



WIPO

WORLD
INTELLECTUAL PROPERTY
ORGANIZATION

WIPO Sequence Validator Webdienst

Webinar-Schulung

Unsere heutigen Themen

- Zweck des Validator-Dienstes
- Vergleich mit dem Desktop-Tool
- Was ist ein Webdienst?
- Einführung in die Validator-Endpunkte (Hauptfunktionen)
- Betrieb und Installation des Webdienstes
- Anfrage-/Antwortparameter
- Konfiguration des Webdienstes

WIPO Sequence Validator: Zweck

- **Ziel:** Sicherstellen, dass die eingereichten Sequenzprotokolle mit WIPO ST.26 konform sind
- **Implementierung:** ein für IP-Ämter erstellter Webdienst zur Validierung eingereicherter Sequenzprotokolle
- **IP-Ämter:** Integration in deren eigene IT-Umgebung
- **Anmelder:** erhalten denselben Bericht mit Angabe aller Fehler/Warnungen

WIPO Sequence Validator vs WIPO Sequence Desktop

Validator-Dienst	Desktop-Tool
Für die Ämter konzipiert	Für die Anmelder konzipiert
Webdienst	Desktop-Anwendung
<u>Keine</u> Schnittstelle	Schnittstelle
In die IT-Umgebung integriert	Betrieb durch den Nutzer
Zwei Validierungsebenen	Nur vollständige Validierung
Mehr als eine DTD	Nur neueste DTD

Was ist ein Webservice?

- Ein Dienst zwischen zwei elektronischen Geräten, die über das Internet oder Intranet miteinander kommunizieren
- Verwendet ein standardisiertes XML/JSON-Anfrage/Antwortformat
- Muss nicht in einer bestimmten Programmiersprache implementiert werden (sprachunabhängig)
- Modular, dynamisch und verteilt
- Ein API-Typ, der SOAP oder REST als Kommunikationsmittel verwendet
- Ein Webservice-Vertrag definiert das "Was"/"Wie" und "Wo"

OAS-API-Spezifikation

- Spezifikation für RESTful APIs
- REST steht für REpresentational State Transfer – eine Architektur, die eine Untermenge von HTTP verwendet
- Mit dieser Spezifikation können Mensch und Computer die Fähigkeiten der API verstehen, ohne auf den Source-Code zuzugreifen.
- Programmiersprache ist sprachunabhängig

WIPO Sequence Validator: Grundlagen (1)

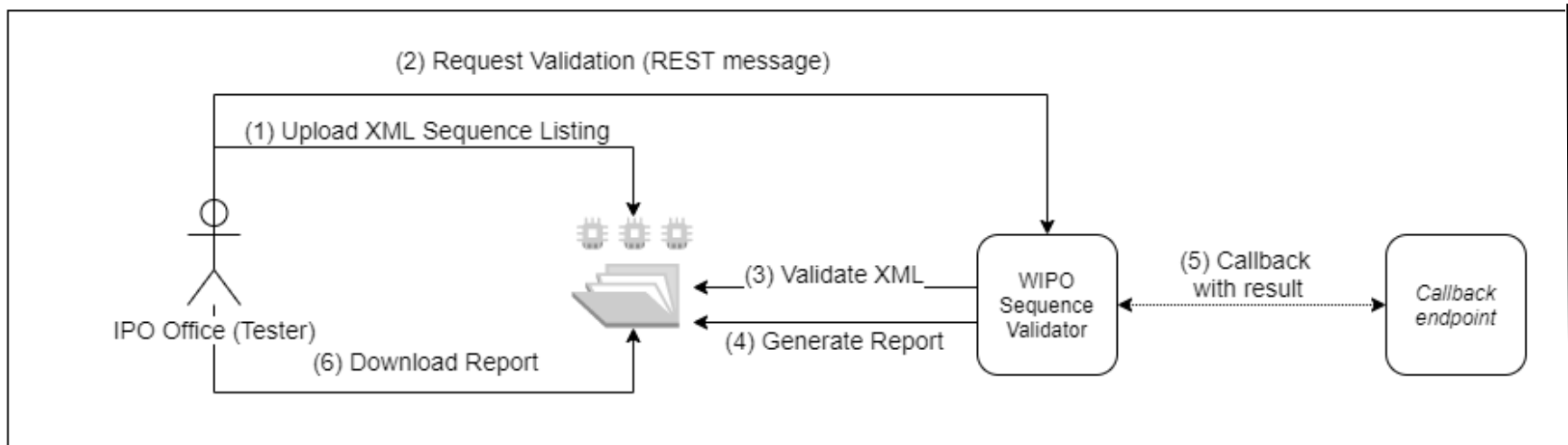
- Funktionen:
 - Validierung eines ST.26-Sequenzprotokolls
 - Bereitstellen des Status der Validierung
 - Callback zum Endpunkt mit Meldungen aus dem Prüfbericht

