# WIPO Sequence Validator

## 사용 매뉴얼

### 버전 **2.3.0**

*본 문서의 목적은 특허청이 WIPO Sequence Validator 웹 서비스를 배포하게 지원하고 Validator의 환경설정을 지원하는 것입니다.*

WIPO 공식 사용 전용

# 목차

# 1. 소개

WIPO Sequence Validator (이하 'Validator' 또는 '도구'로 칭함)의 주요 목적은 각 특허청(IP Offices, IPOs)에 WIPO ST.26 형식의 XML 파일을 검증하기 위한 서비스를 제공하여 XML 파일이 WIPO 표준 ST.26을 따르는지 확인하는 것입니다. WIPO Sequence 데스크탑 애플리케이션을 사용하여 작성된 서열 목록은 WIPO ST.26을 준수하지만, 사용자는 가장 적합하다고 생각하는 어떠한 도구도 사용할 수 있습니다.

본 문서의 목적은 다음 섹션에 자세히 설명한 도구의 구축, 배포, 설정 및 파일 사용자 시스템을 설명하는 것입니다. 문제 해결에 대한 질문은 다음 위키를 참조하십시오 https://www3.wipo.int/confluence/display/WSVAL/Troubleshooting

## 1.1. 검증기 워크플로우 개요

도구는 다음과 같은 네 가지 사용 사례를 제공한다:

- WIPO ST.26 파일의 검증;
- 실행 중인 검증의 상태 요청;
- 환경설정 파일 업데이트 (특허청 관리자만); 및
- 프로세스가 완료되면 검증 프로세스의 결과로 콜백 엔드포인트를 호출합니다.

*참고: 이 콜백 엔드포인트[1] 는 Valdator의 범위 밖에 있습니다. 엔드포인트를 설정하는 것은 이 서비스를 개발하고 설정하는 각 특허청의 책임입니다.*

도구는 웹 서비스로 실행할 수 있는 JAR 파일 또는 Tomcat 서버 상에서 배포할 수 있는 WAR 파일로 구성된다.

양자 모두의 경우에, WIPO ST.26 서열 목록을 검증하려고, 도구는 로컬 파일 시스템으로부터 파일을 사용하고 검증 결과에 관한 검증 보고서를 생성하며, 선택적으로, 콜백 엔드포인트를 호출하여 검증 처리의 결과, 즉, 검증 보고서를 반환한다.

검증기의 주요 워크플로우는 다음과 같다:

a) 각 특허청 IT 시스템에서 WIPO ST.26 XML 파일은 기본 "받은문서" 폴더 또는 원하는 지정 폴더에 저장된다.

[1] 여기서 콜백 엔드포인트는 요청 메시지를 보낼 수 있는 URI로 식별되는 고유 주소이다.

b) IPO 시스템은 파일 검증을 요청하는 HTTP Post를 시작합니다. 환경 설정에 따라, 특허청 시스템은 파일의 "전체" 또는 "형식" 검증을 요청할 수 있습니다. "형식" 검증 프로세스는 ST.26 파일이 XML 파일인지 여부를 확인하고 ST.26 DTD와 비교하여 ST.26 파일을 확인합니다. "전체" 검증 프로세스는 "형식" 검증 프로세스를 수행할 뿐만 아니라 ST.26의 컨텐츠에서 파생된 업무 검증 규칙과 대비하여 ST.26 파일을 검증합니다.

*참고: "형식" 검증 프로세스는 동시에 수행될 수 있으므로 온라인 제출 접수 시스템에만 사용할 것을 권장하는 반면 "전체" 검증은 훨씬 더 오래 걸리므로 배치 프로세스에 권장한다.*

c) 검증이 완료되면, 파일이 "형식" 검증을 통과했는지 여부와, 특허청 IT 시스템이 "전체" 검증을 선택한 경우, 추가로 업무 규칙 검증 프로세스가 올바로 시작되었는지 여부를 나타내는 응답이 제공됩니다.

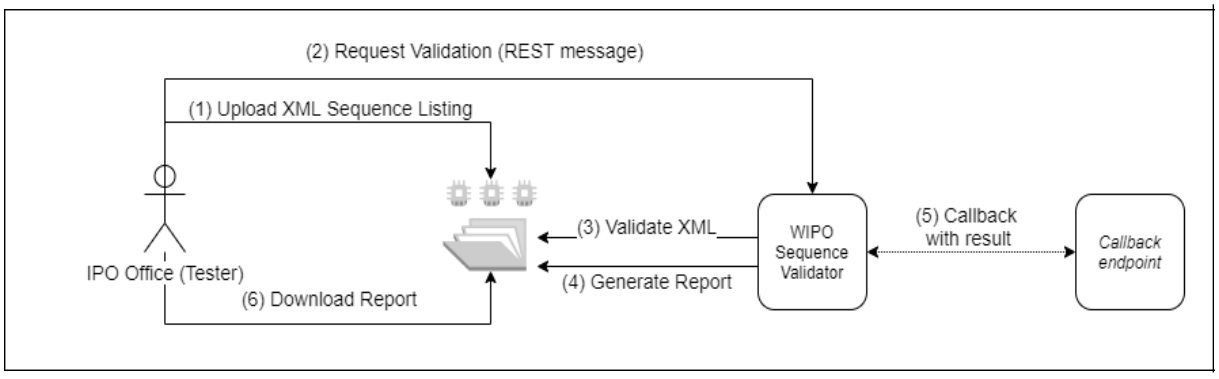
d) 검증기가 "전체" 검증을 수행 중인 경우, "받은문서" 폴더에서 XML 파일을 검색하여 업무 규칙 검증 처리를 시작한 후 다음을 수행한다:

e) 검증기는 지정된 "받은문서" 폴더에 XSD 스키마를 기반으로 XML 보고서 파일 ("report_<filename>.xml")을 생성하고 검증된 WIPO ST.26 XML 파일을 "보낼문서" 폴더로 이동합니다. 선택적으로 검증기는 동시에 이 검증 보고서의 HTML 버전도 생성합니다.

