <version>${fj-doc-version}</version>
</dependency>
```

6.2.1. HTML

This doc handler would render a full html document.

Add this element to <**docHandlerConfig**/> :

```
<docHandler id="html-fm" info="html"
type="org.fugerit.java.doc.freemarker.html.FreeMarkerHtmlTypeHandlerEscapeUTF8" />
```

6.2.2. Fragment HTML

This doc handler would render only the body of the html document.
(useful to inject the body as a fragment).

Add this element to <**docHandlerConfig**/> :

```
<docHandler id="html-fragment-fm" info="fhtml"
type="org.fugerit.java.doc.freemarker.html.FreeMarkerHtmlFragmentTypeHandlerEscapeUT
F8" />
```

6.3. [fj-doc-mod-fop] A PDF/FO DocHandler

To use this doc handler, you will need to add the following dependency :

```
<dependency>
    <groupId>org.fugerit.java</groupId>
    <artifactId>fj-doc-mod-fop</artifactId>
    <version>${fj-doc-version}</version>
</dependency>
```

This module is based on Apache FOP Project for rendering and Apache FreeMarker for composing the source document model.

Usually Apache FOP uses XSL Transformations as a mean of customizing the content. But here we follow the templating approach using FreeMarker.

6.3.1. Basic PDF DocHandler

This doc handler would render a full PDF document with default FOP configuration.

Add this element to <**docHandlerConfig**/> :

```
        <!-- Type handler generating pdf -->
        <docHandler id="pdf-fop" info="pdf"
type="org.fugerit.java.doc.mod.fop.PdfFopTypeHandler"/>
```

6.3.2. Custom configured PDF DocHandler

For this DocHandler it is possible to customize some attributes, for instance :

```
        <!-- Type handler generating pdf -->
        <docHandler id="pdf-fop-config" info="pdf"
type="org.fugerit.java.doc.mod.fop.PdfFopTypeHandler">
            <docHandlerCustomConfig charset="UTF-8" fop-config-mode="classloader"
fop-config-classloader-path="fj-doc-quarkus-tutorial/fop-config.xml"
             fop-pool-min="20" fop-pool-max="40" fop-suppress-events="1"/>
        </docHandler>
```

docHandlerCustomConfig reference for org.fugerit.java.doc.mod.fop.PdfFopTypeHandler

name	type	default	description
charset	*string*	*UTF-8*	This will set the charset to use.

name	type	default	description
fop-config-mode	*string*		Custom fop configuration mode, possible values are : *classloader* (path set **fop-config-classloader-path**) or *inline* (child element).
fop-config-classloader-path	*string*		Path to Apache FOP Configuration file.
fop-suppress-events	*boolean*	*false*	If set to *true* (or *1*), will try to suppress event logging (for example : new page).
pdf-a-mode	*string*		If present will set pdf-a-mode, possible values are : *PDF/A-1a*, *PDF/A-1b*, *PDF/A-2a*, *PDF/A-3a*
pdf-ua-mode	*string*		If present, will set pdf-ua-mode, possible values are : *PDF/UA-1*. Partially compatible with pdf-a-mode.
fop-pool-min	*int*	0	If present, it will create a fo user agent pool, this is the minimum size of the pool.
fop-pool-max	*int*	0	If present, it will create a fo user agent pool, this is the maximum size fo the pool.

If **pdf-a-mode** is set, there will be a strict validation of the PDF (i.e. it will be checked if the font are all embedded and the images should comply to PDF/A standard).

6.3.3. PDF/A DocHandler

See also Apache FOP PDF/A.

Here is an example of PDF/A DocHandler with custom fop configuration :

```
        <docHandler id="pdf_a-fop" info="pdf"
type="org.fugerit.java.doc.mod.fop.PdfFopTypeHandler">
            <docHandlerCustomConfig charset="UTF-8" fop-config-mode="classloader"
fop-config-classloader-path="fop-config-pdfa.xml" pdf-a-mode="PDF/A-1b"/>
        </docHandler>
```

And here a sample configuration file.

6.3.4. PDF/UA DocHandler (inline configuration)

