



WIPO Sequence Validator

Manual de Utilização

Versão 2.3.0

Este documento tem por objetivo auxiliar os Institutos de Propriedade Intelectual a implantar o WIPO Sequence Validator como um serviço web e também a configurar o Validator.

Índice

1. Introdução	3
1.1. Visão geral do fluxo de trabalho do Validator	3
1.2. Definição da estrutura do sistema de arquivos do Validator	6
2. Implantação de WIPO Sequence Validator	7
2.1. Iniciação do Validator como um Spring Boot JAR	7
2.1.1. Implantação do Validator como um aplicativo executável	8
2.2. Implantação como um serviço web WAR	9
3. Relatório de verificação	10
4. Solicitação pelo ponto final de chamada de retorno	11
4.2. Relatório de verificação	15
5. Configuração	16
5.1. Configurações predefinidas	16
5.2. Configuring verification rule VXQV_49	18
5.3. Verificação do estado de funcionamento do ponto final	18
5.4. Mensagens localizadas	19
5.5. Nomes de organismos personalizados	19
5.6. Referenciamento de arquivos DTD ST.26	19
5.6.1. Como indicar uma versão alternativa de DTD para a validação	20
6. API REST do Validator	21
6.1. Validação de um arquivo no formato ST.26 da OMPI	21
6.2. Solicitação do status de validação	23
Anexo I: Exemplo de relatório de verificação em XML	24
Anexo II: Especificação completa da API (YAML)	25
Anexo III: Nomes de propriedades (JSON)	25
Anexo IV: Exemplo de relatório de verificação em HTML	25

1. Introdução

WIPO Sequence Validator (doravante denominado "Validator" ou "a ferramenta") tem por principal objetivo fornecer aos Institutos de PI (IPIs) um serviço web de validação de arquivos XML no formato ST.26 da OMPI a fim de assegurar sua conformidade com a norma ST.26 da OMPI. Embora uma listagem de sequências que tenha sido redigida por meio do aplicativo WIPO Sequence para computadores será conforme com a norma ST.26 da OMPI, os usuários podem usar quaisquer outras ferramentas que considerarem mais apropriadas.

Este documento tem por objetivo explicar a estrutura, a implantação, as configurações e o sistema consumidor de arquivos da ferramenta detalhada nas seções seguintes. Em caso de dúvidas sobre resolução de problemas, consulte o wiki em:

<https://www3.wipo.int/confluence/display/WVSVAL/Troubleshooting>

1.1. Visão geral do fluxo de trabalho do Validator

A ferramenta disponibiliza os quatro casos de uso seguintes:

- Validação de um arquivo no formato ST.26 da OMPI;
- Solicitação do status de uma validação em curso;
- Atualização de arquivos de configuração (somente o administrador do IPI); e
- Chame um ponto final de chamada de retorno com o resultado do processo de validação assim que o processo for concluído.

Nota: este ponto final de chamada de retorno¹ está fora do escopo do Validator. Cabe aos Institutos que desenvolvem e instalam este serviço estabelecer o ponto final de chamada.

A ferramenta se compõe de um arquivo JAR que pode ser executado como um serviço web ou de um arquivo WAR que pode ser implantado em um servidor Tomcat.

Em ambos os casos, para validar uma listagem de sequências no formato ST.26 da OMPI, a ferramenta consome arquivos de um sistema local de arquivos e gera um relatório de verificação contendo os resultados da validação e, opcionalmente, retorna os resultados do processo de validação, quer dizer o relatório de verificação, chamando um ponto final de chamada de retorno ("callback endpoint").