- Konform mit WIPO ST.90 (siehe Verweise)

- Web-API-Spezifikation: OAS 2.0 (siehe Verweise)

- Wurde nicht für den externen Gebrauch konzipiert

WIPO Sequence Validator: Grundlagen (2)



Installation des Validators

- Erfordert die Installation von Java 8 (Java 11 wird in Zukunft unterstützt)
- Die Ordnerstruktur muss an einem dem Standort des Webdienstes übergeordneten Standort eingerichtet werden
- Grundlegende Standardkonfiguration kann durch Bereitstellung einer eigenen application.properties-Datei angepasst werden
- Die vollständigen Anweisungen sind in der Bedienungsanleitung des WIPO Sequence Validators enthalten:
https://www.wipo.int/export/sites/www/standards/en/sequence/wipo_sequence_validator_operations_manual.pdf

Standard-Ordnerstruktur

temp/st26

temp/st26/inbox

temp/st26/outbox

temp/st26/params

temp/st26/process

temp/st26/reports

Bereitstellung des Webdienstes

- Zwei Arten von Diensten werden bereitgestellt: WAR und JAR
 - JAR – integrierter Server
 - WAR – die Ämter benötigen einen Tomcat-Server oder einen beliebigen Anwendungsserver, der mit installiertem Spring Boot 2 und Servlet Spec 3.1+ kompatibel ist

- Swagger UI kann verwendet werden, um grundlegende Parameter bereitzustellen/Tests auszuführen unter: [http://\[host-name\]:8080/swagger-ui.html](http://[host-name]:8080/swagger-ui.html)

- **Ämter müssen ihren Callback-Endpunkt selbst bereitstellen, da dieser nicht Teil des Tools ist**

Einsatz des Dienstes als JAR

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

■ Der Standardport ist 8080, doch der Serverport kann geändert werden mit:

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

```

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$$EnhancerBySpringGLIB$$c683b031] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for

```

Einsatz des Dienstes als WAR

- Für Tomcat 8.5:
 - Server stoppen: "\$TOMCAT_ROOT\bin\catalina.bat stop"
 - WAR nach "\$TOMCAT_ROOT\webapps\wipo-sequence-validator.war" kopieren
 - Server starten: "\$TOMCAT_ROOT\bin\catalina.bat start"

- "\$TOMCAT_ROOT" bezieht sich auf den Stammordner des Tomcat-Anwendungsservers

- Beim Starten des Anwendungsservers muss die Dateikodierung auf UTF-8 gesetzt werden

- Standard-Port ist wieder 8080. Um dies zu ändern, muss das Attribut "port" in der Tomcat-Konfigurationsdatei gesetzt werden. Verweis:
https://tomcat.apache.org/tomcat-8.5-doc/config/http.html#Common_Attributes

Validierungsfunktion

- **Beschreibung:** Fordern Sie die Validierung einer vorhandenen ST.26-Datei im "Inbox"-Ordner an. Sie erhalten eine eindeutige "verificationID" zum Abrufen des Status der Validierungsanforderung
- **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)
- **Generiert:** application/json
- **Eingabe:** namefile und Art der Validierung
- **Ausgabe:** verificationID und Prüfbericht

"Vollständige Validierung" versus "Formalitäts"-Validierung

- Es gibt zwei Arten der Validierung, die der Validator durchführen kann: *vollständige Validierung* und *Formalitäts-Validierung*
 - **Formalitäts-Validierung:** Ist die XML-Datei gültig und mit ST.26 DTD konform (severity: XML_WARN | XML_ERROR)
 - **Vollständige Validierung:** Formalitäts-Validierung UND Konformitätsprüfregeln abgeleitet von WIPO ST.26 (severity: ERROR | WARN)
- Die Formalitäts-Validierung kann den Nutzern bei der Einreichung synchron zur Verfügung gestellt werden. Die vollständige Validierung sollte asynchron durchgeführt werden

Nach der Validierung

- Die Sequenzprotokolldatei wird in den lokalen "Outbox"-Ordner bewegt
- Der Prüfbericht wird im lokalen "Reports"-Ordner erzeugt
- Temporäre Ordner im lokalen "Process"-Ordner werden gelöscht
- Es gibt eine "verification ID", die den jeweiligen Prüfprozess identifiziert, und der erstellte Ordner wird mit dieser ID gekennzeichnet
- Wahlweiser Callback an den Endpunkt zur Bereitstellung einer Antwort, die den Prüfbericht enthält (falls von den IP-Ämtern konfiguriert)

Swagger UI (Demo)

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 UI header is green and contains the Swagger logo on the left and a dropdown menu labeled "Select a spec" with "api-infos" selected. The main content area features the title "WIPO Sequence Validator API" with a version indicator "0.1". Below the title, it shows the base URL `localhost:8080/` and a link to the API documentation. The description states "API for the WIPO Sequence Validator". A section titled "validation-controller" is visible, with a right-pointing arrow. Below this, a "Models" section is also visible with a right-pointing arrow.

