



WIPO Sequence Validator - Serviço web

Webinário de formação

O que vamos cobrir hoje

- Propósito do serviço Validator
- Comparação com a ferramenta de computador
- O que é um serviço web?
- Introdução aos pontos finais do Validator (funções principais)
- Operação e Instalação do serviço web
- Parâmetros de solicitação/resposta
- Configuração do serviço Web

WIPO Sequence Validator: propósito

- **Objetivo:** assegurar que as listagens de sequências depositadas estão conformes com a norma ST.26 da OMPI
- **Implementação:** o serviço web foi produzido para auxiliar os Institutos de PI a validar listagens de sequências depositadas
- **Institutos de PI:** incorporam em seus próprios ambientes de TI
- **Requerentes:** recebem o mesmo relatório indicando eventuais erros/avisos

WIPO Sequence Validator vs WIPO Sequence para computadores

Serviço Validator	Ferramenta para computadores
Concebido para Institutos de PI	Concebido para requerentes
Serviço web	Aplicativo para computadores
<u>Sem</u> interface	Interface
Integrado no ambiente de TI	Executado pelos usuários
Dois níveis de validação	Apenas validação completa
Mais de um DTD	Apenas o DTD mais recente

O que é um serviço web?

- É um serviço operando entre dois dispositivos eletrônicos que comunicam através da Internet ou Intranet
- Usa um formato XML/JSON padronizado de solicitação/resposta
- Não precisa ser implementado em uma linguagem específica de programação (agnosticismo de linguagem)
- Modular, dinâmico e distribuído
- Um tipo de API que opera usando SOAP ou REST como meio de comunicação
- Um contrato de serviço web define “o que”, “como” e “onde”

Especificação da API OAS

- Especificação para APIs RESTful
- REST é a sigla inglesa de "REpresentational State Transfer" (transferência representacional de estado) – arquitetura que usa um subconjunto de HTTP
- Com essa especificação, humanos e computadores podem entender as funcionalidades da API sem acessar o código-fonte
- Agnosticismo de linguagem de programação

WIPO Sequence Validator: fundamentos (1)

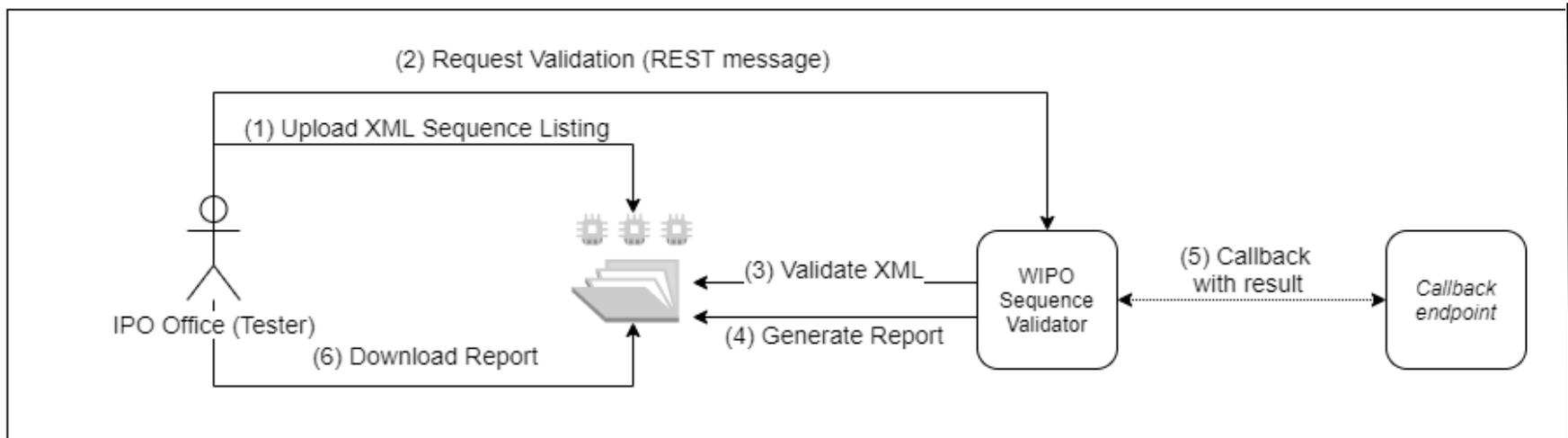
- Serve para:
 - Validar uma listagem de sequências ST.26
 - Fornecer o status de validação
 - Fazer uma chamada de retorno ao ponto final com mensagens do relatório de verificação