O principal fluxo de trabalho do Validator é o seguinte:

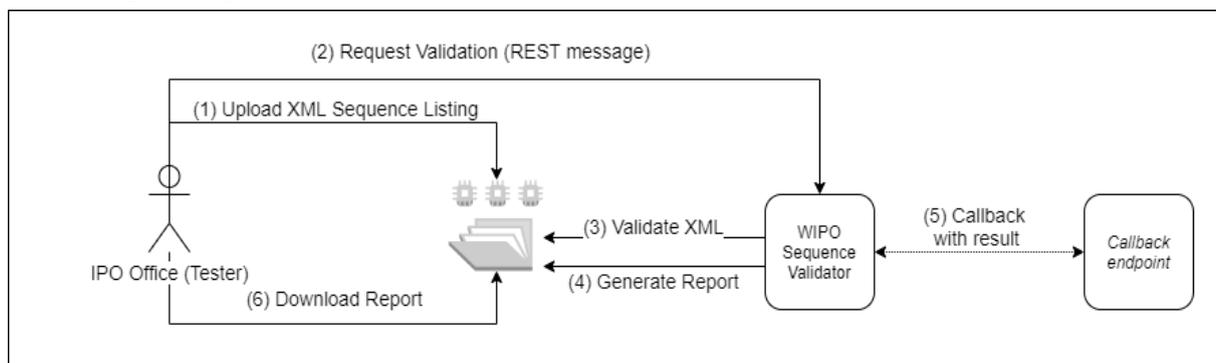
- a) O sistema de TI do IPI respectivo salva um arquivo XML no formato ST.26 da OMPI numa pasta de caixa de entrada ("Inbox") predefinida ou na pasta especificada na solicitação.
- b) O sistema do IPI inicia uma postagem por HTTP solicitando a validação do arquivo. Dependendo da configuração, o sistema do IPI pode solicitar uma validação "completa" ou uma validação "formal" do arquivo. O processo de validação "formal" do arquivo verifica se o arquivo no formato ST.26 é um arquivo XML e valida o arquivo com relação ao DTD ST.26. O processo de validação "completa" valida o arquivo no formato ST.26 com relação às regras de verificação de negócios derivadas do conteúdo da norma ST.26, além de conduzir o processo de validação "formal".

¹ Um ponto final de chamada de retorno ("callback endpoint") aqui é um endereço único identificado por um URI para o qual mensagens de solicitação podem ser enviadas

Nota: Recomenda-se o uso do processo de validação "formal" apenas para um sistema de aceitação de depósitos online, já que este processo pode ser executado de forma síncrona, enquanto que a validação "completa" é recomendada para o processamento em lotes, já que demora muito mais tempo.

- c) Uma vez a validação concluída, uma resposta é fornecida indicando se a validação "formal" do arquivo foi bem-sucedida ou não e, no caso em que o sistema de TI do IPI

- d) selecionou a validação "completa", indicando além disso se o processo de validação das regras de negócios foi iniciado corretamente.
- e) Se o Validator estiver realizando uma validação "completa", recupera o arquivo XML na pasta de caixa de entrada ("Inbox"), inicia o processo de validação das regras de negócios, e executa as etapas seguintes:
- f) O Validator gera um arquivo XML contendo o relatório ("report_<filename>.xml") baseado no esquema XSD numa pasta de entrada "Input" especificada e move o arquivo XML validado no formato ST.26 da OMPI para uma pasta de saída ("Outbox"). Opcionalmente, o Validator também produz ao mesmo tempo uma versão deste relatório de verificação em HTML.
- g) Assim que o relatório de verificação em XML foi criado e sua conformidade com o esquema XSD foi verificada, o Validator cria um arquivo HTML contendo o relatório numa pasta especificada com o nome "Reports".
- h) Após a conclusão do processo de validação da regra de negócios, o ponto final, se configurado, é chamado de volta pelo Validator e a solicitação é povoada com informações adicionais relacionadas com o processo de validação. A estrutura da solicitação e alguns dados de exemplo são dados na seção 4 abaixo.
- i) O ponto final da chamada de retorno ("callback endpoint") deve retornar um código vazio ou de êxito na resposta (nenhum erro). [Nota: Esta etapa é executada somente se o serviço web externo foi disponibilizado e se a chamada foi configurada no Validator.] A conectividade entre o Validator e o ponto final de chamada de retorno também é necessária. Como mencionado acima, o serviço web externo não faz parte do Validator e deve ser desenvolvido e configurado pelos Institutos de acordo com o contrato definido abaixo.
- j) O sistema do IPI pode recuperar o relatório de verificação na pasta dos relatórios ("Reports").



