



WIPO Sequence Validator – 操作手册

2.3.0 版

本文件的目的是支持知识产权局部署 WIPO Sequence Validator 的 web 服务，
并对 Validator 的配置提供支持。

目录

1. 简介	3
1.1. Validator 工作流程概述	3
1.2. 关于 Validator 文件系统结构的定义	5
2. 部署 WIPO Sequence Validator	6
2.1. 将 Validator 作为 Spring Boot JAR 启动	6
2.1.1. 将 Validator 作为可执行的应用程序进行部署	7
2.2. 作为 WAR Web 服务进行部署	8
3. 验证报告	9
4. 回调端点请求	10
4.1. 回调端点请求的格式	10
4.2. 验证报告	14
5. 配置	15
5.1. 默认设置	15
5.2. 配置验证规则 VXQV_49	17
5.3. 检查端点运行状况	17
5.4. 本地化消息	17
5.5. 自定义生物体名称	18
5.6. 引用 ST.26 DTD 文件	18
5.6.1. 如何就验证指明其他的 DTD 版本	18
6. Validator REST API	20
6.1. 验证 WIPO ST.26 文件	20
6.2. 请求验证状态	22
附件一：验证报告示例	23
附件二：完整的 API 规范（YAML）	24
附件三：属性名称（JSON）	24

1. 简介

WIPO Sequence Validator（下称 Validator 或工具）的主要目的是为知识产权局提供一项 web 服务，对 WIPO ST.26 格式的 XML 文件进行验证，以确保这类文件符合产权组织标准 ST.26。虽然使用 WIPO Sequence 桌面应用程序起草的序列列表符合 WIPO ST.26，但用户可以使用自认为最合适的任何工具。

本文件的目的是解释该工具的构成、部署、设置和文件用户系统，详细内容见以下各节。关于任何故障排除问题，请参见 wiki：

<https://www3.wipo.int/confluence/display/ST26software/Validator+Troubleshooting>

1.1. VALIDATOR 工作流程概述

该工具提供以下四个用例：

- 验证 WIPO ST.26 文件；
- 请求正在运行的验证的状态；
- 更新配置文件（仅限知识产权局管理员）；以及
- 当验证过程完成后，用验证过程的结果调用一个回调端点。

注：这个回调端点¹不在 Validator 的范围内。由开发和设置这项服务的主管局来建立这个端点。

该工具由可以作为 Web 服务执行的 JAR 文件组成，或由可以部署在 Tomcat 服务器上的 WAR 文件组成。

在两种情况下，为验证 WIPO ST.26 序列列表，该工具会从本地文件系统中使用文件，生成带有验证结果的验证报告，并可选择通过调用回调端点返回验证过程的结果，即验证报告。

Validator 的主要工作流程如下：

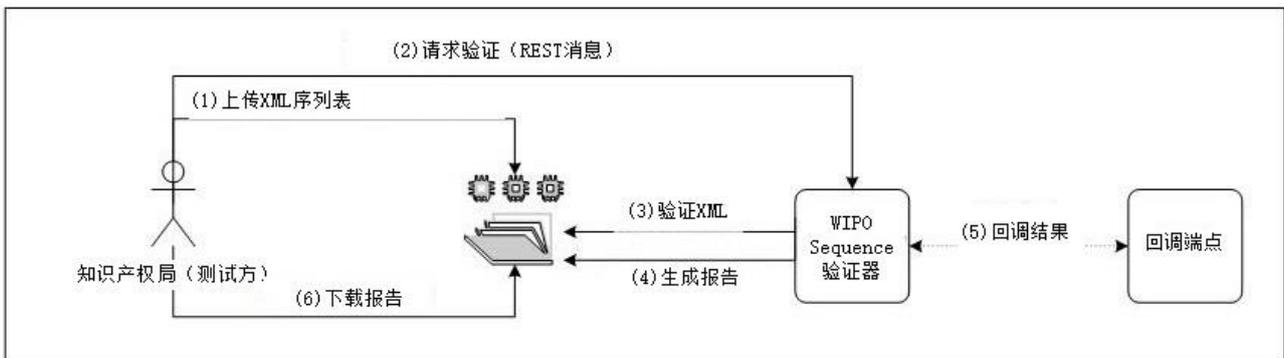
- a) 各知识产权局信息技术系统将 WIPO ST.26 XML 文件保存在默认的“Inbox”文件夹或请求中指定的文件夹中。
- b) 知识产权局系统启动 HTTP Post，请求对文件进行验证。根据配置情况，知识产权局系统可以请求对文件进行“full”或“formality”验证。“formality”验证过程将检查 ST.26 文件是否为 XML 文件，并根据 ST.26 DTD 对文件进行验证。“full”验证过程将比照业务验证规则对 ST.26 文件进行验证（验证规则来自 ST.26 的内容），并执行“formality”验证过程。