f) XML 보고서 파일이 생성되고 XSD 스키마를 준수하는지 확인하자마자, 검증기는 "보고서"라는 지정된 폴더 아래에 HTML 보고서 파일을 생성합니다.

g) 업무 규칙 유효성 검증 프로세스가 완료된 후, 콜백 엔드포인트가 구성된 경우 검증기에서 호출되고, 검증 프로세스와 관련된 추가 정보가 요청에 추가됩니다. 요청의 구조와 일부 샘플 데이터는 아래 섹션 4 에 제공됩니다.

h) 콜백 엔드포인트는 응답에서 빈 코드 또는 성공 코드를 반환해야 합니다 (오류 없음). [*참고: 이 단계는 외부 웹 서비스가 사용 가능하게 되고 호출이 검증기에서 구성된 경우에만 실행됩니다.*] 검증기와 콜백 엔드포인트 간의 연결도 필요합니다. 위에서 언급한 바와 같이, 외부 웹 서비스는 검증기의 일부를 구성하지 않으며 아래에 명시한 계약에 따라 각 특허청이 개발 및 구성해야 합니다.




i) 특허청 시스템은 "보고서" 폴더에서 검증 보고서를 검색할 수 있습니다.

*참고: WIPO 서열 검증기는 웹 API를 사용하여 지적 재산 데이터를 처리 및 통신하기 위한 WIPO 표준: WIPO ST.90을 준수합니다.*

## 1.2. 검증기 파일 시스템 구조의 정의

검증기가 사용하는 파일 시스템의 구조는 5개의 폴더로 구성된다:

- **"받은문서" 폴더**: 이 폴더는 검증을 위해 특허청이 WIPO ST.26 파일을 제공하는 로컬 폴더이다.
- **"처리" 폴더**: 이 폴더는 "받은문서"에 있는 파일을 처리 중 임시로 전달하는 로컬 폴더입니다. 이 폴더에는 2개의 하위폴더가 있습니다:
    - **"전체 검증" 폴더**: 전체 검증 대기 중인 파일 저장.
    - **"형식 검증" 폴더**: 형식 검증 대기 중인 파일 저장.
- **"보낸문서" 폴더**: 검증이 완료되면, 응용 프로그램은 이 로컬 폴더에 WIPO ST.26 파일의 소스를 저장한다
- **"보고서" 폴더**: 이 폴더는 검증 결과를 검증 보고서 파일에 저장하는 로컬 폴더입니다.
- **"매개변수" 폴더**: 이 폴더는 비동기식 심층 검증 처리에 대한 매개변수를 제공하기 위해 검증 요청의 모든 검증 매개변수를 갖는 JSON (.json) 파일이 있는 로컬 폴더이다.

예시 파일 시스템 구조는 아래에 제공된다:

```
/temp/ST26
/temp/ST26/inbox
/temp/ST26/process/full
/temp/ST26/process/formality
/temp/ST26/outbox
/temp/ST26/reports
/temp/ST26/params
```

*[중요: 기본적으로 /temp/ST26 디렉토리는 도구가 있는 상위 디렉토리에 있어야 합니다. 예를 들어, JAR 또는 WAR 파일이 C:/dev에 있는 경우 폴더 구조는 C:/temp/ST26/…으로 생성되어야 합니다.]*

## 2. WIPO 서열 검증기의 배포

앞서 살펴본 바와 같이, 검증기는 아래에 나열된 2개의 바이너리 형식 중 하나로 제공된다. 특허청이 검증기를 배포하려는 인프라 유형에 따라, 특허청은 다른 유형보다 한 유형의 바이너리를 선택하는 것을 선호한다.

검증기가 제공되는 2개의 바이너리는 다음과 같다:

- **바이너리 SpringBoot JAR**: 이 바이너리는 실행 가능한 JAR 파일입니다. 이를 위해서는 Java 8이 설치되어 있어야 합니다.
- **War 패키지 바이너리:**이 바이너리는 Servlet 컨테이너 상에 배포하기 위한 것입니다. Tomcat 8.5와 같은, Spring Boot 2 및 Servlet Spec 3.1+와 호환되는 응용 프로그램 서버가 필요합니다.

다음 섹션은 Java 응용 프로그램 서버 내에서 Spring Boot app 또는 WAR로서 검증기를 배포하는 것을 자세히 설명합니다.

### 2.1. Spring Boot JAR로 검증기 시작하기

Spring Boot JAR는 내장 서버를 포함하고 있어, 별도의 서버가 없어도 검증기 API를 배포할 수 있다. 이는 IT 인프라 차원에서 구성 및 배포를 크게 단순화한다.
내장 서버를 실행하려면, 다음 명령을 실행해야 한다.

*참고: Java 8이 서버에 이미 설치되어 있어야 한다: Java가 UTF-8의 사용을 보장하지 않으므로 시스템 속성 "file.encoding" 이 "UTF-8"로 설정되어야 한다. 이 작업은 다음을 포함하여 수행될 수 있다:*

```
java -D -jar wipo-sequence-validator.jar
```

검증기 API는 Swagger UI를 통해 액세스할 수 있다:
http://[host-name]:8080/swagger-ui.html

검증기 API는 다음 엔드포인트에서 액세스할 수 있다:
http://[host-name]:8080/api/[version]/status
http://[host-name]:8080/api/[version]/validate

여기서, 특허청은 다음과 같이 변경해야 한다:

- [host-name]은 서버 호스트 이름으로 대체해야 한다; 그리고
- [version]은 검증기 API의 버전 (예를 들어, v1.0)으로 대체되어야 한다.