Nota: O WIPO Sequence Validator está em conformidade com a norma da OMPI de processamento e transmissão de dados relativos a propriedade intelectual por meio de Web APIs: [norma ST.90 da OMPI](#).

1.2. Definição da estrutura do sistema de arquivos do Validator

A estrutura do sistema de arquivos utilizado pelo Validator consiste em cinco pastas:

- **Pasta de caixa de entrada ("Inbox"):** É a pasta local onde os arquivos no formato ST.26 da OMPI são colocados por um IPI para validação.
- **Pasta de processamento ("Process"):** É a pasta local pela qual passam temporariamente os arquivos na caixa de entrada ("Inbox") durante o processamento. Esta pasta contém duas subpastas:
 - **Pasta de validação completa ("Full validation"):** onde são armazenados os arquivos aguardando uma validação completa.
 - **Pasta de validação formal ("Formality validation"):** onde são armazenados os arquivos aguardando uma validação formal.
- **Pasta de caixa de saída ("Outbox"):** Uma vez concluída a validação, o aplicativo armazena a fonte do arquivo no formato ST.26 da OMPI nesta pasta local.
- **Pasta de relatórios ("Reports"):** É a pasta local onde são salvos os resultados da validação em um ou mais arquivos contendo relatórios de verificação.
- **Pasta de parâmetros ("Params"):** É a pasta local onde um arquivo JSON (.json) com todos os parâmetros de validação contidos na solicitação de validação estão localizados para fornecer os parâmetros do processo assíncrono de validação profunda.

Abaixo é dado um exemplo de estrutura de um sistema de arquivos:

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

[IMPORTANTE: Por predefinição, o diretório /temp/ST26 deve estar localizado no diretório principal daquele onde a ferramenta está localizada. Por exemplo, se o arquivo JAR ou WAR está localizado em C:/dev, a estrutura da pasta criada deve ser C:/temp/ST26/...]

2. Implantação de WIPO Sequence Validator

Como indicado anteriormente, o Validator é fornecido num dos dois formatos binários listados abaixo. Dependendo do tipo de infraestrutura em que o Instituto deseja implantar o Validator, o IPI pode selecionar um ou outro tipo de binário, segundo sua preferência.

Os dois formatos binários nos quais o Validator é fornecido são:

- **Binary SpringBoot JAR:** Esse arquivo é um arquivo binário JAR executável. Requer que [Java 8](#) seja instalado.
- **Pacote binário War:** Esse arquivo binário destina-se a ser implantado num recipiente de servlet. Um servidor de aplicativos compatível com Spring Boot 2 e Servlet Spec 3.1+ é necessário, tal como [Tomcat 8.5](#).

As seções seguintes detalham a implantação do Validator como um app [Spring Boot](#) ou como um arquivo WAR em um servidor de aplicativos Java.

2.1. Iniciação do Validator como um Spring Boot JAR

O arquivo Spring Boot JAR contém um servidor incorporado, o que permite implantar o API do Validator sem necessitar um servidor separado. Isto simplifica muito a configuração e a implantação no nível da infraestrutura.

Para executar o servidor incorporado, o comando seguinte deve ser executado.

Nota: Java 8 já deve estar instalado no servidor; como Java não garante o uso de UTF-8, a propriedade "file.encoding" do sistema deve ser definida como "UTF-8". Isso pode ser feito incluindo o seguinte:

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

A API Validator pode ser acessada por meio de uma [interface de usuário Swagger](#):
[http://\[host-name\]:8080/swagger-ui.html](http://[host-name]:8080/swagger-ui.html)

A API Validator está acessível nos seguintes pontos finais:

[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)

onde o IPI deve efetuar as modificações seguintes:

- [host-name] deve ser substituído pelo nome do servidor host; e
- [version] deve ser substituído pela versão da API Validator (por exemplo v1.0).