注：建议“formality”验证过程仅针对在线申请受理系统使用，因为可以同步执行，而批量处理时建议使用“full”验证，因为会花费更长时间。

- c) 验证完成后，将提供一条响应，说明文件是否通过了“formality”验证，如果知识产权局信息技术系统选择了“full”验证，则另外说明业务规则验证过程是否已经正确启动。
- d) 如果 Validator 正在进行“full”验证，它将从“Inbox”文件夹中调取 XML 文件，并启动业务规则验证过程，然后进行以下操作：

¹ 此处的回调端点是以 URI 识别的一个唯一地址，可向其发送请求信息。

- e) Validator 在指定的“Input”文件夹中生成一个基于 XSD 架构的 XML 报告文件（“report_<filename>.xml”），并将经过验证的 WIPO ST.26 XML 文件移至“Outbox”文件夹。可供选择的是，验证器也同时产生该验证报告的 HTML 版本。
- f) XML 报告文件一旦被创建并被检查是否符合 XSD 架构后，Validator 就会在名为“Reports”的指定文件夹下创建一个 HTML 报告文件。
- g) 在业务规则验证过程完成后，如果配置了回调端点，则会从 Validator 中调用回调端点，并且请求中会填充与验证过程相关的附加信息。请求结构和一些样本数据在下文第 4 节提供。
- h) 回调端点应该在响应中返回一个空代码或成功代码（无错误）。[注：这一步只有在外部 web 服务可用且调用已在 Validator 中配置好的情况下才会执行。]Validator 和回调端点之间也需要连接。如上所述，外部 web 服务不构成 Validator 的一部分，应由各知识产权局根据下文定义的合约进行开发和配置。
- i) 知识产权局系统可以从“Reports”文件夹中检索到验证报告。



注：WIPO Sequence Validator 符合关于使用 Web API 处理和传送知识产权数据的标准：[WIPO ST.90](#)。

1.2. 关于 VALIDATOR 文件系统结构的定义

Validator 使用的文件系统结构由五个文件夹组成：

- **“Inbox” 文件夹：**这是存储知识产权局提供验证的 WIPO ST.26 文件的本地文件夹。
- **“Process” 文件夹：**这是“Inbox”中的文件在处理过程中临时经过的本地文件夹。它包括两个子文件夹：
 - **“Full validation” 文件夹：**存储有待全面验证的文件
 - **“Formality validation” 文件夹：**存储有待形式验证的文件
- **“Outbox” 文件夹：**验证完成后，应用程序将 WIPO ST.26 文件的源文件储存在该本地文件夹中。
- **“Reports” 文件夹：**这是将验证结果保存在验证报告文件中的本地文件夹。
- **“Params” 文件夹：**这是一个本地文件夹，是带有验证请求中所有验证参数的 JSON（.json）文件所在的位置，参数的目的是为异步深度验证过程提供参数。

文件系统结构示例如下：

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

[重要提示：默认情况下，/temp/ST26 目录应位于工具所在的父目录中。例如，如果 JAR 或 WAR 文件位于 C:/dev，则文件夹结构应被创建为 C:/temp/ST26/...]

2. 部署 WIPO Sequence Validator

如前所述，Validator 以下列两种二进制格式之一提供。根据主管局部署 Validator 时要采用的基础设施类型，知识产权局将从两种二进制格式中优先选择一种。

Validator 提供的两个二进制格式是：

- **二进制 SpringBoot JAR:** 这种二进制是一个可执行的 JAR 文件，需要安装 [Java 8](#)。
- **War 包二进制:** 这种二进制是为了在 Servlet 容器上部署，需要与 Spring Boot 2 和 Servlet Spec 3.1+兼容的应用程序服务器，例如 [Tomcat 8.5](#)。

以下各节将详细介绍将 Validator 部署为 [Spring Boot](#) 应用程序，或部署为 Java 应用程序服务器中的 WAR。

2.1. 将 VALIDATOR 作为 SPRING BOOT JAR 启动

Spring Boot JAR 包含一个嵌入式服务器，它允许部署 Validator API，而无需另外的服务器。这极大地简化了基础设施层面的配置和部署。

为运行嵌入式服务器，应执行以下命令。

注：服务器上必须已经安装 Java 8。由于 Java 不保证使用 UTF-8，系统属性“file.encoding”必须设置为“UTF-8”。这可以通过包含以下内容来实现：

