РУКОВОДСТВО ПРОГРАММИСТА

программной библиотеки визуализации для визуализации электронных документов, подписанных электронной подписью, в утвержденных ФНС России форматах

И5

Листов 35

Москва 2024

Содержание

1. Описание программного интерфейса. 4
2. Настройка среды для работы с библиотекой из консоли. 4
2.1. Установка OpenJDK 11 4
2.2. Вызов библиотеки из консоли. 5
2.2.1 Генерация pdf-файла 5
2.2.2 Вывод списка шаблонов 5
2.2.3 Проведение проверки файлов по xsd и sch 6
2.2.4 Проведение проверки файлов по шаблону. 7
2.2.5 Вывод версии библиотеки 8
2.2.6 Вывод вспомогательной информации библиотеки 8
2.2.7 Вывод содержимого файла readme 9
3. Подключение библиотеки как зависимость. 9
3.1. Установка OpenJDK 11 9
3.2. Установка Maven. 9
3.3. Настройка окружения. 10
3.4. Настройка проекта. 11
3.5. Подключение к проекту. 11
4. Описание основных методов библиотеки. 12
4.1. Методы класса Templater: 12
4.1.1. Статический метод of 12
4.1.2. Метод convertS 13
4.1.3. Метод convertF 14
4.1.4. Метод convertSForResolver 14
4.1.5. Метод convert 15
4.1.6. Метод convertToByte 16
4.1.7. Метод convertFilesToByteArray 17
4.1.8. Метод convertToTemplateDataSource 18
4.1.9. Метод convertFilesToTemplateDataSource 18
4.1.10. Метод getCallback 19
4.1.11. Метод resolver 20
4.1.12. Метод registry 20
4.1.13. Метод getWarnings 21
4.1.14. Метод getNeedWatchSvedFiles 21
4.1.15. Метод deleteTempFiles 22
4.2. Методы интерфейса CheckerTemplater: 23

4.2.1. Метод list 23
4.3. Методы при подключении библиотеки в пакетных операциях: 23
4.3.1. Статический метод main 23
4.3.2. Функционал генерации pdf-файла 25
4.3.3. Проведение проверки файлов по xsd и sch 27
4.3.4. Функционал проверки файлов 27
4.3.5. Функционал вывода описания библиотеки 28
4.3.6. Функционал вывода списка подключенных шаблонов 28
4.4. Описание добавления вложенных файлов 29
4.4.1. Описание метода convert 29
4.4.2. Описание метода removeAttachment 30
4.5. Описание общих объектов данных 31
4.5.1. Описание структуры объекта 31
5. Исключения 34
5.1. Генерируемые исключения статического метода main «Генерации pdf файла» 34
5.2. Генерируемые исключения метода convertS «Генерации потока pdf файла» 34
5.3. Генерируемые исключения метода list 35

1. Описание программного интерфейса.

Библиотека создана для преобразование файлов электронных документов (XML/PDF) и электронных подписей к ним (при наличии, SGN, SIGN, P7S) на основе шаблонов печатных форм (SVED) в электронный документ формата PDF.

Для работы библиотеки необходимы некоторые параметры, который описаны ниже в пункте 2.2.

Библиотека может быть подключена к проекту как зависимость либо вызвана напрямую из консоли.

2. Настройка среды для работы с библиотекой из консоли.

Пример настройки среды будет показан на примере **Ubuntu 20.04.**

2.1. Установка OpenJDK 11.

Установить JDK 11 командой:

```
sudo apt -y install openjdk-11-jdk
```

После этого необходимо переключится на установленную версию:

```
sudo update-alternatives --config java
```

В первой строчке указывается версия java, которая используется на данный момент. Введите необходимую цифру для выбора Java 11. В данном примере на Рис.1 случае будет выбрана цифра 2:

```
There are 3 choices for the alternative java (providing /usr/bin/java).

  Selection    Path                                            Priority   Statu
s
------------------------------------------------------------
  0            /usr/lib/jvm/java-11-openjdk-amd64/bin/java      1111      auto
mode
* 1            /usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java   3         manua
l mode
  2            /usr/lib/jvm/java-11-openjdk-amd64/bin/java      1111      manua
l mode
  3            /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java   1081      manua
l mode

Press <enter> to keep the current choice[*], or type selection number:
```

Рис.1

Убедиться в наличии и работоспособности Java командой:

```
java -version
```

Пример ответа:

```
# java -version
openjdk version "11.0.11" 2023-10-11
OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-Oubuntu2.20.04)
OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubuntu2.20.04, mixed mode, sharing)
```

2.2.Вызов библиотеки из консоли.