기본적으로 서버는 포트 8080에서 실행되고, 포트를 변경하려면 다음과 같이 "--server.port" 명령행 옵션을 추가해야 합니다:

```
java -D"file.encoding=UTF-8" -jar wipo-sequence-validator.jar –-server.port=<port-number>
```

기본적으로 검증기는 자바 가상 머신(JVM) 기본 메모리 설정을 사용한다. 기본 최대 힙 크기는 사용 가능한 실제 메모리의 4분의 1이다.

최대 힙 크기를 수정하려면, 명령행을 사용하여 실행할 때 "-Xmx" 옵션을 사용해야 한다[2]:

```
java -D"file.encoding=UTF-8" -Xmx[size]-jar wipo-sequence-validator.jar
```

#### 2.1.1. 검증기를 실행 가능한 응용 프로그램으로 배포하기

예를 들어, 검증기는 또한 운영 체제의 시작과 함께 실행을 지원하기 위해 운영 체제에 의해 관리되는 서비스로 설치될 수 있다.

WIPO 서열에 지원되는 모든 플랫폼에 대해 이러한 방식으로 Spring Boot JAR 파일을 구성할 수 있다: Windows, Linux 및 Mac OS.

다음 가이드는 각 운영 체제에 대해 JAR 파일을 실행하는 시스템 서비스를 생성하는 방법에 대한 세부사항을 제공한다. 또한 서비스의 다른 옵션을 구성하는 방법 및 응용 프로그램 실행 방법에 대한 정보를 제공한다:

https://docs.spring.io/spring-boot/docs/current/reference/html/deployment-install.html

[2] https://docs.oracle.com/javase/8/docs/technotes/tools/windows/java.html#BABHDAB

## 2.2. WAR 웹 서비스로 배포

제공된 두 번째 바이너리 유형의 경우, WAR 패키지는 Apache Tomcat 8.5와 같은 기존 Java 응용 프로그램 서버에서 배포될 수 있다.

*참고: Servlet 3.1과 호환 가능한 컨테이너가 필요합니다.*

다음 지시사항은 Tomcat 응용 프로그램 서버를 위한 것이다. 여기서, "$TOMCAT_ROOT"는 Tomcat 서버의 루트 폴더를 말하며, 이 값은 파일 경로에 대한 관련 값으로 대체되어야 한다.

a. 서버 중지: $TOMCAT_ROOT\bin\catalina.bat stop
b. war를 $TOMCAT_ROOT\webapps\wipo-sequence-validator.war에 복사
c. 서버 시작: $TOMCAT_ROOT\bin\catalina.bat start

*참고: Java가 UTF-8 사용을 보장하지 않으므로, 시스템 속성 "file.encoding"은 응용 프로그램 서버의 시작시 "UTF-8"로 설정되어야 합니다. 이 작업은 다음을 포함하여 수행될 수 있습니다:*

*-D"file.encoding=UTF-8"*

검증기 API는 위에 지시된 대로 Swagger UI를 통해 액세스될 수 있다:
http://host-name:8080/wipo-sequence-validator/swagger-ui.html

검증기 API는 다음 엔드포인트에서 액세스할 수 있다:
http://[host-name]:8080/wipo-sequence-validator/api/[version]/status
http://[host-name]:8080/wipo-sequence-validator/api/[version]/validate

여기서, 특허청은 다음과 같이 변경해야 한다:

- [host-name]은 서버 호스트 이름으로 대체해야 한다, 그리고
- [version] 은 API의 버전 (예를 들어, v1.0) 으로 대체되어야 한다.

기본적으로, 서버는 포트 8080에서 실행된다. 이를 다른 포트로 변경하려면 Tomcat 구성 파일을 여기에 제공된 지시사항에 따라 수정해야 한다:

https://tomcat.apache.org/tomcat-8.5-doc/config/http.html#Common_Attributes

기본적으로, 검증기는 JVM 기본 메모리 설정을 사용한다. 기본 최대 힙 크기는 사용 가능한 실제 메모리의 4분의 1이다.

최대 힙 크기를 수정하려면, 위의 섹션 2.1에서 나타낸 대로, 명령행을 사용하여 실행할 때 "-Xmx" 옵션을 사용해야 한다.

## 3. 검증 보고서

검증 보고서를 생성할 수 있는 형식은 두 가지가 있습니다: XML과 HTML (또는 둘 다).
도구에 의해 생성된 검증 보고서는 XML 형식이며 사용된 템플릿은 아래에 제공되어 있다:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<VerificationReport applicationNumberText = "123" productionDate ="YYYY-MM-DD"
filingDate = "YYYY-MM-DD" softwareBuildVersion ="2.3.0.-SNAPSHOT" softwareVersion
="2.3.0" sourceFileName="[ST.26 filename]" xsi:noNamespaceSchemaLocation ="st26-
seql-verification-report-2.0.0.xsd" xmlns:xsi= ="http://www.w3.org/2001/XMLSchema-
instance" >
  <VerificationMessageBag
             >
   <VerificationMessage>
      <Severity>[ERROR | WARN | XML_WARN |  XML_ERROR]</Severity>
      <DataElement>[ST.26 element]</DataElement>
      <DetectedSequence>[Sequence ID]</DetectedSequence>
      <DetectedValue>[value]</DetectedValue>
      <MessageKey>[Message key]</MessageKey>
      <ParameterBag>
            <Parameter key="param key">Param value</Parameter>
      <ParameterBag>
      <LocalizedMessage> [Localized message] </LocalizedMessage>
    </VerificationMessage>