Por predefinição, o servidor será executado na porta 8080; para modificar a porta, a linha de comando opcional " `--server.port`" deve ser adicionada, como indicado aqui:

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

Por predefinição, o Validator usará as configurações predefinidas da memória da Máquina Virtual Java (JVM). A área dinâmica para dados tem um tamanho máximo predefinido igual a um quarto da memória física disponível.

Para modificar o tamanho máximo da área dinâmica de dados, é preciso usar a opção "-Xmx" quando executando por meio da linha de comando²:

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

2.1.1. Implantação do Validator como um aplicativo executável

O Validator também pode ser instalado como um serviço gerenciado pelo sistema operacional para permitir sua execução como parte da inicialização do sistema operacional, por exemplo.

É possível configurar um arquivo Spring Boot JAR desta maneira para todas as plataformas compatíveis com WIPO Sequence: Windows, Linux e Mac OS.

O guia seguinte explica detalhadamente como criar um serviço de sistema que executa o arquivo JAR para cada sistema operacional. Também fornece informações sobre como configurar as diferentes opções do serviço e a execução do aplicativo:

<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. Implantação como um serviço web WAR

No caso do segundo tipo de formato binário fornecido, o pacote WAR pode ser implantado em um servidor existente de aplicativos Java, tais como Apache Tomcat 8.5.

Nota: Um recipiente compatível com Servlet 3.1 é necessário.

As instruções a seguir são para um servidor de aplicativos Tomcat. Aqui, "\$TOMCAT_ROOT" refere-se à pasta raiz do servidor Tomcat e este valor deve ser substituído pelo valor relevante para o caminho do arquivo:

- a. Parar o servidor: \$TOMCAT_ROOT\bin\catalina.bat stop
- b. Copiar WAR para \$TOMCAT_ROOT\webapps\wipo-sequence-validator.war
- c. Iniciar o servidor: \$TOMCAT_ROOT\bin\catalina.bat start

Nota: Como Java não garante o uso de UTF-8, a propriedade "file.encoding" do sistema deve ser definida como "UTF-8" durante a inicialização do servidor de aplicativos. Isso pode ser feito incluindo o seguinte:

-D"file.encoding=UTF-8"

A API Validator pode ser acessada por meio de uma interface de usuário Swagger, como indicado acima:

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

A API do Validator está acessível nos seguintes pontos finais:

[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)

onde o IPI deve efetuar as modificações seguintes:

- [host-name] deve ser substituído pelo nome do servidor host; e
- [version] deve ser substituído pela versão da API (por exemplo v1.0).

Por predefinição, o servidor funciona pela porta 8080. Para transferir para outra porta, o arquivo de configuração de Tomcat deve ser modificado seguindo as instruções fornecidas aqui:

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

Por predefinição, o Validator usará as configurações predefinidas da memória da Máquina Virtual Java (JVM). A área dinâmica para dados tem um tamanho máximo predefinido igual a um quarto da memória física disponível.

Para modificar o tamanho máximo da área dinâmica de dados, deve-se usar a opção "-Xmx" quando executando por meio da linha de comando, como indicado acima na seção 2.1.

3. Relatório de verificação

Existem dois formatos nos quais o relatório de verificação pode ser gerado: XML e HTML (ou ambos).

O relatório de verificação gerado pela ferramenta é no formato XML e o modelo usado é fornecido abaixo:

```
<?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>
  </VerificationMessageBag>
</VerificationReport>
```

Os atributos indicados no nível raiz são os seguintes:

- 'applicationNumberText': the application associated with this sequence listing;
- 'productionDate': data em que a validação foi realizada;
- 'filingDate': data de depósito do pedido;
- 'softwareBuildVersion': versão do Validator utilizada para a validação;
- 'softwareVersion': versão de WIPO Sequence utilizada para gerar a listagem de sequências; e
- 'sourceFileName': nome da instância XML da listagem de sequências.