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

Validator API 可以通过 [Swagger UI](#) 访问：

[http://\[host-name\]:8080/swagger-ui.html](http://[host-name]:8080/swagger-ui.html)

Validator API 可以通过以下端点访问：

[http://\[host-name\]:8080/api/\[version\]/status](http://[host-name]:8080/api/[version]/status)

[http://\[host-name\]:8080/api/\[version\]/validate](http://[host-name]:8080/api/[version]/validate)

其中，知识产权局必须做出以下修改：

[host-name]必须替换为服务器主机名；并且

[version]应当替换为 Validator API 的版本（如 v1.0）。

默认情况下，服务器将在端口 8080 上运行，要改变端口，应添加下面所示的“--server.port”命令行选项：

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

默认情况下，Validator 将使用 Java 虚拟机（JVM）默认内存设置。默认的堆大小上限是可用物理内存的四分之一。

要修改堆大小上限，在使用命令行执行时必须使用“-Xmx”选项²：

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

² <https://docs.oracle.com/javase/8/docs/technotes/tools/windows/java.html#BABHDABI>

2.1.1. 将 **Validator** 作为可执行的应用程序进行部署

Validator 也可以作为由操作系统管理的服务来安装，以便支持其在操作系统启动时执行。

这种方式可用于为 **WIPO Sequence** 支持的所有平台——Windows、Linux 和 Mac OS——配置 Spring Boot JAR 文件。

以下指南详细介绍了如何为每个操作系统创建执行 JAR 文件的系统服务，还提供了关于如何配置这项服务的不同选项并执行应用程序的信息。

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

2.2. 作为 WAR WEB 服务进行部署

关于提供的第二种二进制，WAR 包可以部署在现有的 Java 应用程序服务器中，如 Apache Tomcat 8.5。

注：需要与 Servlet 3.1 兼容的容器。

下面的说明是针对 Tomcat 应用程序服务器的。这里的“\$TOMCAT_ROOT”指的是 Tomcat 服务器的根目录，这个值应当用文件路径的相关值代替。

- a. 停止服务器：\$TOMCAT_ROOT\bin\catalina.bat stop
- b. 复制 war 到 \$TOMCAT_ROOT\webapps\wipo-sequence-validator.war
- c. 启动服务器：\$TOMCAT_ROOT\bin\catalina.bat start

注：由于 Java 不保证使用 UTF-8，所以在启动应用程序服务器时，系统属性“file.encoding”必须设置为“UTF-8”。可以通过包含以下内容来实现：-D"file.encoding=UTF-8"

上文指出，Validator API 可以通过 Swagger UI 进行访问：

<http://host-name:8080/wipo-sequence-validator/swagger-ui.html>

Validator API 可以通过以下端点访问：

[http://\[host-name\]:8080/wipo-sequence-validator/api/\[version\]/status](http://[host-name]:8080/wipo-sequence-validator/api/[version]/status)

[http://\[host-name\]:8080/wipo-sequence-validator/api/\[version\]/validate](http://[host-name]:8080/wipo-sequence-validator/api/[version]/validate)

其中，知识产权局必须做出以下修改：

- [host-name]必须替换为服务器主机名；并且
- [version]应当替换为 API 的版本（如 v1.0）。

默认情况下，服务器将在端口 8080 上运行。如果要将其改为其他端口，应按照这里提供的说明修改 Tomcat 配置文件：

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

默认情况下，Validator 将使用 JVM 默认内存设置。默认的堆大小上限是可用物理内存的四分之一。

为了修改堆大小上限，在使用命令行执行时必须使用“-Xmx”选项，如上文第 2.1 节所述。

3. 验证报告

验证报告生成时可以采取两种格式：XML 和 HTML（或者两种均使用）。

该工具生成的验证报告为 XML 格式，所用模板如下：

在根级别显示的属性如下：

```
<?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-seq1-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>
  ...
```

- “applicationNumberText”：与该序列列表相关联的申请；
- “productionDate”：进行验证的日期；
- “filingDate”：提交申请的日期；
- “softwareBuildVersion”：验证使用的 Validator 的版本；
- “softwareVersion”：用于生成序列列表的 WIPO Sequence 的版本；和
- “sourceFileName”：序列列表 XML 示例的名称。

