XDocReport Document Rendering Library

Table of Contents

This module (`incode-module-docrendering-xdocreport`) provides an implementation of the Document subdomain module's `Renderer` interface using the xdocreport library module.

API

The module provides two different implementations of `Renderer`:

- `RendererForXDocReportToDocx` which implements `RendererFromBytesToBytes`
  - uses Freemarker to mail-merge a Word template document, returning the `byte[]` of the resultant Word document
- `RendererForXDocReportToPdf` which implements `RendererFromBytesToBytes`
  - almost the same as the previous renderer, but returns a `byte[]` array of a PDF rather than a Word document.

These classes can be used as the `Renderer` implementation for a Document `RenderingStrategy`. Subclasses of the `RenderingStrategyFSAbstract` fixture script can be used to create such an entity, eg:

- `RenderingStrategyFSForXDocReportToDocx`

```
public class RenderingStrategyFSForXDocReportToDocx extends
RenderingStrategyFSAbstract {
    public static final String REF = "XDD";
    @Override
    protected void execute(ExecutionContext executionContext) {
        upsertRenderingStrategy(
                REF,
                "XDocReport to .docx",
                DocumentNature.BYTES,
                DocumentNature.BYTES,
                RendererForXDocReportToDocx.class, executionContext);
    }
}
```

- `RenderingStrategyFSForXDocReportToPdf`

```
public class RenderingStrategyFSForXDocReportToPdf extends
RenderingStrategyFSAbstract {
    public static final String REF = "XDP";
    @Override
    protected void execute(ExecutionContext executionContext) {
        upsertRenderingStrategy(
                REF,
                "XDocReport to .pdf",
                DocumentNature.BYTES,
                DocumentNature.BYTES,
                RendererForXDocReportToPdf.class, executionContext);
    }
}
```

The document subdomain module also allows `RenderingStrategy`s to be created from the UI; it will "discover" all `Renderer` implementations from the classpath.

How to configure/use

Classpath

Update your classpath by adding this dependency in your dom project's `pom.xml`:

```
<dependency>
    <groupId>org.incode.example.docrendering</groupId>
    <artifactId>incode-example-docrendering-xdocreport-dom</artifactId>
    <version>1.16.1</version>
</dependency>
```

Check for later releases by searching Maven Central Repo.

For instructions on how to use the latest `-SNAPSHOT`, see the contributors guide.

Bootstrapping

In the `AppManifest`, update its `getModules()` method, eg:

```
@Override
public List<Class<?>> getModules() {
    return Arrays.asList(
            ...
            org.incode.example.docrendering.xdocreport.dom
.XdocReportDocRenderingModule.class,
    );
}
```

Known issues

None known at this time.

Dependencies

Maven can report modules dependencies using:

```
mvn dependency:list -o -pl modules/dom/docrendering-xdocreport/impl -D
excludeTransitive=true
```

which, excluding the Incode Platform and Apache Isis itself, returns no direct compile/runtime dependencies.

From the Incode Platform it uses:

- base library module
- freemarker library module
- document example subdomain module