Um exemplo deste relatório de verificação é fornecido no Anexo I deste manual, com os valores permitidos para estes elementos no Anexo III.

No que diz respeito ao grau de severidade indicado, observe as categorias seguintes:

- ERROR - um erro retornado durante a verificação completa ("full")
- WARNING - um aviso retornado durante a verificação completa ("full")
- XML_ERROR - um erro retornado durante a verificação formal ("formality")
- XML_WARN - um erro retornado durante a verificação formal ("formality")

Esse relatório de verificação também é gerado no formato HTML, usando a mesma folha de estilo utilizada em WIPO Sequence. Um exemplo de relatório de verificação em HTML é fornecido no Anexo IV deste manual.

4. Solicitação pelo ponto final de chamada de retorno

A solicitação endereçada ao Validator pelo ponto final de chamada de retorno ("callback endpoint") deve incluir os parâmetros seguintes que detalham as localizações dos arquivos e o processo de

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

validação:

O campo "seqInputLocation" da solicitação de validação deve ser definido para indicar o caminho até a listagem de sequências XML a ser validada. Se o Instituto deixar este campo em branco, a ferramenta tentará validar o arquivo XML com o nome de arquivo "nameFile" localizado na pasta predefinida de caixa de entrada ("Inbox"). O parâmetro "nameFile" identifica o arquivo contendo a listagem de sequências a ser validada.

O parâmetro "verificationReportOutputPath" na solicitação fornece a localização do arquivo contendo o relatório de verificação (.xml e/ou .html), dependendo da configuração do arquivo "application.property" gerado pela ferramenta. Se o usuário deixar o campo em branco ou introduzir um caminho de arquivo inválido, o relatório de verificação será armazenado na pasta predefinida de relatórios, "Reports".

4.1. Formato da solicitação enviada pelo ponto final de chamada de retorno ("callback endpoint")

Se a propriedade "api.URL" estiver configurada, o Validator tentará enviar os resultados da validação para um ponto final tendo o URL especificado.

Para comunicar com o Validator, o ponto final de chamada de retorno deve estar em conformidade com o contrato de serviços web (YAML) seguinte:

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

Além disso, a solicitação deve ser um objeto JSON tendo esta estrutura:

```
{
  "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"
}
```

Este é um exemplo de instância JSON que será transmitido ao ponto final externo que chamou o Validator:

```
{
  "processID": "1608194222169dvVE",
  "seqIType": "ST.26",
  "httpStatus": "SUCCESS",
  "currentApplicationNumber": "string",
  "currentSQLVersionNumber": "string",
  "parentApplicationNumber": "string",
  "parentSQLVersionNumber": "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@5887
858, "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. Relatório de verificação

Conforme mencionado na seção 3, após a validação, o relatório de verificação gerado está localizado em "verificationReportOutputPath", que por predefinição para o arquivo do relatório em XML está em: "/temp/st26/reports/[verificationID]/report.xml" e para o arquivo do relatório em HTML está em:

"/temp/st26/reports/[verificationID]/report.html".

No arquivo "application.properties", você pode ativar ou desativar a geração de relatórios HTML. O valor "true" ativa a geração de relatórios e o valor "false" a desativa.

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

O conteúdo deste relatório é enviado ao ponto final da chamada de retorno ("callback endpoint") no campo "errorSummary" do "ServiceRequest". Um exemplo deste campo é fornecido nos exemplos de solicitações fornecidos acima na seção 4.

5. Configuração

5.1. Configurações predefinidas

O Validator é configurado por meio de um arquivo de propriedades. O arquivo predefinido "application.properties" tem os valores seguintes: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

## --PREFERÊNCIAS --
#Para ativar a regra VXQV49, defina este valor como verdadeiro; o valor por
predefinição é falso. 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
```

³ Válido em fevereiro de 2023

```
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 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
```

Para modificar o valor dos parâmetros fornecidos aqui, é preciso usar um arquivo "application.properties" alternativo. As diferentes opções existentes estão detalhadas na documentação de Spring Boot: <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>

