



# WIPO Sequence Validator

## 操作マニュアル

バージョン 2.3.0

*この文書は、知的財産庁をサポートして、WIPO Sequence Validator Webサービスをデプロイすること、および Validatorの設定をサポートすることを目的とします。*

WIPO FOR OFFICIAL USE ONLY

## 目次

1. はじめに.....	3
1.1. Validatorのワークフローの概要.....	3
1.2. Validatorのファイルシステム構造の定義.....	6
2. WIPO Sequence Validatorをデプロイする.....	7
2.1. Spring Boot JARとしてValidatorを起動する.....	7
2.1.1. Validatorを実行可能なアプリケーションとしてデプロイする.....	9
2.2. WARウェブサービスとしてデプロイする.....	10
3. 検証レポート.....	11
4. コールバックエンドポイントリクエスト.....	10
4.2. 検証レポート.....	16
5. 設定.....	17
5.1. デフォルト設定.....	17
5.2. Configuring verification rule VXQV_49.....	19
5.3. エンドポイントの正常性を確認する.....	19
5.4. ローカライズされたメッセージ.....	19
5.5. ユーザー生物名.....	19
5.6. ST.26 DTDファイルを参照する.....	19
5.6.1. 検証用の代替DTDバージョンを指定する方法.....	20
6. Validator REST API.....	24
6.1. WIPO ST.26ファイルを検証.....	24
6.2. 検証ステータスを要求する.....	26
Annex I：XML検証レポートの例.....	27
Annex II：完全なAPI仕様（YAML）.....	28
Annex III：プロパティ名（JSON）.....	28
Annex IV：HTML検証レポートの例.....	28

## 1. はじめに

WIPO Sequence Validator（以下「Validator」または「ツール」と呼ぶ）の主な目的は、WIPO ST.26形式のXMLファイルを検証して、WIPO Standard ST.26に準拠していることを確認するためのWebサービスを知的財産庁（IPO）に提供することです。WIPO Sequenceデスクトップアプリケーションを使用して作成された配列表はWIPO ST.26に準拠しますが、ユーザーは自身が最適と考えるツールを使用することができます。

この文書は、以下のセクションで詳述するツールのビルド、デプロイ、設定、およびファイルコンシューマーシステムについて説明することを目的としています。トラブルシューティングに関する質問は、以下のwikiを参照してください：

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

### 1.1. Validatorのワークフローの概要

このツールは、次の4つのユースケースを提供します。

- WIPO ST.26ファイルの検証
- 実行中の検証のステータスのリクエスト
- 設定ファイルの更新（IPO管理者のみ）
- 検証プロセス完了時の、検証プロセスの結果を使用したコールバックエンドポイントの呼び出し

注：コールバックエンドポイントはValidatorではカバーしていません。エンドポイントの確立は、このサービスを展開してセットアップする知的財産庁が行ってください。

このツールは、Webサービスとして実行可能なJARファイル、またはTomcatサーバーにデプロイできるWARファイルで構成されています。

いずれの場合も、WIPO ST.26配列表の検証のために、ツールはローカルファイルシステムからファイルを消費し、検証結果を含む検証レポートを作成します。また、必要に応じて、コールバックエンドポイントを呼び出して、検証プロセスの結果（検証レポート）を返します。

Validatorの主なワークフローは次の通りです。

- a) 各IPO ITシステムでは、WIPO ST.26 XMLファイルをデフォルトの[未処理]フォルダまたはリクエストにおいて指定されたフォルダに保存します。
- b) IPOシステムは、ファイルの検証を要求するHTTP Postを起動します。設定に応じて、IPOシステムはファイル