```
    <docHandlerConfig registerById="true">
        <docHandler id="pdf_ua-fop" info="pdf"
type="org.fugerit.java.doc.mod.fop.PdfFopTypeHandler">
            <docHandlerCustomConfig charset="UTF-8" fop-config-mode="inline" pdf-ua-
mode="PDF/UA-1">
```

```
                <fop version="1.0">

                  <strict-configuration>true</strict-configuration>
                  <strict-validation>true</strict-validation>
                  <base>.</base>
                  <font-base>.</font-base>
                    <renderers>
                       <renderer mime="application/pdf">
                          <pdf-ua-mode>PDF/UA-1</pdf-ua-mode>
                          <pdf-a-mode>PDF/A-1b</pdf-a-mode>
                          <version>1.4</version>
                       </renderer>
                    </renderers>
                  <!-- Source resolution in dpi (dots/pixels per inch) for
determining the size of pixels in SVG and bitmap images, default: 72dpi -->
                  <source-resolution>72</source-resolution>
                  <!-- Target resolution in dpi (dots/pixels per inch) for
specifying the target resolution for generated bitmaps, default: 72dpi -->
                  <target-resolution>72</target-resolution>
                  <default-page-settings height="11in" width="8.26in"/>
                </fop>
            </docHandlerCustomConfig>
        </docHandler>
```

6.3.5. FO DocHandler

This doc handler would render a full FO intermediate document.

```
        <!-- Type handler generating xls:fo style sheet -->
        <docHandler id="fo-fop" info="fo"
type="org.fugerit.java.doc.mod.fop.FreeMarkerFopTypeHandlerUTF8" />
```

6.4. [fj-doc-mod-poi] : a XLS/XLSX DocHandler

To use this doc handler, you will need to add the following dependency :

```
<dependency>
    <groupId>org.fugerit.java</groupId>
    <artifactId>fj-doc-mod-poi</artifactId>
    <version>${fj-doc-version}</version>
</dependency>
```

This module is based on Apache POI Project.

6.4.1. POI Handler Basics

The following information apply to both xlsx and xls DocHandler.

You will need at least a table, with an *id* set. (in this example the id is **data-table**).

And the excel-table-id element with the comma separated enumeration of the table to render as sheet. :

```
<info name="excel-table-id">data-table=print</info>
```

The given table (**data-table**) will be rendered as the named sheet (**print**) in the couple : **$tableId** =**$sheetName**.

Here is a full example.

```
<?xml version="1.0" encoding="utf-8"?>
<doc
xmlns="http://javacoredoc.fugerit.org"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://javacoredoc.fugerit.org
https://www.fugerit.org/data/java/doc/xsd/doc-2-1.xsd" >
  <metadata>
    <!-- property specific for xls/xlsx -->
    <info name="excel-table-id">data-table=print</info>
  </metadata>
  <body>
        <table columns="3" colwidths="30;30;40"  width="100" id="data-table"
padding="2">
            <row>
                <cell align="center" border-color="#000000" border-width="1"><para
style="bold">Name</para></cell>
                <cell align="center"><para style="bold">Surname</para></cell>
                <cell align="center"><para style="bold">Title</para></cell>
```

```
            </row>
            <row>
                <cell><para><![CDATA[Luthien]]></para></cell>
                <cell><para><![CDATA[Tinuviel]]></para></cell>
                <cell><para><![CDATA[Queen]]></para></cell>
            </row>
            <row>
                <cell><para><![CDATA[Thorin]]></para></cell>
                <cell><para><![CDATA[Oakshield]]></para></cell>
                <cell><para><![CDATA[King]]></para></cell>
            </row>
        </table>
  </body>
</doc>
```

More elements specific to excel format are available here.

6.4.2. Xlsx DocHandler

This doc handler would a XLSX document.

Add this element to <**docHandlerConfig**/> :

```
<!-- XLSX type hanlder  -->
<docHandler id="xlsx-poi" info="xlsx"
type="org.fugerit.java.doc.mod.poi.XlsxPoiTypeHandler" />
```

6.4.3. Xls DocHandler

This doc handler would a XLS document.

Add this element to <**docHandlerConfig**/> :