- Conforme com a norma ST.90 da OMPI (ver as referências)

- Especificação da API da web: OAS 2.0 (ver as referências)

- Não foi concebido para ser exposto externamente

WIPO Sequence Validator: fundamentos (2)



Instalação do Validator

- Requer Java 8 instalado (será futuramente compatível com Java 11)
- A estrutura de pastas deve ser estabelecida na localização principal do serviço web
- A configuração padrão de base pode ser adaptada com seu próprio arquivo `application.properties`
- Conjunto completo de instruções disponível no Manual de Utilização do WIPO Sequence Validator:
https://www.wipo.int/export/sites/www/standards/pt/sequence/wipo_sequence_validator_operations_manual.pdf

Estrutura padrão das pastas

temp/st26

temp/st26/inbox

temp/st26/outbox

temp/st26/params

temp/st26/process

temp/st26/reports

Implantação do serviço web

- Dois tipos de serviço disponíveis: WAR e JAR
 - JAR – Servidor integrado
 - WAR – Os Institutos de PI precisarão instalar um servidor Tomcat, ou qualquer servidor de aplicativos compatível com Spring Boot 2 e Servlet Spec 3.1+

- A interface de usuário Swagger pode ser usada para fornecer parâmetros básicos/realizar testes em: [http://\[host-name\]:8080/swagger-ui.html](http://[host-name]:8080/swagger-ui.html)

- **Os Institutos de PI devem fornecer seus pontos finais de chamada de retorno, pois não fazem parte da ferramenta**

Implantação do serviço como um JAR

■ `java -D"file.encoding=UTF-8" -jar wipo-sequence-validator.jar`

■ A porta 8080 está predefinida mas a porta do servidor pode ser alterada usando:

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



```
Command Prompt - java -D"file.encoding=UTF-8" -jar wipo-sequence-validator.jar
5 Dir(s) 121,863,659,520 bytes free

C:\dev>java -D"file.encoding=UTF-8" -jar wipo-sequence-validator.jar
#####
WIPo Sequence Validator
#####
Application Version: 1.1.0-beta4-SNAPSHOT

=====
2021-04-14 09:43:14 [main] INFO c.w.st26.ipotool.IPOToolApplication - Starting IPOToolApplication v1.1.0-beta4-SNAPSHOT on LAP536593 with PID 9988 (C:\dev\wipo-sequence-validator.jar started by DevFrancis in C:\dev)
2021-04-14 09:43:14 [main] INFO c.w.st26.ipotool.IPOToolApplication - No active profile set, falling back to default profiles: default
2021-04-14 09:43:14 [main] INFO o.s.b.w.s.c.AnnotationConfigServletWebServerApplicationContext - Refreshing org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApplicationContext@de0a01f: startup date [Wed Apr 14 09:43:14 CEST 2021]; root of context hierarchy
2021-04-14 09:43:16 [main] INFO o.s.c.s.PostProcessorRegistrationDelegate$BeanPostProcessorChecker - Bean 'asyncConfiguration' of type [com.wipo.st26.ipotool.services.async.AsyncConfiguration$$EnhancerBySpringCGLIB$$c683b031] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for
```

Implantação do serviço como um WAR

- Para Tomcat 8.5:
 - Parar o servidor: “\$TOMCAT_ROOT\bin\catalina.bat stop”
 - Copiar WAR para “\$TOMCAT_ROOT\webapps\wipo-sequence-validator.war”
 - Iniciar o servidor: “\$TOMCAT_ROOT\bin\catalina.bat start”
- “\$TOMCAT_ROOT” se refere à pasta raiz do servidor Tomcat de aplicativos
- Quando o servidor de aplicativos é lançado, é preciso usar a codificação UTF-8
- 8080 é de novo a porta padrão. Para alterá-la, é preciso definir o atributo "port" no arquivo de configuração de Tomcat. Referência: https://tomcat.apache.org/tomcat-8.5-doc/config/http.html#Common_Attributes