[hostname]:8080/swagger-ui.html

(Prüfen) Statusfunktion

- **Beschreibung:** Anfrage des Status der Validierung einer bestimmten ST.26-Datei
- **Endpoint (JAR):** [http://\[host-name\]:8080/api/\[version\]/status](http://[host-name]:8080/api/[version]/status)
- **Endpoint (WAR):** [http://\[host-name\]:8080/wipo-sequence-validator/api/\[version\]/status](http://[host-name]:8080/wipo-sequence-validator/api/[version]/status)
- **Generiert:** application/json
- **Eingabe:** verificationID
- **Ausgabe:** status
(RUNNING/FINISHED_VALID/FINISHED_INVALID/NOT_FOUND/VERIFICATION_ID_ERROR)

Callback-Endpunkt

- Callback-Endpunkt: eine separate API, die die Informationen von Ihrem Webdienst erhält
- Die Validierung kann auch durch den Callback-Endpunkt durchgeführt werden, der den Webdienst aufruft
- Wird asynchron durchgeführt
- Muss dem in der Bedienungsanleitung angegebenen Webdienst-Vertrag entsprechen
- Anfrage- und Antwortformate müssen ebenfalls in dem definierten Standardformat bereitgestellt werden

Anfrageformat

■ Validierungsfunktion

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

■ Statusfunktion

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

In der Antwort enthaltene Angaben (1)

- **processID**: bereits durch den WIPO Sequence Validator bereitgestellt
- **seqType**: ein fester Wert: 'ST.26'
- **httpStatus**: HTTP-Statuscode vom Dienst: 'Success' / 'Failure'
zusammen mit dem HTTP-Fehlercode und entsprechender Beschreibung
- **applicationNumber**: für die Patentanmeldung, zu der das
Sequenzprotokoll gehört, z.B. PCTUS1234567 oder 23456789.
- **currentSQLVersionNumber**: die Versionsnummer dieses
Sequenzprotokolls (intern von einem Amt zugewiesen, z.B. 1.1.)
- **parentApplicationNumber**: eine etwaige zugehörige Hauptanmeldung,
z.B. 12345678.
- **parentSQLVersionNumber**: die Änderungsnummer des Sequenzprotokolls
der Hauptanmeldung (intern von einem Amt zugewiesen, z.B. 3.2.)
- **verificationReportOutputPath**: gibt den vollständigen Pfad für den
Bericht an
- **startTime**: Zeitpunkt, zu dem der Dienst mit der Validierung begann,
z.B. 2019-07-29 15:59:37.784
- **endTime**: Zeitpunkt, zu dem der Dienst die Validierung abgeschlossen
hat, z.B. 2019-07-29 15:59:37.859
- **elapsedTime**: die für die Validierung des Sequenzprotokolls
insgesamt benötigte Zeit, z.B. 0 hr(s) 0 min(s) 0 sec(s) 75 ms

...

In der Antwort enthaltene Angaben (2)

- **totalWarningQuantity**: Summe aller Warnungen aus dem Prüfbericht, z.B. 2.
- **totalErrorQuantity**: Summe aller Fehler aus dem Prüfbericht, z.B. 0.
- **seqInputQuantity**: Gesamtzahl der Sequenzen aus dem Abschnitt Allgemeine Informationen im vom Anmelder eingegebenen Projekt, z.B. 10.
- **seqIDQuantity**: Prüfschritt zum Vergleich der Anzahl der Sequenzen im Sequenzprotokoll mit der vom Anmelder eingegebenen Anzahl, z.B. 10.
- **errorSummary**: Zusammenfassung der im Prüfbericht gemeldeten Fehler, mit den folgenden Angaben:

```
[
{errorIndexID (dies ist the fortlaufende Nummerierung jedes gemeldeten
Fehlercodes, z.B. 1, 2, 3, etc.) : , errorCode: (numerische oder
alphanumerische Kennzeichnung des Fehlers, z.B. 1(####) für Fehler, oder
2(####) für Warnungen) , description (Fehlercode-Beschreibung-Paarung)
:, sequenceNumber (SEQ ID), xPath: (kann verwendet werden, um durch
Elemente und Attribute in einem XML -Dokument zu navigieren, z.B. /
ST26SequenceListing/SequenceData[1]/INSDSeq/INSDSeq_sequence) ,
characterRange: (z.B. 10-20 ab Zeichenposition 10 bis Position 20 im
Element INSDSeq_sequence)
} , { (dieselbe Information wird für den nächsten Fehler gemeldet) }, {
... .. }
```