```
<!-- XLSX type hanlder  -->
<docHandler id="xls-poi" info="xlsx"
type="org.fugerit.java.doc.mod.poi.XlsPoiTypeHandler" />
```

6.5. [fj-doc-mod-opencsv] : a CSV DocHandler

To use this doc handler, you will need to add the following dependency :

```
<dependency>
    <groupId>org.fugerit.java</groupId>
    <artifactId>fj-doc-mod-opencsv</artifactId>
    <version>${fj-doc-version}</version>
</dependency>
```

This module is based on OpenCSV.

This doc handler would a CSV document.

Add this element to <**docHandlerConfig**/> :

```
<!-- CSV type hanlder  -->
<docHandler id="csv-opencsv" info="csv"
type="org.fugerit.java.doc.mod.opencsv.OpenCSVTypeHandler"/>
```

You will need at least a table, with an *id* set. (in this example the id is **data-table**).

And the csv-table-id element with the comma separated enumeration of the table to render as sheet. :

```
<info name="csv-table-id">data-table</info>
```

Here is a full example.

```
<?xml version="1.0" encoding="utf-8"?>
<doc
xmlns="http://javacoredoc.fugerit.org"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://javacoredoc.fugerit.org
https://www.fugerit.org/data/java/doc/xsd/doc-2-1.xsd" >
  <metadata>
    <info name="csv-table-id">data-table</info>
  </metadata>
  <body>
        <table columns="3" colwidths="30;30;40"  width="100" id="data-table"
padding="2">
            <row>
                <cell align="center" border-color="#000000" border-width="1"><para
style="bold">Name</para></cell>
                <cell align="center"><para style="bold">Surname</para></cell>
```

```
                <cell align="center"><para style="bold">Title</para></cell>
            </row>
            <row>
                <cell><para><![CDATA[Luthien]]></para></cell>
                <cell><para><![CDATA[Tinuviel]]></para></cell>
                <cell><para><![CDATA[Queen]]></para></cell>
            </row>
            <row>
                <cell><para><![CDATA[Thorin]]></para></cell>
                <cell><para><![CDATA[Oakshield]]></para></cell>
                <cell><para><![CDATA[King]]></para></cell>
            </row>
        </table>
    </body>
</doc>
```

More elements specific to CSV format are available here.

In comparison to Xlsx Doc Handler the OpenCSV one is able to render only one table at once.
<<<

6.6. [fj-doc-mod-openpdf-ext] : a PDF and HTML DocHandler

To use this doc handler, you will need to add the following dependency :

```
<dependency>
    <groupId>org.fugerit.java</groupId>
    <artifactId>fj-doc-mod-openpdf-ext</artifactId>
    <version>${fj-doc-version}</version>
</dependency>
```

This module is based on OpenPDF (based on a fork of iText).

6.6.1. [fj-doc-mod-openpdf-ext-pdf] : PDF DocHandler

This doc handler would a PDF document.

Add this element to <**docHandlerConfig**/> :

```
<!-- OpenPDF type hanlder  -->
<docHandler id="openpdf" info="openpdf"
type="org.fugerit.java.doc.mod.openpdf.ext.PdfTypeHandler"/>
```

More elements specific to fixed size formats, like PDF, are available here.

docHandlerCustomConfig reference for org.fugerit.java.doc.mod.openpdf.ext.PdfTypeHandler"

name	type	default	description
charset	*string*	*UTF-8*	This will set the charset to use.

Additionaly **fonts** can be configured as child elements of **docHandlerCustomConfig**

name	type	default	description
name	*string*		Name of the font
path	*string*		Path of the font (in classloader of file)

Here is a custom configuration example :

```
<!-- OpenPDF type hanlder  -->
<docHandler id="openpdf" info="openpdf"
type="org.fugerit.java.doc.mod.openpdf.ext.PdfTypeHandler">
    <docHandlerCustomConfig charset="UTF-8">
        <font name="TitilliumWeb" path="font/TitilliumWeb-Regular.ttf"/>
    </docHandlerCustomConfig>
</docHandler>
```

6.6.2. [fj-doc-mod-openpdf-ext-html] : HTML DocHandler

This doc handler would a HTML document.

Add this element to <**docHandlerConfig**/> :