2.2.1 Генерация pdf-файла

Команда:

java –jar template-extend-lib.jar <маршрут поиска шаблонов> <маршрут выгрузки результирующего файла> <маршрут входных данных>

Возможные результаты:

1. Сгенерированный pdf-файл;
2. Генерация исключения.

2.2.2 Вывод списка шаблонов

Команда:

java –jar template-extend-lib.jar <маршрут поиска шаблонов> --list

Возможные результаты:

- Список подключенных шаблонов по пути: <путь указанный при вызове>;

- Мета информация по шаблону:

№	Категория	Описание
1	Наименование	Тип документа
2	Статус	Статус
3	Наименование, заданное при создании	Наименование, заданное автором
4	Формат	Формат документа

5	Редакция	Номер редакции
6	Дата создания	Дата, когда был создан шаблон
7	Дата публикации	Дата, когда документ был передан на публикацию и опубликован без ошибок

2.2.3 Проведение проверки файлов по xsd и sch

Команда:

java –jar template-extend-lib.jar <маршрут поиска проверяемых файлов> <маршрут выгрузки результирующего файла>

Возможные результаты:

Сгенерированный файл отчёта в формате json:

№	Категория	Описание
1	xsdSchName	Наименования документа проверки (xsd/sch)
2	xsdSchSha	Контрольная сумма документа проверки
3	xmlName	Наименование проверяемого документа
4	xmlSha	Контрольная сумма проверяемого документа
5	validationTime	Контрольная сумма проверяемого документа
6	validationDate	Дата проведения проверки в формате дата время и смещение по UTC (25.10.2024T14:30:15+03:00)
7	fatalErrors	Критические ошибки, найденные при проверке
8	errors	Ошибки, найденные при проверке

9	warnings	Предупреждения, найденные при проверке
10	message	Сообщение об ошибке
11	location	Тэг ошибки в файле

Примечание:

В директории *<маршрут поиска проверяемых файлов>* должны быть расположены файл xml и xsd/файл схематрона в единственном экземпляре.

2.2.4 Проведение проверки файлов по шаблону.

Команда:

java –jar template-extend-lib.jar <маршрут поиска шаблонов> <маршрут выгрузки результирующего файла> <маршрут входных данных> -v

Возможные результаты:

Сгенерированный файл отчёта в формате json:

№	Категория	Описание
1	xsdSchName	Наименования документа проверки (xsd/sch)
2	xsdSchSha	Контрольная сумма документа проверки
3	xmlName	Наименование проверяемого документа
4	xmlSha	Контрольная сумма проверяемого документа
5	validationTime	Контрольная сумма проверяемого документа
6	validationDate	Дата проведения проверки в формате дата время и смещение по UTC (25.10.2024T14:30:15+03:00)
7	fatalErrors	Критические ошибки, найденные при проверке

8	errors	Ошибки, найденные при проверке
9	warnings	Предупреждения, найденные при проверке
10	message	Сообщение об ошибке
11	location	Тэг ошибки в файле

Примечание:

Обратите внимание на параметр в конце строки вызова метода проведения проверки по шаблону, без этого параметра будет проведена визуализация документа.

2.2.5 Вывод версии библиотеки

Команда:

java –jar template-extend-lib.jar <маршрут поиска шаблонов/ -> --version

Возможные результаты:

Выводится версия библиотеки:

Версия программного обеспечения: <номер версии библиотеки>

Примечание:

При вызове параметра *version* без указания маршрута поиска шаблонов, в месте вызова библиотеки должна быть директория с шаблонами.

2.2.6 Вывод вспомогательной информации библиотеки

Команда:

java –jar template-extend-lib.jar <маршрут поиска шаблонов/ -> --help

Возможные результаты:

Выполняются все вспомогательные команды, которые доступны при использовании библиотеки:

--version

--list

--readme

При вызове параметра *help* без указания маршрута поиска шаблонов, в месте вызова библиотеки должна быть директория с шаблонами.

2.2.7 Вывод содержимого файла readme

Команда:

java –jar template-extend-lib.jar --readme

Возможные результаты:

Содержание файла **readme**.

При вызове параметра *readme* без указания маршрута поиска шаблонов, в месте вызова библиотеки должна быть директория с шаблонами.

3. Подключение библиотеки как зависимость.

Пример установки и настройки **Java** и **Maven** будет показан на примере **Ubuntu 20.04.**

3.1. Установка OpenJDK 11.

Установить JDK 11 командой:

```
sudo apt -y install openjdk-11-jdk
```

После этого необходимо переключится на установленную версию:

```
sudo update-alternatives --config java
```

Убедиться в наличии и работоспособности Java командой:

```
java -version
```

Пример ответа:

```
# java -version
openjdk version "11.0.11" 2023-10-11
OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-Oubuntu2.20.04)
OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubuntu2.20.04, mixed mode, sharing)
```

3.2. Установка Maven.