本手册附件一提供了该验证报告的示例，附件三提供了这些组件允许填入的值。就指明的严重级别而言，请注意以下类别：

- ERROR – 在“full”验证时返回错误
- WARNING – 在“full”验证时返回警告
- XML_ERROR – 在“formality”验证时返回错误
- XML_WARN – 在“formality”验证时返回警告

验证报告也以 HTML 格式制作，使用的样式表与 WIPO Sequence 中使用的相同。HTML 验证报告示例在本手册附件四中提供。

4. 回调端点请求

回调端点向 Validator 发出的请求应包括以下参数，详细说明文件位置和验证过程：

```
{
  "currentApplicationNumber": "string",
  "currentSEQVersionNumber": "string",
  "parentApplicationNumber": "string",
  "parentSEQVersionNumber": "string",
  "seqInputLocation": "C:/temp/valid2Warning.xml",
  "verificationReportOutputPath": "C:/temp/report.xml",
  "nameFile": "valid2Warning.xml",
  "type": "full"
}
```

验证请求的“seqInputLocation”字段应设置为指明要验证的 XML 序列列表的路径。如果主管局将此字段留空，该工具将尝试验证默认“Inbox”文件夹中以“nameFile”为文件名的 XML 文件。“nameFile”参数识别了要验证的序列列表文件。

根据该工具生成的 application.property 文件的设置，验证请求中的“verificationReportOutputPath”将提供该工具生成的验证报告（.xml 和/或.html）的文件位置。如果用户将该字段留空或引入一个无效的文件路径，验证报告将被存储在默认的“Reports”文件夹中。

4.1. 回调端点请求的格式

如果配置了“api.URL”属性，Validator 将尝试把验证结果发送到指明了 URL 的端点。

为了与 Validator 进行通信，回调端点必须符合以下 Web 服务合约（YAML）：

https://www.wipo.int/standards/en/sequence/callback_end_point_web_service_contract.yml

此外，该请求应当是具有以下结构的 JSON 对象：

```
{
  "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, "seqType": "string", "startTime": "string", "totalErrorQuantity":
  0,
  "totalWarningQuantity": 0,
  "verificationReportOutputPath": "string"
}
```

这是一个 JSON 实例的示例，它将被发送到对 Validator 作出了调用的外部端点：

```
{
  "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,
  "seqlInputQuantity": 3,
  "seqlDQuantity": 3,
  "errorSummary": [
    {
      "index": 0, "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"
    },
  ],
}
```

```
    "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. 验证报告

如 3 所述，在验证之后，生成的验证报告位于 “verificationReportOutputPath”，XML 报告文件的默认位置是：/temp/st26/reports/[verificationID]/report.xml，HTML 报告文件的默认位置是：/temp/st26/reports/[verificationID]/report.html。

在 application.properties 文件中，可以启用或禁用 HTML 报告的生成。值为 “true” 时启用报告生成，值为 “false” 时禁用报告生成。

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

该报告的内容被发送到 “ServiceRequest” 的 “errorSummary” 字段中的回调端点。上文第 4 节给出的请求示例中提供了这个字段的示例。

5. 配置

5.1. 默认设置

Validator 是通过使用一个属性文件来进行配置的。默认的“application.properties”文件有以下值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
```

³ 2023 年 2 月生效。

```
#software version from pom.xml
app.version=@project.version@
```

```
## -- 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 concurrentlyprocessing.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
```

要修改此处提供的参数值，应使用另外的“application.properties”文件。在 Spring Boot 文档中详细介绍了几个选项：<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>。

最简单的选项是提供一个新的“application.properties”文件，这个文件按操作顺序在以下位置被搜索：

- (a) 当前目录下的“/config”文件夹[注：如果 Validator 在 Tomcat 中作为 WAR 文件部署，这个文件夹将在“lib”文件夹下，例如“/opt/apache-tomcat/lib/config”]；
- (b) 当前目录[注：如果 Validator 在 Tomcat 中作为 WAR 文件部署，这个文件夹将在“lib”文件夹下，例如“/opt/apache-tomcat/lib/”]；
- (c) \$classpath 或 config 包；然后是
- (d) \$classpath 根目录。

另外，启动该工具时，可以通过在命令行建立参数来指定配置文件的路径和名称：

对于 JAR 部署：

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

对于 Tomcat 上的 WAR 部署，需要在 CATALINA_OPTS 中添加以下条目：

- export CATALINA_OPTS="-Dspring.config.location=<PATH_TO_FILE>

使用 WAR 部署时，也可以将新的“application.properties”文件复制到 Web 应用程序的“WEB-INF/classes”文件夹中，或者对现有的进行编辑。

