# File Generator

## Overview

The File Generator application will guide the user through the process of generating files for the system.

The user will navigate to the applications homepage in a browser, where they will be presented with various options to define the type of files that get generated.

Once the relevant options have been selected, the user will click a generate button to start the file generation process.

The options are then sent to a REST service endpoint which will generate the specified files on the server.

The user then clicks the Create Media 1 button.

This request is handled by invoking another REST service endpoint which will guide the user through the media creation process.

The user then clicks the Create Media 2 button.

This request is handled by invoking another REST service endpoint which will guide the user through the media creation process.

## Purpose

The objective in building the application in this way, is to prove that the overall structure of the implementation meets the needs of the customer. Further development can then follow, for example

1. Add in secondary user authorisation
2. Add in specific file generation capability
3. Add in unit testing
4. Add in integration testing
5. Final release build process

Once the overall application structure has been proven, it should provide a useable template for the final production version.

## Implementation

The File Generator diagram below shows how the various elements of the application interact.

The REST Service is implemented in Java using Spring 5 REST Services.

The File generation is done by code in an external jar. For the purposes of this early version of the application an appropriate representative jar will be created. The representative jar will have the same basic functionality as the production jar but will not generate the actual data content required.

The UI aspects will be created using reactJS and Googles Material UI (MUI) components.

The REST endpoints will invoke CDInterface (Powershell) via ProcessBuilder to handle the interface to the CD Writer.

## Detail

A number of separate components are needed to create this application, as follows

1. An executable jar that can be called to generate a number of different file sets
2. A React UI to provide User interaction
3. A Spring Server to provide the REST endpoints
4. CDInterface (Already developed)

## Wireframe

In the initial version the application UI will be

Text

Description automatically generated with medium confidence

Diagram

Description automatically generated