Установите Maven, введя в свой терминал следующую команду:

```
sudo apt install maven
```

В случае успешной установки команда

```
mvn --version
```

Результат должен выглядеть примерно так:

```
Apache Maven 3.6.3

Maven home: /usr/share/maven

Java version: 1.8.0_382, vendor: Private Build, runtime:/usr/lib/jvm/java-11-openjdk-amd64/jre

Default locale: ru_RU, platform encoding: UTF-8

OS name: "linux", version: "6.2.0-33-generic", arch: "amd64", family: "unix"
```

3.3. Настройка окружения.

Обращаем внимание, что система сборки Maven изначально настроена на работу с OpenJDK 8.

Настройка переменных окружения для Java 11 и Maven:

Для использования Maven необходимо настроить переменные окружения. Создайте файл /etc/environment.d/100-maven.conf, для этого введите следующую команду:

```
sudo vi /etc/environment.d/100-maven.conf
```

и добавьте открывшийся файл следующие строки:

```
JAVA_HOME=/usr/lib/jvm/default-java

M2_HOME=/opt/maven

MAVEN_HOME=/opt/maven

PATH=/opt/maven/bin:${PATH}
```

Эти строки добавят директории Java в переменную окружения PATH, позволяя системе находить исполняемые файлы Java при вызове команд.

Для применения изменений необходимо выйти из текущей сессии, введя команду:

```
exit
```

и зайти в терминал снова.

3.4. Настройка проекта.

При создании проекта убедитесь, что используемая вами среда разработки использует JDK 11. В примере будет показана проверка настроек проекта в IntelliJ IDEA.

- Нажать правую кнопку мыши на корневой папке проекта.
- Выбрать пункт Open Modules Settings.
- Проверить что в разделах Project и SDKs выбрана установленная JDK 11.
- В правой части приложения нажать Maven.
- Нажать на значок настроек и выбрать пункт Maven Settings, как показано на Рис.2.
- В открывшемся пункте проверить версию Maven, она должна совпадать или быть или выше версией из п. 3.2.
- В подпунктах Importing и Runner проверить версию JDK 11.

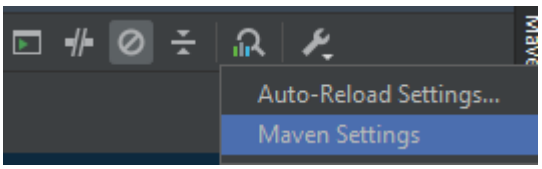

Рис.2

3.5. Подключение к проекту.

Для установки библиотеки как зависимости в проект, выполните следующие шаги:

- Создать в корневой папке проекта директорию для библиотеки, в примере создана папка libs.
- Скопировать jar файл библиотеки в созданную папку.
- Откройте файл pom.xml вашего проекта.
- Добавьте следующую зависимость вашего проекта:

```
<dependency>

    <groupId>ru.kontur.fnsviewer</groupId>

    <artifactId>templater-extend-lib</artifactId>

    <version>0.0.0</version>

    <scope>system</scope>

    <systemPath>${project.basedir}/libs/templater-extend-lib.jar </systemPath>

</dependency>
```

Примечание.

Для использования библиотеки в **Spring Boot** приложении необходимо дополнительно добавить следующие строки в файл pom.xml.

```
<plugin>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-maven-plugin</artifactId>
    <configuration>
      <includeSystemScope>true</includeSystemScope>
    </configuration>
</plugin>
```

4. Описание основных методов библиотеки.

Библиотека template-extend-lib предназначена для упрощения работы с документами. Основные функции библиотеки включают валидацию и визуализацию документов. Templater — основной класс библиотеки template-extend-lib, упакованной в jar-файл. Он служит для работы с файлами, необходимыми для визуализации документов, и предоставляет основные методы взаимодействия с библиотекой.

4.1.Методы класса Templater:

Класс **Templater** содержит основные методы для работы с файлами. Методы доступны только при подключении библиотеки в качестве зависимости.

4.1.1. Статический метод of

Краткое описание метода:

Метод необходим для обработки аргументов, приходящих в основной метод main (метод описан в пункте 2.3.1.), и дальнейшего их использования. Метод of создаёт экземпляр класса Templater. В случае отсутствия или не соответствии аргументов метод выводит исключение.

Примечание:

Метод может принимать **несколько** аргументов, но работу осуществляет только с первым и вторым.