5.2. 配置验证规则 VXQV_49

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

如果用户想要启用规则 VXQV_49，可以将 `application.properties` 文件中的 `optionalEnglishQualifierValue` 值设置为 `'true'`，并可以通过更新 `optionalRuleType` 的值为 `'ERROR'` 或 `'WARNING'` 来设置规则的严重级别。这两个属性的默认值如上所示。

5.3. 检查端点运行状况

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

Validator 服务已执行一个 `/health` 端点，提供关于应用程序“运行状况”的基本信息。

要了解 `/health` 端点，URL 为 <http://localhost:8080/wipo-sequence-validator/actuator/health> URL。端点应显示以下内容：

- 只要应用程序运行良好，状态将为 `'UP'`。
- 如果由于与数据库连接或磁盘空间不足等任何问题，导致应用程序运行不正常，则会显示 `'DOWN'`。

`/health` 端点只简单显示 `'UP'` 或 `'DOWN'` 的状态。`application.properties` 文件的以下属性提供了完整详细信息，包括作为检查运行状况过程一部分而被检查的每个运行状况指标的状态。

```
# Show details of health endpoint
management.endpoint.health.show-details=always
```

`/health` 端点现在包括作为运行状况检查过程一部分而运行的“`DiskSpaceHealthIndicator`”的详细信息。

`/health` 端点将像这样显示为一系列键值对，并包含更多详细信息。如下例所示：

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

5.4. 本地化消息

Validator 可以提供本地化消息，例如在验证报告中，可用 PCT 十种正式语言（阿拉伯文、德文、俄文、法文、韩文、葡萄牙文、日文、西班牙文、英文和中文）之一提供消息。

默认情况下，这些消息以英文提供。要将 Validator 配置为以其他语言提供消息，“`application.properties`”文件中的“`validator_locale`”参数必须设置为适当的语言代码。

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

注：为了应用在新的“application.properties”文件中设置的属性，应重新启动 Validator。

5.5. 自定义生物体名称

为便于主管局提供各自的自定义生物体名称（这些名称不是原始预定义生物体名称列表的组成部分），可以通过在“alternativeResourceBasePath”文件夹中创建名为“custom_organism.json”的新文件，来提供自定义生物体列表。该文件应具备如下结构：

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

注：与预定义生物体名称列表不同，所有生物体都包含在一个 JSON 文件中，而不是按字母表的每个字母分开放入一个 JSON 文件中。

5.6. 引用 ST.26 DTD 文件

默认情况下，Validator 引用 ST.26 DTD 最新版本。WIPO Sequence Validator 的当前版本基于 WIPO ST.26 DTD³1.3 版本⁴。

这版最新的 ST.26 DTD 包含在 Validator 库中，Validator 库位于源代码的“/src/main/resources”文件夹（这是 JAR 或 WAR 文件中引用的已定义文件路径）。它在上述文件夹的“catalog.xml”文件中被引用，如下所示：

```
<?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>
```

关于如何添加新的 DTD 的说明，详见下文。在验证过程中，将使用 XML 文件的 DOCTYPE 声明中设置的 DTD 版本。首先，“publicId”将被用来识别要使用的 DTD 文件的位置。如果“publicId”不包含在目录中，系统将尝试在 Java 进程正在被执行的根文件夹中定位 DTD 文件。

5.6.1. 如何就验证指明其他的 DTD 版本

为了能够验证引用旧版 ST.26 DTD 的 WIPO ST.26 文件，必须向 Validator 提供这个 ST.26 DTD 文件，以便进行适当的验证。

要实现这一目标，有两种可供选择的方法：

- 解压 JAR 文件，并在“src/main/resources”文件夹中包含对附加的或其他的 ST.26 DTD 文件的引用。
- 修改“catalog.xml”文件，为附加的 ST.26 DTD 添加新条目或编辑现有条目。

例如：

```
<?xml version="1.0" encoding="UTF-8"?>
```

⁴ 自 2023 年 2 月生效

```
<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>
```

无需修改 JAR 文件，但应遵循以下步骤：

- a. 将“catalog.xml”和所有 DTD 复制到本地文件夹中；
- b. 修改“catalog.xml”，纳入对附加的 ST.26 DTD 的引用；并且
- c. 在启动时设置该 Java 系统属性：xml.catalog.files=<path_to_catalog.xml>