...
```

루트 레벨에 표시된 특성은 다음과 같습니다:

- 'applicationNumberText': 이 서열 목록과 연관된 애플리케이션;
- 'productionDate': 검증이 수행된 날짜;
- 'filingDate': 출원일;
- 'softwareBuildVersion': 검증에 사용되는 검증기의 버전;
- 'softwareVersion': 서열 목록을 생성하기 위해 사용되는 WIPO 서열의 버전; 및
- 'sourceFileName': XML 인스턴스를 나열하는 서열의 이름.

이 검증 보고서의 예는 부속서 III에 제공된 이러한 구성 요소에 대한 허용 값과 함께 이 매뉴얼의 부속서 I로 제공됩니다.

표시된 심각도와 관련하여, 다음 분류에 유의하십시오.

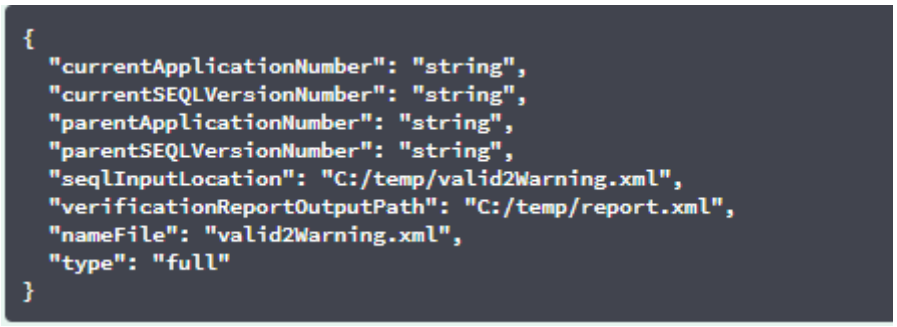
- ERROR - '전체' 확인 중에 반환된 오류

- WARNING - '전체' 확인 중에 반환된 경고
- XML_ERROR - '형식' 확인 중에 반환된 오류
- XML_WARN - '형식' 확인 중에 반환된 경고

검증 보고서는 WIPO Sequence에서 사용되는 동일한 서식 시트를 사용하여 HTML 형식으로도 생성됩니다. HTML 검증 보고서는 본 매뉴얼의 부속서 IV로 예시됩니다.

## 4. 콜백 엔드포인트 요청

검증기에 대한 콜백 엔드포인트에 의해 이루어진 요청은 파일 위치 및 검증 처리를 자세히 설명하는




다음 매개변수를 포함해야 한다:
검증 요청의 "seqlInputLocation" 필드는 검증할 XML 서열 목록의 경로를 표시하도록 설정해야 합니다. 특허청에서 이 필드를 비워두면, 도구는 기본 "받은문서" 폴더에 있는 파일명 "nameFile"을 갖는 XML 파일을 검증하려고 시도합니다. "nameFile" 매개변수는 검증할 서열 목록 파일을 확인합니다.

요청 내의 "verificationReportOutputPath" 는 도구에서 생성된 application.property 파일의 설정에 따라 검증 보고서(.xml 및/또는 .html)의 파일 위치를 제공합니다. 사용자가 필드를 비워두거나 유효하지 않은 파일 경로를 입력하면, 검증 보고서가 기본 "보고서" 폴더에 저장됩니다.

### 4.1. 콜백 엔드포인트 요청 형식

속성 "api.URL"이 구성되는 경우, 검증기는 검증의 결과를 지정된 URL을 사용하여 엔드포인트로 보내려고 시도합니다.
검증기와 통신하려면, 콜백 엔드포인트가 다음 웹 서비스 계약(YAML)을 준수해야 합니다.
https://www.wipo.int/standards/en/sequence/callback_end_point_web_service_contract.yml

또한, 요청은 이 구조를 갖는 JSON 오브젝트여야 한다.

```
{
 "currentApplicationNumber":
 "string",
 "currentSEQLVersionNumber":
 "string", "elapsedTime": 0,
 "endTime":
 "string",
 "errorSummary":
 [
  {
   "dataElement":"string",
   "detectedSequence": "string", "index":
   0,
   "key": "string",
   "locmessage": "string",
   "params":
   { "additionalProp1":
   "string",
   "additionalProp2":
   "string",
   "additionalProp3":
   "string"
   },
   "paramsForXML": [
    {
     "key": "string",
     "value": "string"
    }
   ],
   "reportValue": "string",
   "sequenceIDNumber":
   "string",
   "type": "string"
  }
 ],
 "httpStatus": "string",
 "parentApplicationNumber":
 "string",
 "parentSEQLVersionNumber":
 "string", "processID": "string",
 "seqIDQuantity": 0,
 "seqInputQuantity": 0, "seqlType": "string", "startTime": "string",
 "totalErrorQuantity": 0,
 "totalWarningQuantity": 0,
 "verificationReportOutputPath":
 "string"
}
```

이는 검증기를 호출한 외부 엔드포인트로 보낼 예시 JSON 오브젝트이다.