A opção mais simples seria de fornecer um novo arquivo "application.properties" que será procurado nas localizações seguintes, por ordem de execução:

- a) Uma pasta "/config" no atual diretório *[Nota: se o Validator for implantado como um arquivo WAR em Tomcat, esta pasta estará na pasta "lib", por exemplo "/opt/apache-tomcat/lib/config"]*;
- b) O atual diretório *[Nota: se o Validator for implantado como um arquivo WAR em Tomcat, esta pasta estará na pasta "lib", por exemplo "/opt/apache-tomcat/lib/"]*;
- c) A `$classpath` or `config` package; then
- d) The `$classpath` root.

Além disso, é possível especificar o caminho e o nome do arquivo de configuração estabelecendo o parâmetro na linha de comando seguinte, ao iniciar a ferramenta:

No caso da implantação de um arquivo JAR:

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

No caso da implantação de um arquivo WAR em Tomcat, a seguinte entrada deve ser adicionada a CATALINA_OPTS:

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

Ao usar a implantação de WAR, o novo arquivo "application.properties" pode também ser copiado para a pasta "WEB-INF/classes" do aplicativo web, ou o arquivo atual é editado.

5.2. Configuring verification rule VXQV_49

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

Se o usuário deseja ativar a regra VXQV_49, pode definir o valor "optionalEnglishQualifierValue" no arquivo "application.properties" como "true" e pode definir a severidade da regra atualizando o valor "optionalRuleType" para "ERROR" ou "WARNING". Os valores predefinidos destas duas propriedades estão indicados acima.

5.3. Verificação do estado de funcionamento do ponto final

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

O serviço Validator implementou um "/health endpoint" que fornece informações de base sobre o estado de funcionamento do aplicativo.

Para explorar o ponto final "/health", o URL é <http://localhost:8080/wipo-sequence-validator/actuator/health> URL. O ponto final deve exibir o seguinte:

- O status será "UP" enquanto o estado de funcionamento do aplicativo for bom.
- Será "DOWN" se o aplicativo não estiver funcionando bem devido a algum problema, tais como conectividade com a base de dados ou falta de espaço no disco, etc.

O ponto final "/health" indica apenas um simples status "UP" ou "DOWN". A propriedade a seguir no arquivo "application.properties" fornece os detalhes completos, incluindo o status de cada indicador do estado de funcionamento verificado como parte do processo de verificação do

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

estado de funcionamento.

O ponto final "/health" agora inclui os detalhes do "DiskSpaceHealthIndicator" que é executado como parte do processo de verificação do estado de funcionamento.

O ponto final "/health" indicará assim uma série de pares de valores-chaves e incluirá mais detalhes. Um exemplo é dado abaixo:

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

5.4. Mensagens localizadas

O Validator pode fornecer uma mensagem localizada, por exemplo no relatório de verificação, em cada uma das dez línguas oficiais do PCT (árabe, chinês, inglês, francês, alemão, português, japonês, coreano, russo e espanhol).

Por predefinição, essas mensagens são fornecidas em inglês. A fim de configurar o Validator para que forneça essas mensagens em outras línguas, o parâmetro "validator_locale" do arquivo "application.properties" deve ser definido com o código da língua apropriado.

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

Nota: É preciso reiniciar o Validator para que as propriedades definidas no novo arquivo "application.properties" sejam aplicadas.

5.5. Nomes de organismos personalizados

Os Institutos podem fornecer seus próprios nomes de organismos personalizados numa lista de organismos personalizados que não fazem parte da lista original predefinida de nomes de organismos, criando um novo arquivo, chamado "custom_organism.json", na pasta "alternativeResourceBasePath". Este arquivo deve ter a estrutura seguinte:

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

Nota: Ao contrário da lista predefinida de nomes de organismos, todos os organismos estão contidos em um único arquivo JSON, em vez de serem separados em um arquivo JSON para cada letra do alfabeto.