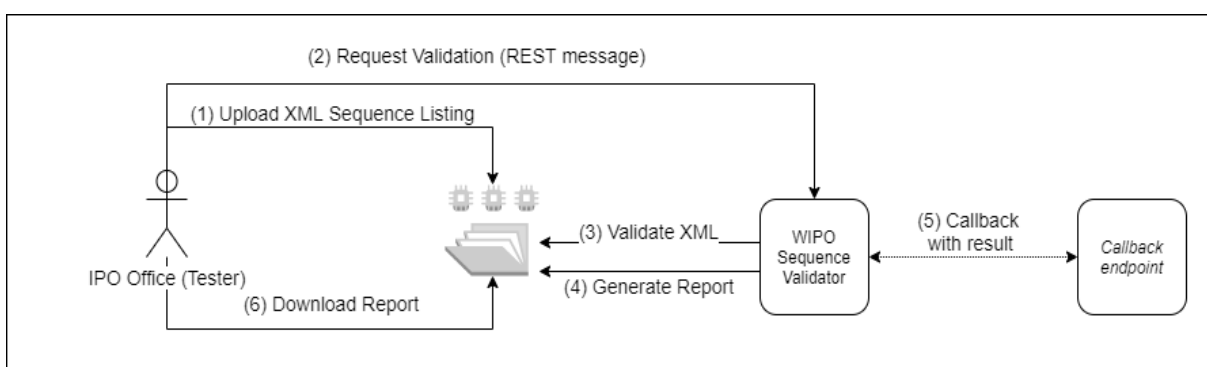
---

<sup>1</sup> ここでのコールバックエンドポイントは、リクエストのメッセージを送信可能なURIによって識別される固有のアドレスです。

の「完全」検証または「形式」検証を要求できます。「形式」検証プロセスでは、ST.26ファイルがXMLファイルであるかどうかをチェックし、ST.26 DTDに対してファイルを検証します。「完全」検証プロセスでは、ST.26の内容に基づくビジネス検証ルールに対してST.26ファイルを検証し、「形式」検証プロセスも実行します。

注：「形式」検証プロセスは同期して実行することができるため、オンライン出願受理システムには「形式」検証プロセスのみを使用することを推奨します。「完全」検証は相当な時間を要するため、バッチ処理に推奨されます。

- c) 検証が完了すると、ファイルが[形式]検証を通過したか、  
或いはIPO ITシステムが[完全]検証を選択している場合には、
- d) ビジネスルール検証プロセスが正しく開始されたか、を示すレスポンスが表示されます。
- e) Validatorは、「完全」検証を実行している場合、[未処理]フォルダからXMLファイルを取得し、ビジネスルール検証プロセスを開始して、次の処理を実行します。
- f) Validatorは、XSD schemaに基づいて、指定された[Input]フォルダにXMLレポートファイル([report\_<ファイル名>.xml])を作成し、検証済みのWIPO ST.26 XMLファイルを[処理済み]フォルダに移します。必要に応じて、Validatorはこの検証レポートのHTMLバージョンも同時に作成します。
- g) XMLレポートファイルが作成され、XSD schemaに準拠していることが確認された後、Validatorは指定された[Reports]フォルダにHTMLレポートファイルを作成します。
- h) ビジネスルール検証プロセスが完了すると、コールバックエンドポイントが（設定されている場合）Validatorから呼び出され、検証プロセスに関する追加情報がリクエストに追加されます。リクエストの構造といくつかのサンプルデータを、後述のセクション4で示しています。
- i) コールバックエンドポイントは、レスポンスに空または成功コードを返します（エラーなし）。[注：このステップは、外部Webサービスが使用可能にされており、Validatorで呼び出しが設定されている場合にのみ実行されま  
ず。]Validatorとコールバックエンドポイントとの接続も必要です。前述のように、外部WebサービスはValidatorの一部を構成するものではなく、以下に定める契約に従って知的財産庁によって展開、設定されるものです。



- j) IPOシステムは、[Reports]フォルダから検証レポートを取得できます。

注：WIPO Sequence Validatorは、Web APIを使用した知的財産データの処理および通信に関するWIPO

Standard : [WIPO ST.90](#)に準拠しています。

## 1.2. Validatorのファイルシステム構造の定義

Validatorが使用するファイルシステムの構造は、5つのフォルダで構成されています。

- **[未処理]フォルダ**：これは、IPOが検証のためにWIPO ST.26ファイルを入れるローカルフォルダです。
- **[プロセス]フォルダ**：これは、[未処理]内のファイルが処理中に一時的に通過するローカルフォルダです。このフォルダは2つのサブフォルダを含んでいます：
  - **[完全検証]フォルダ**：完全検証待ちのファイルを格納します
  - **[形式検証]フォルダ**：形式検証待ちのファイルを格納します
- **[処理済み]フォルダ**：検証が完了すると、アプリケーションはこのローカルフォルダに WIPO ST.26ファイルのソースを格納します。
- **[Reports]フォルダ**：検証結果が検証レポートファイルに保存されるローカルフォルダです。
- **[パラメータ]フォルダ**：非同期ディープ検証プロセスのパラメータを指定するために、検証リクエストのすべての検証パラメータを含むJSON（.json）ファイルが置かれているローカルフォルダです。