A função de validação

- **Descrição:** Solicita a validação de um arquivo ST.26 existente na pasta de caixa de entrada. Retorna um verificationID exclusivo para recuperar mediante solicitação o status da solicitação de validação
- **Endpoint (JAR):** [http://\[host-name\]:8080/api/\[version\]/validate](http://[host-name]:8080/api/[version]/validate)
- **Endpoint (WAR):** [http://\[host-name\]:8080/wipo-sequence-validator/api/\[version\]/validate](http://[host-name]:8080/wipo-sequence-validator/api/[version]/validate)
- **Produz:** application/json
- **Entrada:** nameFile e tipo de validação
- **Saída:** verificationID e relatório de verificação

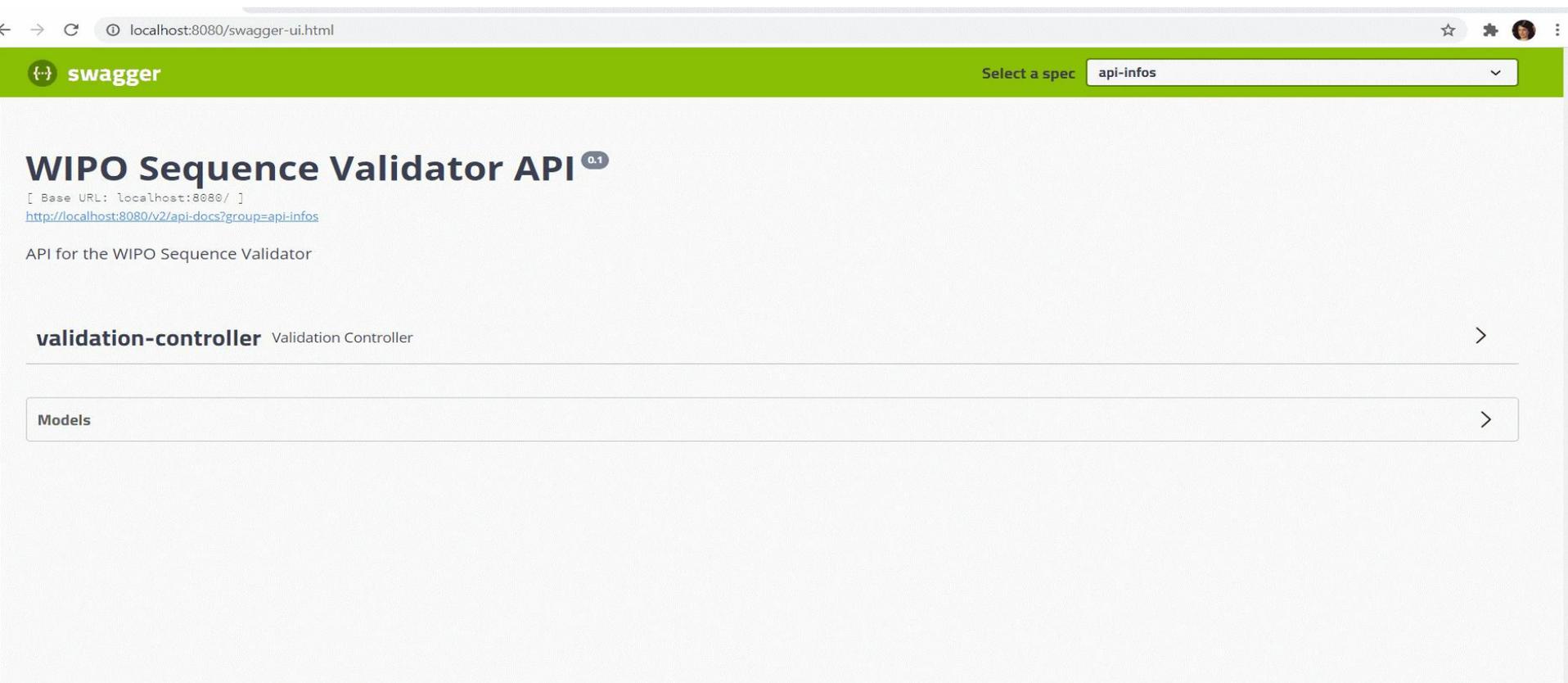
Validação "completa" versus "formal"