Входные аргументы:

№	Тип	Обязательность	Описание
1	String	O	Путь до расположения конкретного шаблона xslx / расположения директории наборов sved
2	String	O	Путь до хранения результатов в формате pdf

Выходные аргументы:

№	Тип	Описание
1	Templater	Инициализация класса Templater

Примеры вызова на языке java:

```
String template = "/path/to/template";

String output = "/path/to/result";

String inputPath = "/path/to/input-data";

Templater templater = Templater.of(new String[]{template, output, inputPath, "-v"});
```

4.1.2. Метод convertS

Краткое описание метода:

Создание потока из сгенерированного pdf-файла и возвращение его (потока) в вызывающую систему.

Входные аргументы:

№	Тип	Обязательность	Описание
1	List<String>	O	Список путей до расположения входных аргументов

Выходные аргументы:

№	Тип	Описание
1	Optional<InputStream>	Созданный поток из сгенерированного pdf-файла

Результатом работы данного метода является поток из результирующего pdf-файла.

Пример вызова на языке java:

```
List<String> filePaths = Arrays.asList("path/to/file1.xml", "path/to/file2.xml");
Optional<InputStream> resultStream = convertS(filePaths);
```

4.1.3. Метод convertF

Краткое описание метода:

Метод convertF конвертирует список файлов в список пар потока байт файлов и самих переданных файлов для визуализации и передаёт их в метод convert.

Входные аргументы:

№	Тип	Обязательность	Описание
1	List<File> files	О	Список файлов, необходимых для визуализации

Выходные аргументы:

№	Тип	Описание
1	Optional<InputStream>	Созданный поток из сгенерированного pdf-файла

Пример вызова на языке java:

```
List<File> files = Arrays.asList(new File("path/to/file1.xml"), new File("path/to/file2.xml"));
Optional<InputStream> resultStream = convertF(files);
```

4.1.4. Метод convertSForResolver

Краткое описание метода:

Метод обрабатывает список путей к файлам (filesPath), возвращая список объектов File, соответствующих файлам в этих путях.

Входные аргументы:

№	Тип	Обязательность	Описание
1	final List<String> filesPath	O	Список путей до файлов, необходимых для визуализации

Выходные аргументы:

№	Тип	Описание
1	List<File>	Список файлов, соответствующих переданным путям

Пример вызова на языке java:

```
List<String> filesPath = Arrays.asList("path/to/file1.xml", "path/to/folder",
"path/to/file2.xml");

List<File> resolvedFiles = convertSForResolver(filesPath);
```

4.1.5. Метод convert

Краткое описание метода:

Метод создаёт файл PDF на основе переданных байтов и выводит его в файл в указанной директории. В конце он возвращает поток из созданного pdf-файла, если он создан.

Входные аргументы:

№	Тип	Обязательность	Описание
1	List<Map.Entry<ByteArrayInputStream, File>> bytes	O	Список пар потока байт файлов и самих переданных файлов для визуализации

Выходные аргументы:

№	Тип	Описание
1	Optional<InputStream>	Созданный поток из сгенерированного pdf-файла

Пример вызова на языке java:

```
List<Map.Entry<ByteArrayInputStream, File>> byteFiles = List.of (
Map.entry(new ByteArrayInputStream(Files.readAllBytes(Path.of("path/to/file1.xml"))),
new File("path/to/file1.xml")),
Map.entry(new ByteArrayInputStream(Files.readAllBytes(Path.of("path/to/file2.xml"))),
new File("path/to/file2.xml"))
);
Optional<InputStream> pdfStream = convert(byteFiles);
```

4.1.6. Метод convertToByte

Краткое описание метода:

Метод создаёт массив байтов, представляющих содержимое pdf-файла, на основе переданных байтов и выводит его в файл в указанной директории. В конце он возвращает поток ввода из созданного файла, если файл создан

Входные аргументы:

№	Тип	Обязательность	Описание
1	List<Map.Entry<ByteArrayInputStream, File>> bytes	O	Список пар потока байт файлов и самих переданных файлов для визуализации

Выходные аргументы:

№	Тип	Описание
1	Optional<byte[]>	Массив байтов, представляющих содержимое pdf-файла

Пример вызова на языке java:

```
List<Map.Entry<ByteArrayInputStream, File>> byteFiles = List.of( Map.entry(new ByteArrayInputStream(Files.readAllBytes(Path.of("path/to/file1.xml"))), new File("path/to/file1.xml")),
Map.entry(new ByteArrayInputStream(Files.readAllBytes(Path.of("path/to/file2.xml"))), new File("path/to/file2.xml"))
);
Optional<byte[]> pdfBytes = convertToByte(byteFiles);
```

4.1.7. Метод convertFilesToByteArray

Краткое описание метода:

Метод преобразует список файлов в список пар потока байт файлов и самих переданных файлов.