```
{
  "processID": "1608194222169dvVE",
  "seqlType": "ST.26",
  "httpStatus": "SUCCESS",
  "currentApplicationNumber": "string",
  "currentSEQLVersionNumber": "string",
  "parentApplicationNumber": "string",
  "parentSEQLVersionNumber": "string",
  "verificationReportOutputPath":
  "C:/temp/report.xml", "startTime": "2020-12-
  17 09:36:54.000000",
  "endTime": "2020-12-17 09:37:26.000607",
  "elapsedTime": 32607,
  "totalWarningQuantity": 1,
  "totalErrorQuantity": 2,
  "seqInputQuantity": 3,
  "seqIDQuantity":
  3,
  "errorSummary":
  [
    {
       "index": 0, "reportValue": "",
       "type": "WARNING",
       "params":com.wipo.st26.ipotool.models.ServiceRequest@58
       87858, "key": "X_EARLIEST_PRIO_APPLICATION_ID_MISSING",
       "locmessage": "Earliest priority application information is absent. It must be
 provided when a priority claim is made to an earlier application.",
       "detectedSequence": "",
       "dataElement": "PROPERTY_NAMES.EARLIEST_PRIORITY_APPLICATION"
     },
```

```
            "reportValue": "",
            "type": "WARNING",
            "params":com.wipo.st26.ipotool.models.ServiceRequest@5887858,
            "key": "X_EARLIEST_PRIO_APPLICATION_ID_MISSING",
            "locmessage": "Earliest priority application information is absent. It must be provided when a priority claim is made to an earlier application.",
            "detectedSequence": "",
            "dataElement": "PROPERTY_NAMES.EARLIEST_PRIORITY_APPLICATION"
        },
        {
            "index": 0,
            "reportValue": "-",
            "type": "ERROR",
            "params": {},
            "key": "INVENTION_TITLE_MISSING",
            "locmessage": "The invention title is missing. At least one invention title must be entered.",
            "detectedSequence": "",
            "dataElement": "PROPERTY_NAMES.INVENTION_TITLE_BAG"
        },
        {
            "index": 1,
            "reportValue": "-",
            "type": "ERROR",
            "params": {},
            "key": "INVENTION_TITLE_MISSING",
            "locmessage": "The invention title is missing. At least one invention title must be entered.",
            "detectedSequence": "",
            "dataElement": "PROPERTY_NAMES.INVENTION_TITLE_BAG"
        }
    ]
}
```

## 4.2. 검증 보고서

섹션 3에서 언급한 대로, 검증 후 생성된 검증 보고서는 verificationReportOutputPath에 있으며, 이는 기본적으로 XML 보고서 파일의 경우 "/temp/st26/reports/[verificationID]/report.xml" 이고, HTML 보고서 파일의 경우 "temp/st26/reports/[verificationID]/report.html"입니다.

application.properties 파일에서, HTML 보고서 생성을 활성화하거나 비활성화할 수 있습니다. 보고서 생성을 활성화하려면 값이 "true"이고 비활성화하려면 값이 "false"입니다.

```
#turning on/of printing HTML Report
app.property.html=true
```

이 보고서의 내용은 ServiceRequest의 errorSummary 필드에 있는 콜백 엔드포인트로 전송됩니다. 이 필드의 예는 위의 섹션 4에 제공된 요청 예에서 제공됩니다.

# 5. 환경 설정

## 5.1. 기본 설정

검증기는 속성 파일을 사용하여 구성됩니다. 기본 application.properties 파일은 다음 값을 가집니다[3]:

```
########### WIPO Sequence Validator properties

## -- FOLDERS --
#Base path to be used by the rest of folders
app.basePath=/temp/st26/
#Folder to put the files to be processed\
app.inboxPath=${app.basePath}inbox/

#Folder to store the ST26 files once validated
app.outboxPath=${app.basePath}outbox/
#Folder to store the validation reports
app.reportsPath=${app.basePath}reports/

#Folder to store the parameters
app.paramsPath=${app.basePath}params/

#Parent folder for full and formality folders
app.processPath=${app.basePath}process/

#Files in process for a full validation are stored in this folder
app.process.fullPath=${app.processPath}full/
#Files in process for a formality validation are stored in this folder
app.process.formalityPath=${app.processPath}formality/

alternativeResourceBasePath=${app.basePath}alt_resources

## --PREFERENCES --
#To enable the rule VXQV49 set this value to true, default value is false.
app.preferences.optionalEnglishQualifierValue=false
# Please enter either: ERROR or WARNING to specify the type of the verification message for the rule
VXQV_49, default value is "WARNING".
 app.preferences.optionalRuleType=WARNING

#locale used for the localized messages from the verification report validator_locale=en
#Setting the location of the endpoint
api.URL=
#turning on/of printing HTML Report
app.property.html=true
#software version from pom.xml
app.version=@project.version@
```

[3] 2023년 2월에 유효

```
## -- WATCHER --

# These properties control the process looking for files in the folders to be processed
# (see: https://docs.spring.io/spring-framework/docs/current/javadoc-
api/org/springframework/scheduling/concurrent/ThreadPoolTaskExecutor.html)

processing.delay=10000
processing.corePoolSize=5
#Max number of files being validated concurrently
processing.maxPoolSize=10
processing.queueCapacity=1000 processing.enabled=true

##-- LOGGING (see https://logback.qos.ch/manual/configuration.html)

logging.level.root=info
logging.level.org.wipo=info
logging.pattern.console=%d{yyyy-MM-dd HH:mm:ss} [%thread] %-
5level %logger{36} - %msg%n