- O Validator pode executar dois tipos de validação: *completa* e *formal*
 - **Formal:** verifica se o arquivo XML é válido e conforme com o ST.26 DTD (graus de severidade: XML_WARN | XML_ERROR)
 - **Completa:** regras de validação formal E de verificação da conformidade derivadas da norma ST.26 da OMPI (graus de severidade: ERRO | AVISO)
- A validação formal pode ser realizada de forma síncrona pelos usuários durante o depósito. A validação completa deve ser executada de forma assíncrona

Depois da validação

- O arquivo da listagem de sequências é movido para a pasta local de caixa de saída ("Outbox")
- Um relatório de verificação é gerado na pasta local de relatórios ("Reports")
- As pastas temporárias na pasta local de processamento ("Process") são excluídas
- Um ID de verificação identifica o processo específico de verificação e a pasta criada é rotulada com este ID
- Chamada opcional de retorno para o ponto final para fornecer a resposta incluindo o relatório de verificação (se assim configurado pelos Institutos de PI)

Interface Swagger de Usuário (demonstração)



The screenshot shows a web browser window displaying the Swagger UI for the WIPO Sequence Validator API. The browser's address bar shows the URL `localhost:8080/swagger-ui.html`. The Swagger logo is visible in the top left corner of the interface. A dropdown menu in the top right corner is set to "api-infos". The main content area displays the API title "WIPO Sequence Validator API" with a version indicator "0.1". Below the title, the base URL is shown as `localhost:8080/` and a specific API endpoint is provided: `http://localhost:8080/v2/api-docs?group=api-infos`. A description states "API for the WIPO Sequence Validator". A section for "validation-controller" (Validation Controller) is visible, along with a "Models" section, both with expandable arrows.

[hostname]:8080/swagger-ui.html

Função (Verificação) de status

- **Descrição:** Solicita o status de validação de um arquivo ST.26 específico
- **Ponto final (JAR):** [http://\[host-name\]:8080/api/\[version\]/status](http://[host-name]:8080/api/[version]/status)
- **Ponto final (WAR):** [http://\[host-name\]:8080/wipo-sequence-validator/api/\[version\]/status](http://[host-name]:8080/wipo-sequence-validator/api/[version]/status)
- **Produz:** application/json
- **Entrada:** verificationID
- **Saída:** status
(RUNNING/FINISHED_VALID/FINISHED_INVALID/NOT_FOUND/VERIFICATION_ID_ERROR)

Ponto final da chamada de retorno

- Ponto final da chamada de retorno: uma API separada que recebe as informações do seu serviço web
- A validação também pode ser executada por uma chamada de retorno feita pelo ponto final para o serviço web
- Será realizada de forma assíncrona
- Deve cumprir com o contrato de serviço web fornecido no manual de utilização
- A solicitação e a resposta devem estar no formato padrão determinado

Formato da solicitação

■ Função de validação

```
{  
  "currentApplicationNumber": "string",  
  "currentSQLVersionNumber": "string",  
  "parentApplicationNumber": "string",  
  "parentSQLVersionNumber": "string",  
  "sqlInputLocation": "string",  
  "verificationReportOutputPath": "string",  
  "nameFile": "file.xml",  
  "type": "full or formality"  
}
```

■ Função de status

```
{  
  "verificationID": "1552208288697FNc2"  
}
```

Resposta fornecida (1)

- **processID:** já fornecida por WIPO Sequence Validator
- **seqType:** tem um valor fixo: "ST.26"
- **httpStatus:** Código do status HTTP fornecido pelo serviço: "Bem-sucedido"/"Mal-sucedido", junto com o código de erro HTTP e a descrição correspondente
- **applicationNumber:** do pedido de patente do qual a listagem de sequências faz parte, por exemplo PCTUS1234567 ou 23456789
- **currentSEQLVersionNumber:** o número da versão desta listagem de sequências (atribuído internamente por um Instituto de PI, por exemplo 1.1)
- **parent ApplicationNumber:** qualquer pedido de origem associado, por exemplo 12345678
- **parent SEQLVersionNumber:** o número da modificação da listagem de sequências de origem (atribuído internamente por um Instituto de PI, por exemplo 3.2)
- **verificationReportOutputPath:** fornece o caminho completo para o relatório
- **startTime:** o momento quando o serviço iniciou a validação, por exemplo 2019-07-29 15:59:37.784
- **endTime:** o momento quando o serviço concluiu a validação, por exemplo 2019-07-29 15:59:37.859
- **elapsedTime:** indica o tempo total necessário para validar a listagem de sequências, por exemplo 0 h 0 min 0 seg 75 ms