Входные аргументы:

№	Тип	Обязательность	Описание
1	List<File> files	О	Список файлов, необходимых для визуализации

Выходные аргументы:

№	Тип	Описание
1	Optional<byte[]>	Массив байтов, представляющих содержимое pdf-файла

Пример вызова на языке java:

```
List<File> files = List.of( new File("path/to/file1.xml"), new File("path/to/file2.xml") );
```

```
Optional<byte[]> pdfBytes = convertFilesToByteArray(files);
```

4.1.8. Метод convertToTemplateDataSource

Краткое описание метода:

Метод предназначен для конвертации файлов в формат данных шаблона (TemplateDataSource) на основе извлечённых данных, связанных с содержимым документа.

Входные аргументы:

№	Тип	Обязательность	Описание
1	List<Map.Entry<ByteArrayInputStream, File>> bytes	О	Список пар потока байт файлов и самих переданных файлов

Выходные аргументы:

№	Тип	Описание
1	Optional<TemplateDataSource>	Возвращает объект TemplateDataSource

Пример вызова на языке java:

```
List<Map.Entry<ByteArrayInputStream, File>> byteStreamFiles = List.of( Map.entry(new ByteArrayInputStream(Files.readAllBytes(Path.of("path/to/file1.xml"))), new File("path/to/file1.xml")), Map.entry(new ByteArrayInputStream(Files.readAllBytes(Path.of("path/to/file2.xml"))), new File("path/to/file2.xml")) );

Optional<TemplateDataSource> templateDataSource = convertToTemplateDataSource (byteStreamFiles);
```

4.1.9. Метод convertFilesToTemplateDataSource

Краткое описание метода:

Метод преобразует список пар потока байт файлов и самих переданных файлов и передаёт в метод convertToTemplateDataSource.

Входные аргументы:

№	Тип	Обязательность	Описание
1	List<File> files	О	Список файлов

Выходные аргументы:

№	Тип	Описание
1	Optional<TemplateDataSource>	Объект в формате данных шаблона TemplateDataSource

Пример вызова на языке java:

```
List<File> files = List.of( new File("path/to/file1.xml"), new File("path/to/file2.xml") );
Optional<TemplateDataSource> templateDataSource =
convertFilesToTemplateDataSource(files);
```

4.1.10. Метод getCallback

Краткое описание метода:

Этот метод возвращает объект callback

Входные аргументы:

№	Тип	Обязательность	Описание
1	-	-	Метод не принимает аргументов

Выходные аргументы:

№	Тип	Описание
1	Function	Функция обратного вызова, связанная с текущей операцией

Пример вызова на языке java:

```
Function callbackFunction = getCallback();
```

4.1.11. Метод resolver

Краткое описание метода:

Этот метод создаёт объект типа DocumentsProcessor, который используется для обработки документов

Входные аргументы:

№	Тип	Обязательность	Описание
1	-	-	Метод не принимает аргументов

Выходные аргументы:

№	Тип	Описание
1	DocumentsProcessor	Экземпляр DocumentsProcessor для обработки документов с настройками криптосервиса и зарегистрированными модуля

Пример вызова на языке java:

```
DocumentsProcessor processor = resolver();
```

4.1.12. Метод registry

Краткое описание метода:

Этот метод создаёт и возвращает объект DocumentModulesRegistry, который представляет собой реестр модулей для обработки документов.

Входные аргументы:

№	Тип	Обязательность	Описание
1	-	-	Метод не принимает аргументов

Выходные аргументы:

№	Тип	Описание
1	DocumentModulesRegistry	Реестр модулей для обработки документов с зарегистрированным шаблонным модулем

Пример вызова на языке java:

```
DocumentModulesRegistry registry = registry();
```

4.1.13. Метод getWarnings

Краткое описание метода:

Этот метод возвращает объект warnings типа WarningsCollector. Он используется для хранения и управления предупреждениями, которые могут возникнуть во время обработки документов.

Входные аргументы:

№	Тип	Обязательность	Описание
1	-	-	Метод не принимает аргументов

Выходные аргументы:

№	Тип	Описание
1	WarningsCollector<DocumentProcessingWarning>	Коллекция предупреждений при обработке документов

Пример вызова на языке java:

```
WarningsCollector<DocumentProcessingWarning> warningsCollector = getWarnings();
```

4.1.14. Метод getNeedWatchSvedFiles

Краткое описание метода:

Этот метод возвращает объект AtomicBoolean, который используется для индикации необходимости отслеживания файлов типа .sved.

Входные аргументы:

№	Тип	Обязательность	Описание
1	-	-	Метод не принимает аргументов

Выходные аргументы:

№	Тип	Описание
1	AtomicBoolean	Флаг, указывающий, требуется ли отслеживание файлов .sved

Пример вызова на языке java:

```
AtomicBoolean watchSvedFilesFlag = getNeedWatchSvedFiles();
 if (watchSvedFilesFlag.get()) {
System.out.println("Отслеживание файлов .sved включено.");
} else {
System.out.println("Отслеживание файлов .sved выключено.");
}
```

4.1.15. Метод deleteTempFiles

Краткое описание метода:

Этот метод удаляет временные файлы, созданные в процессе обработки документов.