5.6. Referenciamento de arquivos DTD ST.26

Por predefinição, o Validator referencia a versão mais recente do DTD ST.26. A versão atual de WIPO Sequence Validator é baseada na versão 1.3 do DTD³ ST.26 da OMPI.

Esta cópia da versão mais recente do DTD ST.26 está incluída na biblioteca do Validator localizada na pasta "/src/main/resources" do código fonte (é o caminho de arquivo definido e referenciado no arquivo JAR ou WAR). Está referenciado no arquivo "catalog.xml" na mesma pasta, como ilustrado

```
<?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>
```

abaixo:

As instruções sobre como incluir um novo DTD estão detalhadas abaixo. Durante a validação, a versão do DTD definida na declaração DOCTYPE do arquivo XML será usada. Primeiramente, o "publicId" será usado para identificar a localização do arquivo DTD a ser usado. Se "publicId" não estiver incluído no catálogo, o sistema tentará localizar o arquivo DTD na pasta raiz onde o processo Java está sendo executado.

³ Válido a partir de fevereiro de 2023

5.6.1. Como indicar uma versão alternativa de DTD para a validação

Para poder validar arquivos no formato ST.26 da OMPI que referenciam uma versão mais antiga do DTD ST.26, este arquivo DTD ST.26 deve ser posto à disposição do Validator para possibilitar a validação apropriada.

Para isto, há dois caminhos alternativos:

- Descompactar o arquivo JAR e incluir uma referência ao arquivo DTD ST.26 adicional ou alternativo na pasta "scr/main/resources".
- Modificar o arquivo "catalog.xml" e adicionar uma nova entrada para o arquivo DTD ST.26 adicional ou editar o arquivo existente.

Por exemplo:

```
<?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>
```

Em vez de modificar o arquivo JAR, é preciso seguir as etapas seguintes:

- a. Copiar "catalog.xml" e todos os DTDs em uma pasta local;
- b. Modificar "catalog.xml" de modo que inclua uma referência ao DTD ST.26 adicional; e
- c. Definir esta propriedade Java do sistema durante o lançamento:
"xml.catalog.files=<path_to_catalog.xml>"

Nota: Em Tomcat para Windows, isso é possível adicionando esta variável de ambiente:

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

[IMPORTANTE: A inclusão de uma versão diferente do DTD ST.26 permitirá a validação "formal" do arquivo XML com relação ao DTD ST.26 referenciado, mas a validação "completa" provavelmente exigirá uma modificação do código fonte para implementar as regras de verificação. Portanto, recomenda-se o uso de múltiplos DTDs somente quando realizando uma validação "formal".]

6. API REST do Validator

Esta seção especifica os casos de uso da API Validator. Existem três serviços ou casos de uso:

- Validar um arquivo na pasta de entrada ("Inbox"); (seção 6.1)
- Validar um arquivo como parte da solicitação (seção 6.1); e
- Solicitar o status de uma validação (seção 6.2).

A especificação para a API para este serviço (inscrita em OAS 3.0 [arquivo YAML]) é fornecida na íntegra no Anexo II.

6.1. Validação de um arquivo no formato ST.26 da OMPI

Mapeamento da solicitação	/api/v1/validate
Método	POST
Consome	application/json
Produz	application/json
Operação	Solicita a validação de um arquivo existente no formato ST.26 da OMPI na pasta de caixa de entrada ("Inbox"). Retorna um código ("verificationID") exclusivo para recuperar o status da solicitação de validação.
Solicitação	<pre>{ "currentApplicationNumber": "string", "currentSQLVersionNumber": "string", "parentApplicationNumber": "string", "parentSQLVersionNumber": "string", "sqlInputLocation": "string" (Location of Input.xml file), "verificationReportOutputPath": "string" (Destino do arquivo report.xml), "nameFile": "file.xml", "type": "string" (Possible values: full formality), }</pre>
Respostas	<ul style="list-style-type: none"> • '202': "Accepted" (aceito). A validação formal do arquivo no formato ST.26 da OMPI foi bem-sucedida e a verificação da conformidade com as regras de negócios foi iniciada. Este código também incluirá uma mensagem de resposta indicando um código único para recuperar o relatório de verificação ("verificationID"). O arquivo no formato ST.26 da OMPI é movido para a pasta "Process" para processamento. • '400': "Bad request" (solicitação errada). A solicitação REST não foi bem formulada ou a validação XML do arquivo no formato ST.26 da OMPI não foi bem-sucedida. Este código será