```
<!-- OpenPDF type hanlder  -->
<docHandler id="openpdf-html" info="openpdf-html"
type="org.fugerit.java.doc.mod.openpdf.ext.HtmlTypeHandler"/>
```

6.7. [fj-doc-mod-openrtf-ext] : a RTF DocHandler

To use this doc handler, you will need to add the following dependency :

```
<dependency>
    <groupId>org.fugerit.java</groupId>
    <artifactId>fj-doc-mod-openrtf-ext</artifactId>
    <version>${fj-doc-version}</version>
</dependency>
```

This module is based on OpenRTF.

This doc handler would a RTF document.

Add this element to **<docHandlerConfig/>** :

```
<!-- OpenRTF type hanlder  -->
<docHandler id="openrtf" info="openrtf"
type="org.fugerit.java.doc.mod.openrtf.ext.RtfTypeHandler"/>
```

More elements specific to fixed size formats, like RTF, are available here.

7. Frequently asked questions (FAQ)

This sections contains some answers to frequently asked questions.

7.1. How do I create a new project using Fugerit Venus Doc?

It is possible to simply run the plugin :

```
mvn org.fugerit.java:fj-doc-maven-plugin:init \
-DgroupId=org.example.doc \
-DartifactId=fugerit-demo \
-Dextensions=base,freemarker,mod-fop
```

For more information see : Maven Plugin Goal Init.

7.2. How do I validate the doc format after freemarker processing?

If using the FreemarkerDocProcessConfig configuration
you can simply set to true the attributes :

- validating="true"
- failOnValidate="true" (if you want the generation to fail in case of validation errors)

For instance :

```
<freemarker-doc-process-config
    xmlns="https://freemarkerdocprocess.fugerit.org"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="https://freemarkerdocprocess.fugerit.org
https://www.fugerit.org/data/java/doc/xsd/freemarker-doc-process-1-0.xsd"
    validating="true"
    failOnValidate="true">
```

On log you will find something like :

- DocValidationResult failed!, errors : 2
- Validation error 0, org.xml.sax.SAXParseException; lineNumber: 42; columnNumber: 22; cvc-complex-type.2.4.a: Invalid content was found starting with element '{"http://javacoredoc.fugerit.org":h}'. One of '{"http://javacoredoc.fugerit.org":phrase,

"http://javacoredoc.fugerit.org":para}' is expected.

- Validation error 1, org.xml.sax.SAXParseException; lineNumber: 49; columnNumber: 49; cvc-complex-type.3.2.2: Attribute 'attribute-not-allowed' is not allowed to appear in element 'para'.

For more information see :
FreemarkerDocProcessConfig configuration : Attributes.

7.3. How do I clean source document before parsing to Doc Model?

If using the FreemarkerDocProcessConfig configuration
you can simply set to true the attributes :

- cleanSource="true"

For instance :

```
<freemarker-doc-process-config
    xmlns="https://freemarkerdocprocess.fugerit.org"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="https://freemarkerdocprocess.fugerit.org
https://www.fugerit.org/data/java/doc/xsd/freemarker-doc-process-1-0.xsd"
    cleanSource="true">
```

In case of XML source, invalid characters will be stripped :

\u0009\u000A\u000D\u0020-\uD7FF\uE000-\uFFFD\u10000-\u10FFF

See https://www.w3.org/TR/xml/#charsets and GitHub Issue 213 - An invalid XML character (Unicode: 0x2) was found in the element content

For instance the following XML :

```
<para id="to-clean">test clean \u0002 end test.</para>
```

Would be cleaned to :

```
<para id="to-clean">test clean end test.</para>
```

For more information see :
FreemarkerDocProcessConfig configuration : Attributes.

7.4. How do I have a Excel document cells resize to fit the content.

It is possible to use the elements :

- excel-try-autoresize
- excel-fail-on-autoresize-error

For instance :

```
<info name="excel-try-autoresize">true</info>
<info name="excel-fail-on-autoresize-error">false</info>
```

The DocHandler will try to automatically resize the cell content of the Excel Sheet.