Входные аргументы:

№	Тип	Обязательность	Описание
1	-	-	Метод не принимает аргументов

Выходные аргументы:

№	Тип	Описание
1	-	Метод не возвращает значения (void).

Пример вызова на языке java:

```
tmpDir = Files.createTempDirectory("exampleTempDir");
Path tempFile = Files.createFile(tmpDir.resolve("tempFile_" + UUID.randomUUID() + ".txt"));
deleteTempFiles();
```

4.2.Методы интерфейса CheckerTemplater:

4.2.1. Метод list

Краткое описание метода:

Метод выводит список шаблонов sved формата, находящихся в переданной ему (методу) папке.

Входные данные:

№	Тип	Обязательность	Описание
1	String	O	Путь до расположения директории наборов sved

Выходные данные:

№	Тип	Описание
1	List<Parameters>	Список доступных шаблонов

Исключения:

№	Тип	Описание
1	SvedLibraryPathIsEmpty	Генерация исключения если заданный параметр который определяет место хранения файлов формата sved не задан/пуст или путь не существует в файловой системе.

4.3.Методы при подключении библиотеки в пакетных операциях:

4.3.1. Статический метод main

Краткое описание метода:

Основной метод класса Templater имеет несколько сценариев использования. Сценарий использования зависит от аргументов, приходящих в метод.

1) Сценарий генерации pdf-файла на основе шаблона.

При передаче аргументов, описанных ниже, метод main позволяет создать pdf-файл и сохранить его в указанную папку.

Входные аргументы:

№	Тип	Обязательность	Описание
1	String	O	Путь до расположения конкретного шаблона xslx / расположения директории наборов sved
2	String	O	Путь до хранения результатов в формате pdf
3	String	O	Путь до расположения входных данных

* Здесь и далее условное обозначение обязательности параметров:

O – обязательный.

H – необязательный.

Метод является void методом и ничего не возвращает. Результатом работы данного метода является pdf-файл, сохраненный в директорию, на которую указали вторым параметром в вызове.

Пример вызова:

java -jar template-extend-lib.jar path/to/templates path/to/result path/to/input/data

2) Проведение проверки файлов по xsd и sch

При передаче аргументов, описанных ниже, метод main позволяет создать файл, содержащий отчёт по проверке файлов xml на основе xsd и sch в формате json. Данные, предоставляемые в отчёте, описаны в пункте 4.

Входные аргументы:

№	Тип	Обязательность	Описание
1	String	O	Путь до папки, в которой расположен xml и схема по которой осуществляется проверка (файл xsd или файл схематрона)
2	String	O	Путь до хранения результатов в формате json

В этом случае результатом работы данного метода является json-файл с отчётом о проверки xml файла на соответствие схеме.

Пример вызова:

java -jar template-extend-lib.jar path/to/files path/to/result

3) Проведение проверки на основе шаблона.

При передаче аргументов, описанных ниже, метод main позволяет создать файл, содержащий отчёт по проверке файлов xml на основе шаблона в формате json. Данные, предоставляемые в отчёте, описаны в пункте 4.

Входные аргументы:

№	Тип	Обязательность	Описание
1	String	O	Путь до расположения конкретного шаблона xslx / расположения директории наборов sved
2	String	O	Путь до хранения результатов в формате pdf
3	String	O	Путь до расположения входных данных
3	String	O	Обязательный параметр -v

* Здесь и далее условное обозначение обязательности параметров:

- O – обязательный.
- H – необязательный.

Также необходимо указать параметр -v для проведения проверки файлов по шаблону, в противном случае будет проведена визуализация.

Пример вызова:

java -jar template-extend-lib.jar path/to/templates path/to/result path/to/input/data -v

4.3.2. Функционал генерации pdf-файла

Краткое описание функционала:

Позволяет в терминальном/командном/пакетном вызове сгенерировать pdf-файл на основе переданных параметров

Входные данные:

№	Тип	Обязательность	Описание
1	String	O	Путь до расположения конкретного шаблона xslx / расположения директории наборов sved
2	String	O	Путь до расположения входных данных
3	String	O	Путь до хранения результатов в формате pdf

* Здесь и далее условное обозначение обязательности параметров:

- O – обязательный.