	<p>acompanhado de uma mensagem de resposta indicando os detalhes do erro</p> <ul style="list-style-type: none"> • '400': "Bad request" - "FILENAME_NOT_VALID". O nome do arquivo inclui caracteres inválidos. Embora o nome do arquivo possa conter caracteres não latinos, alguns sinais de pontuação são considerados inválidos • '404': "Not Found" (não encontrado) O arquivo no formato ST.26 da OMPI não foi encontrado na pasta de caixa de entrada ("Inbox") • '500': "Internal Server Error" (erro interno do servidor). Ocorreu um erro interno. Este código será acompanhado de uma mensagem de resposta indicando os detalhes do erro
Condição prévia	O arquivo no formato ST.26 da OMPI deve ser fornecido na pasta da caixa de entrada ("Inbox") especificada pelo usuário.
Condição subsequente	<p>O arquivo no formato ST.26 da OMPI é movido para a localização indicada em "verificationReportOutputPath" ou para a pasta da caixa de saída ("Outbox") na localização seguinte:</p> <p>"/[outbox]/[verificationID]/[file.xml]"</p> <p>The verification report is generated at:</p> <p>"/[reports]/ [verificationID]/report_[file.xml or file.html]"</p>

Nota: O nome de arquivo validado pode incluir caracteres não latinos mas nenhum carácter reservado será aceite.

6.2. Solicitação do status de validação

Mapeamento da solicitação	/api/v1/status
Método	POST
Consome	application/json
Produz	application/json
Operação	Solicitação do status de validação de um arquivo específico no formato ST.26 da OMPI
Solicitação	<pre>{ verificationID: {type: string} }</pre>
Respostas	<ul style="list-style-type: none"> • '200': Êxito. Este código também incluirá uma mensagem de resposta indicando um ID único com o status do processo de verificação, selecionado entre um dos seguintes: <ul style="list-style-type: none"> ○ "Running" (em execução): o arquivo está sendo processado ○ "FINISHED-VALID" (concluído-válido): a validação formal do arquivo foi bem-sucedida e o resultado da validação está disponível na pasta de relatórios ○ "FINISHED-INVALID" (concluído-inválido): o processo foi concluído mas a validação formal do arquivo não foi bem-sucedida. O resultado da validação está disponível na pasta de relatórios. ○ "NOT_FOUND" (não encontrado): o "verificationID" não foi encontrado ○ "VERIFICATION_ID_ERROR" (erro no ID de verificação): o "verification ID" não foi incluído na solicitação • '400': "Bad request" (solicitação errada). A solicitação REST não foi bem formulada • '500': Erro de servidor. Ocorreu um erro interno. Este código também incluirá uma mensagem de resposta indicando os detalhes do erro
Condição subsequente	O Validator fornece o status da validação
Pressupostos	-

[Segue o Anexo I]

Anexo I: Exemplo de relatório de verificação em XML

```
<?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>
```

Esta instância de XML pode também ser baixada aqui:

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

[Segue o Anexo II]

Anexo II: Especificação completa da API (YAML)

A especificação completa da API pode ser baixada aqui:

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

[Segue o Anexo III]

Anexo III: Nomes de propriedades (JSON)

Os nomes das propriedades utilizadas na ferramenta podem ser baixados aqui:

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

Observe que isto faz parte do código-fonte e é atualizado conforme necessário juntamente com a implementação.

[Segue o Anexo IV]

Anexo IV: Exemplo de relatório de verificação em HTML

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	

[Fim do documento]