# HEALTH ENDPOINT
management.endpoints.jmx.exposure.include=health #
Show details of health endpoint
management.endpoint.health.show-details=always
```

여기에 제공된 매개변수의 값을 수정하려면 대체 "application.properties" 파일을 사용해야 합니다. Spring Boot 문서에 자세히 설명된 몇 가지 옵션이 있습니다: https://docs.spring.io/spring-boot/docs/2.0.6.RELEASE/reference/html/boot-features-external- config.html#boot-features-external-config-application-property-files

가장 간단한 옵션은 새로운 application.properties 파일을 제공하는 것이며, 이 파일은 작업 순서에 따라 다음 위치에서 검색됩니다:

a) 현재 디렉토리 내의 "/config" 폴더*[참고: 검증기가 Tomcat에서 WAR 파일로 배포된 경우, 이 폴더는 "lib" 폴더, 예를 들어, "/opt/apache-tomcat/lib/config"에 있습니다]*;

b) 현재 디렉토리 *[참고: 검증기가 Tomcat에서 WAR 파일로 배포된 경우, 이 폴더는 "lib" 폴더, 예를 들어, "/opt/apache-tomcat/lib/"에 있습니다]*;

c) $classpath 또는 config 패키지;

d) $classpath 루트.

또한, 도구를 시작할 때 명령행에서 매개변수를 확립하여 구성 파일의 경로와 이름을 지정할 수 있다.

JAR 배포를 위해:

- java -jar -Dspring.config.location= <PATH_TO_FILE> wipo-sequence-validator.jar

Tomcat의 WAR 배치의 경우, CATALINA_OPTS 에 다음 항목을 추가해야 합니다:

- export CATALINA_OPTS="-Dspring.config.location=

WAR 배포를 사용할 때, 새로운 application.properties 파일을 웹 애플리케이션의 "WEB-INF/classes" 폴더에 또한 복사하거나 기존 파일을 편집할 수 있습니다.

### 5.2. Configuring verification rule VXQV_49

```
app.preferences.optionalEnglishQualifierValue=false
app.preferences.optionalRuleType=WARNING
```

application.properties 파일의 optionalEnglishQualifierValue 값은 사용자가 VXQV_49 규칙을 활성화하려는 경우 'true'로 설정할 수 있으며, optionalRuleType 값을 'ERROR' 또는 'WARNING'으로 업데이트하여 규칙의 심각도를 설정할 수 있습니다. 이 두 속성의 기본값은 위에 나와 있습니다.

### 5.3. 엔드포인트 상태 확인

```
# HEALTH ENDPOINT
management.endpoints.jmx.exposure.include=health
```

Validator 서비스는 애플리케이션의 'health'에 대한 기본 정보를 제공하는 /health 엔드포인트를 구현하였습니다.

/health 엔드포인트를 탐색하기 위한 URL은 http://localhost:8080/wipo-sequence-validator 입니다. /actuator/health URL. The endpoint should display the following:

- 애플리케이션이 정상인 한 상태는 'UP'입니다.
- 데이터베이스 연결이나 디스크 공간 부족 등의 문제로 인해 애플리케이션이 비정상 상태가 되면 'DOWN'이 표시됩니다.

/health 엔드포인트는 단순한 'UP' 또는 'DOWN' 상태만 표시합니다. application.properties 파일의 다음 속성은 상태 점검 프로세스의 일부로 확인된 모든 상태 표시기의 상태를 포함하여 전체 세부 정보를

```
# 상태 엔드포인트의 세부 정보 표시
management.endpoint.health.show-
```

제공합니다.

/health 엔드포인트에는 상태 점검 프로세스의 일부로 실행되는 DiskSpaceHealthIndicator의 세부 정보가 포함됩니다.

/health 엔드포인트는 이와 같이 일련의 키-값 쌍으로 표시되며 추가 세부 정보를 포함합니다. 아래에 예가 나와 있습니다.

{"status":"UP","details":{"diskSpace":{"status":"UP","details":{"total":511123124224,"free":373225091072,"threshold":10485760}}}}

## 5.4. 현지화 메시지

Validator는 공식적인 10개의 PCT 언어(아랍어, 중국어, 영어, 프랑스어, 독일어, 포르투갈어, 일본어, 한국어, 러시아어 및 스페인어) 각각의 현지화 메시지를, 예를 들어, 검증 보고서에서 제공할 수 있습니다.

기본적으로 이러한 메시지는 영어로 제공됩니다. 이러한 메시지를 다른 언어로 제공하도록 검증기를 구성하려면, application.properties 파일의 validator_locale 매개변수를 적절한 언어 코드로 설정해야

```
#Local used for the localized messages from the verification report
validator_locale=en
```

합니다.

*참고: 새 "application.properties" 파일에 설정된 속성을 적용하려면 검증기를 다시 시작해야 한다.*

## 5.5. 사용자 정의 생물명

사전 정의된 유기체 이름의 원래 목록의 일부를 형성하지 않는 고유한 사용자 지정 유기체 이름을 특허청이 제공하기 위해 "custom_organism.json"이라는 새 파일을 생성하여 사용자 지정 유기체

```
[
      {"value":"Custom Organism Sample"},
      {"value":"Custom Organism Sample 2"}
]
```

목록을 "alternativeResourceBasePath" 폴더에 제공할 수 있습니다. 이 파일의 구조는 다음과 같아야 합니다:

*참고: 생물명의 사전 정의된 목록과 달리, 모든 유기체가 알파벳의 각 문자에 대해 JSON 파일로 분리되지 않고 단일 JSON 파일에 포함되어 있다.*

## 5.6. ST.26 DTD 파일 참조하기

기본적으로 검증기는 최신 버전의 ST.26 DTD를 참조합니다. WIPO 서열 검증기의 현재 버전은 WIPO ST.26 DTD[3] 버전 1.3을 기반으로 합니다.

최신 버전의 ST.26 DTD 사본은 소스 코드의 “/src/main/resources” 폴더에 있는 검증기 라이브러리에