Сообщения об ошибках:

№	Тип	Описание
1	BreakingJar	Поврежден jar файл библиотеки.
2	EmptyArgs	Генерация исключения при проверки переданных параметров если количество параметров меньше двух
3	EmptyListFile	> Пустой список параметров при передачи. Передайте: > первым параметром путь к набору шаблонов > вторым параметров путь где будет сохранен результирующий pdf файл > последующие параметры указывают пути к xml и sgn файлам.
4	SvedLibraryPathIsEmpty	Заданный параметр который определяет место хранения файлов формата sved не задан/пуст или путь не существует на файловой системе (или входной шаблон xlsx)
5	SaveResultPdfException	Ошибка сохранения результирующего файла на стороне библиотеки
6	XmlValidationException	Ошибка валидации XML-файла по XSD-схеме

Результатом работы данного метода является pdf-файл сохраненный в директорию на которую указали третьим параметром в вызове.

Пример вызова:

java -jar template-extend-lib.jar path/to/templates path/to/result path/to/input/data

4.3.3. Проведение проверки файлов по xsd и sch

Краткое описание функционала:

Позволяет в терминальном/командном/пакетном вызове провести проверку файлов по xsd и sch

Входные аргументы:

№	Тип	Обязательность	Описание
1	String	O	Путь до папки, в которой расположен xml и схема по которой осуществляется проверка (файл xsd или файл схематрона)
2	String	O	Путь до хранения результатов в формате json

В этом случае результатом работы данного метода является json-файл с отчётом о проверки xml файла на соответствие схеме.

Пример вызова:
java -jar template-extend-lib.jar path/to/files path/to/result

4.3.4. Функционал проверки файлов

Краткое описание функционала:

Позволяет в терминальном/командном/пакетном вызове провести проверку файлов по шаблону

Входные аргументы:

№	Тип	Обязательность	Описание
1	String	O	Путь до расположения конкретного шаблона xslx / расположения директории наборов sved
2	String	O	Путь до хранения результатов в формате pdf
3	String	O	Путь до расположения входных данных
3	String	O	Обязательный параметр -v

* Здесь и далее условное обозначение обязательности параметров:

- О – обязательный.
- Н – необязательный.

Также необходимо указать параметр -v для проведения проверки файлов по шаблону, в противном случае будет проведена визуализация.

Результатом работы данного метода является файл отчёта формата json.

Пример вызова:
java -jar template-extend-lib.jar path/to/templates path/to/result path/to/input/data -v

4.3.5. Функционал вывода описания библиотеки

Краткое описание метода:

Позволяет в терминальном/командном/пакетном вызове отобразить список подключенных шаблонов sved формата.

Входные данные:

№	Тип	Обязательность	Описание
1	String	О	Аргумент который указывает на необходимость вывод описание использования библиотеки (--readme)

Результатом будет передана текстовая информация в системный вывод описание процесса использования библиотеки

Пример вызова:
java -jar template-extend-lib.jar path/to/templates --readme

4.3.6. Функционал вывода списка подключенных шаблонов

Краткое описание метода:

Позволяет в терминальном/командном/пакетном вызове отобразить список подключенных шаблонов sved формата.

Входные данные:

№	Тип	Обязательность	Описание
1	String	О	Аргумент который на расположение папки, содержащей шаблоны.

2	String	O	Аргумент который указывает на необходимость вывода списка информации о шаблонах (--list)

Исключения:

№	Тип	Описание
1	SvedLibraryPathIsEmpty	Генерация исключения если заданный параметр, который определяет место хранения файлов формата sved не задан/пуст или путь не существует в файловой системе.

Результатом будет передана текстовая информация в системный вывод о том какие шаблоны находятся по заданному пути.

Пример вызова:

java -jar template-extend-lib.jar path/to/templates -- list

4.4. Описание добавления вложенных файлов

При визуализации документа, xml файлы и файлы подписи по умолчанию включаются в результирующий PDF-документ как вложения.

За создание вложенных файлов отвечает метод convert интерфейса Converter. Сам интерфейс находится в пакете core-module. В этом же пакете находится интерфейс Extractor, в котором находится метод **removeAttachment**, предназначенный для удаления вложенного файла по имени.

Методы доступны только при подключении библиотеки в качестве зависимости.

4.4.1. Описание метода convert

Краткое описание метода:

Метод создаёт вложения, из списка документов.

Входные аргументы:

№	Тип	Обязательность	Описание
1	Path	O	Путь до pdf-файла
2	Map<String, InputStream> embed	O	Список пар имени файла и потока байт

Выходные аргументы:

№	Тип	Описание
1	Optional<Path>	Путь к обновлённому pdf-файла

Пример вызова на языке java:

```
Path resultPath = Paths.get("C:/output/result.pdf");
Map<String, InputStream> embed = Map.of(
        "example.xml", Files.newInputStream(Paths.get("C:/input/example.xml")),
        "document.sig", Files.newInputStream(Paths.get("C:/input/document.sig"))
    );
converter.convert(resultPath, embed);
```

4.4.2. Описание метода removeAttachment

Краткое описание метода:

Удаляет вложенный файлы из pdf-документа по имени.