Resposta fornecida (2)

- **totalWarningQuantity**: soma de todos os avisos no relatório de verificação, por exemplo 2
- **totalErrorQuantity**: soma de todos os erros indicados no relatório de verificação, por exemplo 0
- **seqInputQuantity**: número total de sequências na seção de informações gerais do projeto informado pelo requerente, por exemplo 10
- **seqIDQuantity**: etapa de verificação do número de sequências na listagem de sequências em relação ao número de sequências informadas pelo requerente, por exemplo 10
- **errorSummary** resumo dos erros relatados no relatório de verificação, com os seguintes detalhes:

```
[
{errorIndexID (esta é a numeração sucessiva de cada código de erro
relatado, por exemplo 1, 2, 3, etc.) , errorCode: (identificação
numérica ou alfanumérica do erro, por exemplo 1(####) para erros, ou
2(####) para avisos) , descrição (par código de erro-descrição),
sequenceNumber (SEQ ID), xpath: (pode ser usado para navegar pelos
elementos e atributos em um documento XML , por exemplo /
ST26SequenceListing/SequenceData[1]/INSDSeq/INSDSeq_sequence),
characterRange: (por exemplo 10-20, das posições de caracteres 10 a 20
na INSDSeq_sequence do elemento)
} , {(mesmas informações relacionadas para o próximo erro) },
```

Relatório de verificação

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<verificationReport productionDate="YYYY-MM-DD" sourceFileName="[ST.26 filename]">
  <verificationMessages>
    <message>
      <severity>[ERROR | WARN | XML_WARN | XML_ERROR]</severity>
      <dataElement>[ST.26 element]</dataElement>
      <detectedSequence>[Sequence ID]</detectedSequence>
      <detectedValue>[value]</detectedValue>
      <messageKey>[Message key]</messageKey>
      <params>
        | <param key="param key">Param value</param>
      </params>
      <localizedMessage> [Localized message] </localizedMessage>
    </message>
    ...
  </verificationMessages>
</verificationReport>

```

Relatório de verificação: Exemplo

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<VerificationReport productionDate="2021-05-10" sourceFileName="EPOShort.xml">
  <VerificationMessageBag>
    <VerificationMessage>
      <Severity>WARNING</Severity>
      <DataElement>PROPERTY_NAMES.FEATURE_QUALS</DataElement>
      <DetectedSequence>2</DetectedSequence>
      <DetectedValue>2'-O-methyladenosine</DetectedValue>
      <MessageKey>X_FEATURE_QUALS_QUAL_NOTE_VAL_OTHER</MessageKey>
      <ParameterBag/>
      <LocalizedMessage>Verify that the 'note' qualifier describes the 'OTHER' value for the 'mod_base' qualifier.</LocalizedMessage>
    </VerificationMessage>
    <VerificationMessage>
      <Severity>WARNING</Severity>
      <DataElement>PROPERTY_NAMES.FEATURE_QUALS</DataElement>
      <DetectedSequence>6</DetectedSequence>
      <DetectedValue>2'-O-methyladenosine</DetectedValue>
      <MessageKey>X_FEATURE_QUALS_QUAL_NOTE_VAL_OTHER</MessageKey>
      <ParameterBag/>
      <LocalizedMessage>Verify that the 'note' qualifier describes the 'OTHER' value for the 'mod_base' qualifier.</LocalizedMessage>
    </VerificationMessage>
    <VerificationMessage>
      <Severity>ERROR</Severity>
      <DataElement>PROPERTY_NAMES.APPLICANT</DataElement>
      <DetectedSequence></DetectedSequence>
      <DetectedValue>{"name":"DAIICHI SANKYO COMPANY, LTD.", "languageCode":""}</DetectedValue>
      <MessageKey>LANGUAGE_CODE_MISSING</MessageKey>
      <ParameterBag/>
      <LocalizedMessage>The language code has not been entered.</LocalizedMessage>
    </VerificationMessage>
  </VerificationMessageBag>
</VerificationReport>