注：在 Tomcat for Windows 中可以通过添加以下环境变量来实现：

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

[重要提示：添加不同版本的 ST.26 DTD 将允许根据所引用的 ST.26 DTD 对 XML 文件进行‘formality’验证，但‘full’验证可能需要修改源代码，以便执行检验规则。因此，建议仅在进行‘formality’验证时使用多个 DTD。]

6. Validator REST API

本节明确介绍了 Validator API 用例。有三种服务或用例：

- 验证 “Inbox” 文件夹中的文件（6.1 节）；
- 验证作为请求组成部分的文件（6.1 节）；以及
- 请求验证状态（6.2 节）。

本项服务的 API 规范（用 OAS 3.0 [YAML File]编写）完整内容见附件二。

6.1. 验证 WIPO ST.26 文件

请求映射	/api/v1/validate
方法	POST
使用	application/json
制作	application/json
操作	请求对 “Inbox” 文件夹中现有的 WIPO ST.26 文件进行验证。 返回一个唯一的 “verificationID”，用于检索验证请求的状态。
请求	<pre>{ "currentApplicationNumber": "string", "currentSEQLVersionNumber": "string", "parentApplicationNumber": "string", "parentSEQLVersionNumber": "string", "seqlInputLocation": "string" (Location of Input.xml file), "verificationReportOutputPath": "string" (Destination of report.xml), "nameFile": "file.xml", "type": "string" (Possible values: full formality), }</pre>
响应	<ul style="list-style-type: none"> • '202': “Accepted”。WIPO ST.26 文件通过了形式验证，业务规则检验已启动。该代码还将包括一条响应消息，注明检索验证报告的唯一代码（“verificationID”）。WIPO ST.26 文件被移到 “Process” 文件夹进行处理。 • '400': “Bad request”。REST 请求的格式不正确，或 WIPO ST.26 文件没有通过 XML 验证。该代码将以一条响应消息作为补充，说明错误的详细情况。

	<ul style="list-style-type: none"> • '404': “Not Found”。在 “Inbox” 文件夹中没有找到 WIPO ST.26 文件。 • '500': “Internal Server Error”。发生内部错误。该代码将以一条响应消息作为补充，说明错误的详细情况。
前置条件	必须在用户指定的 “Inbox” 文件夹中提供 WIPO ST.26 文件。
后置条件	<p>WIPO ST.26 文件被移至 verificationReportOutputPath 提供的位置，或在以下位置的 “Outbox” 文件夹：</p> <p>“/[outbox]/[verificationID]/[file.xml]”</p> <p>生成验证报告的位置：</p> <p>“/[reports]/ [verificationID]/report_[file.xml or file.html]”</p>

注：经验证的文件名可以包括非拉丁字符，但不接受任何保留字符。

6.2. 请求验证状态

请求映射	/api/v1/status
方法	POST
使用	application/json
制作	application/json
操作	请求特定 WIPO ST.26 文件的验证状态。
请求	<pre>{ verificationID: {type: string} }</pre>
响应	<ul style="list-style-type: none"> • '200': 成功。该代码还将包括一条响应信息，注明一个独特的 ID，这个 ID 表示验证过程的状态，选自以下几种： <ul style="list-style-type: none"> ○ “RUNNING”：文件正在得到处理 ○ “FINISHED-VALID”：文件成功通过形式验证，验证结果在报告文件夹中 ○ “FINISHED-INVALID”：过程已完成，但文件未通过形式验证。验证结果在报告文件夹中 ○ “NOT_FOUND”：未找到“verificationID” ○ “VERIFICATION_ID_ERROR”：“verificationID”未列入请求 • '400': 错误请求。REST 请求的格式不正确。 • '500': 服务器错误。发生内部错误。该代码还将包括一条响应消息，说明错误的详细情况
后置条件	Validator 提供验证状态。
假设	-

[后接附件一]

附件一：XML 验证报告示例

```
<?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>
```

该 XML 示例也可以在此下载：

https://www.wipo.int/standards/en/sequence/example_verification_report.xml

[后接附件二]

附件二：完整的 API 规范（YAML）

完整的 API 规范可以在此下载：

https://www.wipo.int/standards/en/sequence/complete_validator_api_specification.yml

[后接附件三]

附件三：属性名称（JSON）

该工具内使用的属性名称可以在此下载：

<https://www.wipo.int/standards/en/sequence/propertynames.json>

请注意这是部分源代码，并在执行过程中根据需要进行更新。

[后接附件四]

附件四：HTML 验证报告示例

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	

[文件完]