Входные аргументы:

№	Тип	Обязательность	Описание
1	String	O	Путь до pdf-файла
2	String	O	Имя вложения, которое необходимо удалить

Выходные аргументы:

№	Тип	Описание
1	-	Метод не возвращает значения (void).

Пример вызова на языке java:

```
String pdfFilePath = "C:/output/result.pdf";

String attachmentName = "example.xml";

removeAttachment(pdfFilePath, attachmentName);
```

4.5.Описание общих объектов данных

4.5.1. Описание структуры объекта

Parameters

№	Имя	Тип	Описание
1	docTypeName	String	Тип документа
2	url	String	Интернет адрес расположения приказа
3	Description	String	Краткое описание
4	docId	String	Номер в БД Конструктора
5	Status	Integer	Статус шаблона
6	friendlyName	String	Имя данное автором шаблона при создании
7	Parameters	List<Parameter>	Параметры привязки шаблона к конкретному типу документа
8	Logs	List<InfoLog>	Информация по смене статусов
9	Format	String	Формат документа под который разрабатывался шаблон
10	Redaction	String	Текущая редакция шаблона
11	dateCreate	String	Дата создания шаблона
12	datePublish	String	Дата публикации шаблона

13	dateChange	String	Дата изменения шаблона
14	Author	String	Имя пользователя автора
15	Guid	String	Уникально строковое обозначение шаблона

Описание структуры объекта Parameter

№	Имя	Тип	Описание
1	Knd	String	КНД
2	Code	String	Код
3	codeName	String	Наименование кода
4	xsdId	String	Присвоенный идентификатор xsd
5	xsdName	String	Имя файла xsd схемы
6	Guid	String	Присвоенный guid

Описание структуры объекта InfoLog

№	Имя	Тип	Описание
1	Date	String	Дата смены статуса
2	Status	Integer	Статус
3	Info	String	Краткое описание
4	Log	String	Информация по статусу

Описание структуры перечисления StatusFileService

№	Имя	Значение	Описание
1	READY	0	Опубликован
2	DEVELOP	1	В разработке
3	CREATE	2	Новый
4	ARCHIVE	3	Отправлен в архив

5	DELETE	4	Удален
6	SEND	5	Отправлен на публикацию
7	ERROR	6	Содержит ошибки
8	SEND_UNREADY	7	Отправка на снятия с публикации
9	ERROR_UNREADY	8	Ошибка при снятии с публикации
10	UNREADY	9	Снят с публикации

5. Исключения

5.1. Генерируемые исключения статического метода main «Генерации pdf- файла»

№	Тип	Описание
1	BreakingJar	Генерация исключение при попытки проверки внутренней конфигурации библиотеки
2	EmptyArgs	Генерация исключения при проверки переданных параметров если количество параметров меньше двух
3	EmptyListFile	Генерация исключения при проверки наполнения директории / существования, шаблонов sved / xslx
4	SvedLibraryPathIsEmpty	Генерация исключения если заданный параметр который определяет место хранения файлов формата sved не задан/пуст или путь не существует на файловой системе (или входной шаблон xlsx)
5	SaveResultPdfException	Генерация исключения при возникновении ошибки сохранения результирующего файла на стороне библиотеки
6	ConvertLibraryException	Ошибка конвертации xml в pdf на стороне библиотеки
7	XmlValidationException	Ошибка валидации XML-файла по XSD-схеме

5.2. Генерируемые исключения метода convertS «Генерации потока pdf-файла»

№	Тип	Описание
1	BreakingJar	Генерация исключение при попытки проверки внутренней конфигурации библиотеки
2	EmptyArgs	Генерация исключения при проверки переданных параметров если количество параметров меньше двух

3	EmptyListFile	Генерация исключения при проверки наполнения директории / существования, шаблонов sved / xslx
4	SvedLibraryPathIsEmpty	Генерация исключения если заданный параметр который определяет место хранения файлов формата sved не задан/пуст или путь не существует на файловой системе (или входной шаблон xlsx)
5	SaveResultPdfException	Генерация исключения при возникновении ошибки сохранения результирующего файла на стороне библиотеки
6	ConvertLibraryException	Ошибка конвертации xml в pdf на стороне библиотеки
7	XmlValidationException	Ошибка валидации XML-файла по XSD-схеме

5.3.Генерируемые исключения метода list

№	Тип	Описание
1	SvedLibraryPathIsEmpty	Генерация исключения если заданный параметр который определяет место хранения файлов формата sved не задан/пуст или путь не существует на файловой системе (или входной шаблон xlsx)