```
<?xml version="1.0" encoding="UTF-8"?>
<catalog xmlns="urn:oasis:names:tc:entity:xmlns:xml:catalog">
  <public publicId="-//WIPO//DTD Sequence Listing 1.3//EN"
  uri="ST26SequenceListing_V1_3.dtd"/>
</catalog>
```

포함된다 (이는 JAR 또는 WAR 파일 내에서 참조되는 정의된 파일 경로이다). 이는 "catalog.xml" 파일에서 참조된다. 아래와 같은 동일한 폴더의 파일:

새로운 DTD를 포함하는 방법에 대한 지침은 아래에 자세히 설명되어 있습니다. 검증 동안, XML 파일의 DOCTYPE 신고서에 설정된 DTD 버전이 사용됩니다. 먼저 publicId 는 사용할 DTD 파일의 위치를 확인합니다. publicId 가 카탈로그에 포함되어 있지 않은 경우, 시스템은 Java 프로세스가 실행 중인 루트 폴더에서 DTD 파일을 찾으려고 시도합니다.

[3] 2023년 2월부터 유효

### 5.6.1. 검증을 위한 대체 DTD 버전을 표시하는 방법

ST.26 DTD의 이전 버전을 참조하는 WIPO ST.26 파일을 검증할 수 있으려면, 이 ST.26 DTD 파일이 적절한 검증을 허용하기 위해 검증기에 사용할 수 있어야 한다.
이를 위해, 두 가지 대체 방법이 있다:

- JAR 파일의 압축을 풀고 "src/main/resources" 폴더에 추가 또는 대체 ST.26 DTD 파일에 대한 참조를 포함시킵니다.
- catalog.xml 파일을 수정하고 추가 ST.26 DTD에 대한 새 항목을 추가하거나 기존 항목을 편집합니다.

예를 들어:

```
<?xml version="1.0" encoding="UTF-8"?>
<catalog xmlns="urn:oasis:names:tc:entity:xmlns:xml:catalog">
<public publicId="-//WIPO//DTD Sequence Listing 1.2//EN"
uri="ST26SequenceListing_V1_2.dtd"/>
<public publicId="-//WIPO//DTD Sequence Listing 1.3//EN"
uri="ST26SequenceListing_V1_3.dtd"/>
</catalog>
```

JAR 파일을 수정하는 대신, 다음 단계를 따라야 한다:

a. catalog.xml 및 모든 DTD를 로컬 폴더에 복사합니다;

b. 추가 ST.26 DTD에 대한 참조를 포함하도록 catalog.xml 을 수정합니다; 그리고

c. 시작 시, 이 Java 시스템 속성 설정: xml.catalog.files=<path_to_catalog.xml>

*참고: Windows용 Tomcat에서 이 환경 변수를 추가하여 수행할 수 있다:*

