# Backend

## SpringApplication class

This is the main class of this application. To run the server, this main class needs to be run.

## Controller class

The Controller class provides the frontend REST API, with which it receives HTTP Request on port 8080 and calls methods from the Service classes to process them. The allowed HTTP-Methods are GET,POST,PUT,DELETE.

### Uploading document:

> POST -> /upload/{document_name} * saves or updates the text file in the request's body as a document and its related SpecItem. * expects a multipart request and the text file with RequestParam "file" * sends status code 201 and success message back if successful * sends status code 400 back if the text file's format is not correct

## Importer classes

## File generator

The `filegenerator` package consists of a series of interfaces and the corresponding implementations for the file generation. The main interface, which acts as an entry point for the generation process, is named `FileGenerator` and exposes the methods for creating a file. Each procedure takes the name of the target file and the number of spec items to be made. An additional method accepts a parameter that describes the desired number of incomplete spec items to be created. This method gives the possibility create only a specific number of attributes of a spec item, which are chosen at random. The generation process follows a set of rules to create files that have a rigid structure. The following block depicts an exemplary file that follows the format:

```
 CommitHash: 29ec1a67-b329-
CommitDate: 2022-11-29 16:15:53
CommitMsg: oz8q4u3seczhm1n4tt8j
CommitAuthor: Williams Homenick




Fingerprint: 61x8knj3353789wy
ShortName: SpecItem_0
Category: Category3
LC-Status: Status4
UseInstead: SpecItem_32
TraceRefs: SpecItem_30,SpecItem_62,SpecItem_58,SpecItem_59,SpecItem_99,SpecItem_72
LongName: corrupti autem esse sit architecto repudiandae aut
Content: totam]}]magnam][]qui](]ad]ipsam]iure]molestiae]||])]![]{]\t]*]explicabo]-]sint];]ipsam]iusto]e
```

On the top of each generated file, there is commit information. All components of a commit but the commit date are randomly generated. The interface `CommitProvider` exposes a single method called `generateCommit`, which creates a `Commit` object populated with the necessary information. After the commit information, a given number of spec items is appended. Spec items are generated via `SpecItemProvider`. The content of each spec item is generated via `ContentCreator`. `ContentCreator` generates highly randomized content of various complexity. The created content can span multiple lines and have many different

characters. The created structure is written to a .txt file with the help of `WriterService`. The writer service also ensures the correct formatting of a file (number of spaces and new lines).

## Service classes

Here the applications and business logics are implemented.

### SpecItemService

`void savesDocument(String filename)` save or update a document in database

### FileStorageService

`void storeFile(MultipartFile file, String filename)` save a multipart file in /src/main/resources/tmp

## Models

The models are objects that are to be stored in database. * Specitem: This is the main objects that we want to store. * Document: A document contains many specitems and user updates or add new specitems through it * Commit: contains commit information and allows versioning mechanism

## Repositories

Repositories serve as connector to the database and provide developers ways to query and save data. There is one repo for each model.

# Frontend

We use React Router to navigate between pages and components.

## Components

- main_page: This component represents the home page (/).
- specitems_page: This component represents the page where all specitems are shown and filtered (/specitems). It calls GET method to the server to get a (filtered) list of specitems. This component receives exportContext from its parent main_page, which relates to the list of to be exported specitems in the next component.
- export_page: This component represents the page where users can export saved specitems to downloadable text file (/export). The specitems that are saved in specitems_page component are forwarded by its parent main_page to here.