```

Configuração padrão dos parâmetros

- Caminho de base a ser usado pelas outras pastas=**../temp/st26/**
- Pasta onde colocar os arquivos a serem processados=Base Path+**/inbox/**
- Pasta onde armazenar os arquivos ST26 depois de validados=Base Path+**/outbox/**
- Pasta onde armazenar o relatórios de validação=Base Path+**/reports/**
- Pasta principal para pastas de validação completa e formal=Base Path+**/process/**
- Pasta onde armazenar os parâmetros=Base Path+**/params/**
- AlternativeResourceBasePath=Base Path+**/alt_resources**
- Localização usada para as mensagens localizadas no relatório de verificação **en**
- URL do ponto final de chamada de retorno que será usado para informar sobre os resultados da validação=**http://callbackservice/api/endpoint**
- **Também para registrar, etc.**

Configuração alternativa (1)

- A configuração pode ser realizada com um novo arquivo `application.properties`

- Se um arquivo `application.properties` alternativo for fornecido, a ferramenta procurará uma alternativa na ordem seguinte:
 - Uma pasta `"/config"` dentro do diretório atual
 - O diretório atual
 - Um pacote `$classpath` ou `config`; e então
 - A raiz `$classpath` OU
 - No local especificado usando um parâmetro na linha de comando

- **Nota: O Validator deverá ser reiniciado antes que a nova configuração seja aplicada**

Configuração alternativa (2)

- Em particular, é possível realizar as duas personalizações seguintes:
 - Língua das mensagens do relatório de verificação – defina com o código da língua apropriada o parâmetro "validator_locale" do arquivo application.properties, por exemplo "validator_locale=es" para alterar a língua para o espanhol.
 - Nomes de organismos personalizados: forneça um só arquivo JSON de nomes de organismos que não fazem parte da lista empacotada como:

```
[  
  {"value": "Custom Organism Sample"},  
  {"value": "Custom Organism Sample 2"}  
]
```
 - O novo arquivo custom_organism.json deve estar localizado no local indicado pelo caminho alternativeResourceBasePath.

Configuração alternativa: DTD (1)

- Por predefinição, o serviço Validator validará com o DTD mais recente: disponível em `"/src/main/resources"` (versão atual 1.3)

- Também é possível validar com um DTD mais antigo usando dois métodos diferentes:
 - Primeiro:
 - Descompactar o arquivo JAR e incluir uma referência ao arquivo DTD ST.26 adicional ou alternativo na pasta `"/src/main/resources"`;
 - Modificar o arquivo `"catalog.xml"` adicionando uma nova entrada para o DTD ST.26 adicional ou editar o existente.

Configuração alternativa: DTD (2)

- Em segundo lugar, em vez de modificar o arquivo JAR:
 - Copiar "catalog.xml" e todos os DTDs em uma pasta local;
 - Modificar "catalog.xml" de modo que inclua uma referência ao DTD ST.26 adicional; e
 - Definir esta propriedade Java do sistema durante o lançamento:
"xml.catalog.files=<path_to_catalog.xml>"

■ **Nota: Este método só funcionará para a verificação formal já que podem existir atualizações necessárias das regras de verificação de negócios derivadas do conteúdo da Norma**

Sessão de perguntas e respostas

standards@wipo.int

Referências

- Página inicial de WIPO Sequence (binários do Validator e manual de utilização em português):

<https://www.wipo.int/standards/pt/sequence>

- Norma ST.26 da OMPI (em inglês):

<https://www.wipo.int/export/sites/www/standards/en/pdf/03-26-01.pdf>

- Perguntas frequentes sobre a implementação da norma ST.26 da OMPI (em inglês):

<https://www.wipo.int/standards/en/sequence/faq.html>

- Norma ST.90 da OMPI (em inglês):

<https://www.wipo.int/export/sites/www/standards/en/pdf/03-90-01.pdf>

- Especificação de OAS 2.0 (em inglês):

<https://swagger.io/specification/v2/>