```
set "JAVA_OPTS=%JAVA_OPTS%
 -Dxml.catalog.files=C:\\temp\\tomcat\\sharedclasspath\\catalog.xml"
```

*[중요: ST.26 DTD의 다른 버전을 포함하면 참조된 ST.26 DTD에 대한 XML 파일의 '형식' 검증이 가능하지만 '전체' 검증은 검증 규칙을 구현하기 위해 소스 코드를 변경해야 할 수 있다. 따라서 '형식' 검증을 수행할 때에만 다중 DTD를 사용하는 것이 권장된다.]*

# 6. 검증기 REST API

이 섹션에서, 검증기 API의 사용 사례가 명시된다: 3개의 서비스 또는 사용 사례가 있다:

- "받은문서" 폴더에 있는 파일을 검증하십시오; (섹션 6.1)
- 요청의 일부로 파일을 검증 (섹션 6.1); 및
- 검증 상태를 요청합니다 (섹션 6.2).

(OAS 3.0 [YAML 파일]로 작성된) 이 서비스에 대한 API 사양은 전체적으로 부속서 II로 제공됩니다.

## 6.1. WIPO ST.26 파일의 검증

| | |
|---|---|
| 요청 맵핑 | /api/v1/validate |
| 방법 | POST |
| 사용 | 응용 프로그램/json |
| 생산 | 응용 프로그램/json |
| 작업 | "받은문서" 폴더에 있는 기존 WIPO ST.26 파일의 검증을 요청한다.<br>검증 요청의 상태를 검색하기 위한 고유 "verificationID"를 반환한다. |
| 요청 | {<br>"currentApplicationNumber": "string",<br>"currentSEQLVersionNumber": "string",<br>"parentApplicationNumber": "string",<br>"parentSEQLVersionNumber": "string", "seqlInputLocation":<br>"string"(Location of Input.xml file),<br>"verificationReportOutputPath": "string" (report.xml의 목적지),<br>"nameFile": "file.xml",<br>"type": "string" (Possible values: full \| formality),<br>} |
| 응답 | • '202': "허용됨". WIPO ST.26 파일이 형식 검증을 통과하여 사업 규칙 검증이 시작되었습니다. 이 코드는 또한 검증 보고서를 검색하기 위한 고유 코드를 나타내는 응답 메시지를 포함합니다 ("verificationID"). WIPO ST.26 파일은 처리를 위해 "처리" 폴더로 이동합니다.<br>• '400': "요청 실패". REST 요청이 제대로 형성되지 않았거나 |

| | |
|---|---|
| | WIPO ST.26 파일이 XML 검증을 통과하지 못했습니다. 이 코드는 |

| | |
|---|---|
| | 오류의 세부사항을 표시하는 응답 메시지로 보완됩니다.<br>• '400': "요청 실패" - "FILENAME_NOT_VALID". 파일 명칭에 사용할 수 없는 문자가 포함되어 있습니다. 파일 명칭에 로마자가 아닌 문자를 사용할 수는 있지만 특정 문장 부호는 적합하지 않은 것으로 간주됩니다.<br>• '404': "찾을 수 없음". " 받은문서" 폴더에서 WIPO ST.26 파일을 찾을 수 없다<br>• '500': "인터넷 서버 오류". 내부 오류가 발생하였다. 이 코드는 오류의 세부사항을 표시하는 응답 메시지로 보완된다 |
| 사전 조건 | WIPO ST.26 파일은 사용자가 지정한 "받은문서" 폴더에 제공되어야 합니다. |
| 사후 조건 | WIPO ST.26 파일은 validationReportOutputPath에 제공된 위치 또는 다음 위치의 "보낼문서" 폴더로 이동됩니다.<br>"/[outbox]/[verificationID]/[file.xml]" 검증 보고서는<br>다음 위치에 생성됩니다:<br>"/[reports]/ [verificationID]/report_[file.xml or file.html]" |

*참고: 검증된 파일 이름에는 라틴 문자가 아닌 문자가 포함될 수 있지만 지정된 문자는 허용되지 않습니다.*

## 6.2. 검증 상태 요청하기

| | |
|---|---|
| 요청 맵핑 | /api/v1/status |
| 방법 | POST |
| 사용 | 응용 프로그램/json |
| 생산 | 응용 프로그램/json |
| 작업 | 특정 WIPO ST.26 파일에 대한 검증 상태 요청 |
| 요청 | {<br>verificationID: {type: string}<br>} |
| 응답 | □ '200': 성공. 이 코드는 또한 다음에서 선택된 검증 처리의 상태를 갖는 고유 ID를 나타내는 응답 메시지를 포함한다:<br>○ “RUNNING”: 파일이 처리 중이다<br>○ “FINISHED-VALID”: 파일이 형식 검증을 성공적으로 통과했으며, 검증 결과가 보고서 폴더에서 사용 가능하다.<br>○ “FINISHED-INVALID”: 처리가 완료되었지만 파일이 형식 검증을 통과하지 못했다. 검증 결과가 보고서 폴더에서 사용 가능하다<br>○ “NOT_FOUND”: "verificationID"를 찾지 못했다<br>○ “VERIFICATION_ID_ERROR”: "verificationID"가 요청에 포함되지 않았다<br>□ ‘400’: 요청 실패. REST 요청이 잘 형성되지 않았다.<br>□ '500': 서버 오류. 내부 오류가 발생하였다. 이 코드는 또한 오류의 세부사항을 표시하는 응답 메시지를 포함한다 |
| 사후 조건 | 검증기가 검증 상태를 제공한다. |
| 추정 | - |

[부속서 I 첨부]

# 부속서 I: 예시적인 XML 검증 보고서

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<VerificationReport productionDate="2020-12-17" sourceFileName="valid2Warning.xml">
    <VerificationMessageBag>
        <VerificationMessage>
            <Severity>WARNING</Severity>
            <DataElement>PROPERTY_NAMES.EARLIEST_PRIORITY_APPLICATION</DataElement>
            <DetectedSequence/>
            <DetectedValue/>
            <MessageKey>X_EARLIEST_PRIO_APPLICATION_ID_MISSING</MessageKey>
            <ParameterBag/>
            <LocalizedMessage>Earliest priority application information is absent.It must be provided when a priority claim is made to an earlier application.
            </LocalizedMessage>
        </VerificationMessage>
        <VerificationMessage>
            <Severity>ERROR</Severity>
            <DataElement>PROPERTY_NAMES.INVENTION_TITLE_BAG</DataElement>
            <DetectedSequence/>
            <DetectedValue>-</DetectedValue>
            <MessageKey>INVENTION_TITLE_MISSING</MessageKey>
            <ParameterBag/>
            <LocalizedMessage>The invention title is missing. At least one invention title must be entered.</LocalizedMessage>
        </VerificationMessage>
        <VerificationMessage>
            <Severity>ERROR</Severity>
            <DataElement>PROPERTY_NAMES.INVENTION_TITLE_BAG</DataElement>
            <DetectedSequence/>
            <DetectedValue>-</DetectedValue>
            <MessageKey>INVENTION_TITLE_MISSING</MessageKey>
            <ParameterBag/>
            <LocalizedMessage>The invention title is missing. At least one invention title must be entered.</LocalizedMessage>
        </VerificationMessage>
    </VerificationMessageBag>
</VerificationReport>
```

이 XML 사례는 다음에서 다운로드할 수 있습니다:
https://www.wipo.int/standards/en/sequence/example_verification_report.xml

[부속서 II 첨부]

## 부속서 II: 전체 API 사양 (YAML)

완전한 API 사양은 다음에서 다운로드할 수 있습니다:
https://www.wipo.int/standards/en/sequence/complete_validator_api_specification.yml

[부속서 III 첨부]

## 부속서 III: 속성 이름 (JSON)

도구 내에서 사용되는 속성 이름은 다음에서 다운로드할 수 있습니다:
https://www.wipo.int/standards/en/sequence/propertynames.json

*이것은 소스 코드의 일부이며 구현과 함께 필요에 따라 업데이트되는 점을 참고 바랍니다.*

[부속서 IV는 다음과 같습니다]

## 부속서 IV: 예시적인 HTML 검증 보고서

### Verification report

#### Verification Report Information

| | |
|---|---|
| **Production Date** | 2023-01-25 |
| **Application Number** | 123456 |
| **Sequence listing dated** | 2021-02-13 |
| **Software Version** | 2.2.0-SNAPSHOT |
| **Validator Version** | 0.1 |

#### Verification Messages

| Severity | Data Element | Message Text | Detected Value | Detected Sequence |
|---|---|---|---|---|
| ERROR | Feature Location | The feature location includes a residue number greater than the length of the sequence, which is invalid. | 14219 | 1 |
| WARNING | Project | A non English free text language code has been entered, but there are no qualifiers with a non-English free text value. | en | |

[문서의 끝]