Prüfbericht

```

<?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>

```

Prüfbericht: Beispiel

```

<?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>

```


Konfiguration der Parameter - Standard

- Basispfad, der von den restlichen Ordnern verwendet werden soll=**../temp/st26/**
- Ordner, in dem die zu verarbeitenden Dateien abgelegt werden=**Base Path+/inbox/**
- Ordner zum Speichern der ST.26-Dateien nach Validierung=**Base Path+/outbox/**
- Ordner zum Speichern der Validierungsberichte=**Base Path+/reports/**
- Übergeordneter Ordner für Ordner der vollständigen und Formalitäts-Validierung=**Base Path+/process/**
- Ordner zum Speichern der Parameter=**Base Path+/params/**
- AlternativeResourceBasePath=**Base Path+/alt_resources**
- Lokalisierung für die lokalisierten Meldungen im Prüfbericht = **en**
- URL des Callback-Endpunkts, der zur Benachrichtigung über die Ergebnisse der Validierung verwendet wird=**http://callbackservice/api/endpoint**
- **Auch Logging etc.**

Alternative Konfiguration (1)

- Die Konfiguration ist durch Bereitstellung einer neuen Datei "application.properties" möglich

- Wird eine alternative Datei "application.properties" bereitgestellt, sucht das Tool nach einer Alternative in folgender Reihenfolge:
 - Ordner "/config" im aktuellen Verzeichnis
 - Aktuelles Verzeichnis
 - \$classpath oder config package; dann
 - \$classpath root ODER
 - an dem mit dem Parameter in der Befehlszeile festgelegten Ort

- **Hinweis: Der Validator muss neu gestartet werden, damit die neue Konfiguration angewendet werden kann**

Alternative Konfiguration (2)

- Insbesondere können die zwei folgenden Anpassungen vorgenommen werden:
 - Sprache der Meldung für Prüfbericht – der Parameter “`validator_locale`” der Datei `application.properties` muss auf den entsprechenden Sprachcode gesetzt werden, z.B. “`validator_locale=es`”, um die Sprache auf Spanisch zu ändern.
 - Benutzerdefinierte Organismenamen: stellen Sie eine einzelne JSON-Datei mit Organismenamen, die nicht Teil der paketierte Liste sind, bereit als:

```
[  
  {"value": "Custom Organism Sample"},  
  {"value": "Custom Organism Sample 2"}  
]
```
 - Die neue Datei `custom_organism.json` sollte sich an dem von “`alternativeResourceBasePath`” angegebenen Speicherort befinden.

Alternative Konfiguration: DTD (1)

- Standardmäßig validiert der Validator-Dienst gegen die neueste DTD-Version : bereitgestellt unter `"/src/main/resources"` (derzeit Version 1.3)

- Es ist auch möglich, anhand von zwei verschiedenen Methoden eine Validierung gegen eine ältere DTD durchzuführen:
 - Erste Methode:
 - Dekomprimieren der JAR-Datei und Hinzufügen eines Verweises auf die zusätzliche oder alternative ST.26 DTD-Datei in den Ordner `"src/main/resources"`;
 - Ändern der `"catalog.xml"`- Datei durch Hinzufügen eines neuen Eintrags für die zusätzliche ST.26 DTD oder Bearbeitung des vorhandenen Eintrags.

Alternative Konfiguration: DTD (2)

- Zweite Methode: Anstatt die JAR-Datei zu ändern:
 - Kopieren Sie "catalog.xml" und alle DTDs in einen lokalen Ordner;
 - Ändern Sie "catalog.xml", um einen Verweis auf die zusätzliche ST.26 DTD einzubeziehen; und
 - Legen Sie diese Java-Systemeigenschaft beim Start fest:
"xml.catalog.files=<path_to_catalog.xml>"

- **Hinweis: Dies funktioniert nur für die Formalitäts-Validierung, da möglicherweise Aktualisierungen der aus dem Hauptteil der Norm abgeleiteten Geschäftsprüfregeln erforderlich sind**

Q&A-Session

standards@wipo.int

Verweise

- WIPO Sequence Homepage (Validator-Binärdateien und Benutzerhandbuch):

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

- WIPO-Standard ST.26 (in Englisch):

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

- Implementierung von WIPO ST.26 FAQ (in Englisch):

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

- WIPO-Standard ST.90 (in Englisch):

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

- OAS 2.0-Spezifikation:

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