ファイルシステム構造の例を次に示します。

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

**[重要：デフォルトでは、/temp/ST26 ディレクトリをツールが置かれている親ディレクトリに配置する必要があります。例えば、JARまたはWARファイルがC:/dev に配置されている場合、作成されるフォルダ構造は C:/temp/ST26/... とする必要があります。]**

## 2. WIPO Sequence Validatorをデプロイする

前述のように、Validatorは以下の2つのバイナリ形式のいずれかで提供されます。知的財産庁がValidatorをデプロイするインフラストラクチャの種類に応じて、IPOはいずれか一方のバイナリの種類を他方よりも優先して選択します。

Validatorが提供される2つのバイナリは次の通りです。

- **バイナリSpringBoot JAR**：このバイナリは実行可能なJARファイルです。これには、[Java 8](#)をインストールする必要があります。
- **Warパッケージバイナリ**：このバイナリは、サーブレットコンテナにデプロイされるものです。[Tomcat 8.5](#)等のSpring Boot 2およびServlet Spec 3.1+と互換性のあるアプリケーションサーバーが必要です。

次のセクションでは、[Spring Boot](#)アプリケーションとして、またはJava アプリケーションサーバー内のWARとしてValidatorをデプロイすることについて詳述します。

### 2.1. Spring Boot JARとしてValidatorを起動する

Spring Boot JARには組み込みサーバーが含まれており、別のサーバーを必要とすることなくValidator APIをデプロイできます。これにより、インフラストラクチャレベルでの設定とデプロイが大幅に簡素化されます。

組み込みサーバーを起動するには、次のコマンドを実行する必要があります。

注：Java 8をサーバーにインストールしておく必要があります：JavaはUTF-8の使用を保証するものではないので、システムプロパティ「file.encoding」を「UTF-8」に設定する必要があります。これは、以下を含めることで実行できません。

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

Validator APIには[Swagger UI](#)を介してアクセスできます。

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

Validator APIには、次のエンドポイントでアクセスできます。

[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)

IPOは次の変更を加えてください。

- [host-name] はサーバーのホスト名に置き換えてください。
- [version] はValidator APIのバージョン（例：v1.0）に置き換えてください。

デフォルトでは、サーバーはポート8080で稼働します。ポートを変更するには、“--server.port”コマンドラインオプションを次に示すように追加する必要があります。



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

デフォルトでは、ValidatorはJava仮想マシン（JVM）のデフォルトメモリ設定を使用します。デフォルトの最大ヒープサイズは、使用可能な物理メモリの4分の1です。

最大ヒープサイズを変更するには、コマンドラインを使用して実行する際に“-Xmx”オプションを使用してください<sup>2</sup>。

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

### 2.1.1. Validatorを実行可能なアプリケーションとしてデプロイする

Validatorは、例えばオペレーティングシステムの起動時にその実行をサポートするために、オペレーティングシステムによって管理されるサービスとしてインストールすることもできます。

WIPO Sequenceがサポートするすべてのプラットフォーム（Windows、Linux、Mac OS）でこのようにSpring Boot JARファイルを設定することができます。

次のガイダンスでは、各オペレーティングシステムでJARファイルを実行するシステムサービスの作成方法について詳しく説明します。また、サービスの様々なオプションとアプリケーションの実行を設定する方法についての情報も提供します。

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

---

---

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

## 2.2. WARウェブサービスとしてデプロイする

提供されるバイナリの2番目のタイプでは、WARパッケージをApache Tomcat 8.5などの既存のJavaアプリケーションサーバーにデプロイできます。

注: サブレット 3.1 と互換性のあるコンテナが必要です。

以下の手順は、Tomcatアプリケーションサーバー用です。ここで、“\$TOMCAT\_ROOT”はTomcatサーバーのルートフォルダを指しており、この値はファイルパスに適した値に置き換える必要があります。

- a. サーバーを停止する：\$TOMCAT\_ROOT\bin\catalina.bat stop
- b. WARを \$TOMCAT\_ROOT\webapps\wipo-sequence-validator.war にコピーする
- c. サーバーを起動する：\$TOMCAT\_ROOT\bin\catalina.bat start

注：JavaはUTF-8の使用を保証するものではないため、アプリケーションサーバーの起動時にシステムプロパティ [file.encoding]を[UTF-8]に設定してください。これは、以下を含めることにより実行できます。

-D"file.encoding=UTF-8"

前述のように、Validator APIにはSwagger UIを介してアクセスできます。

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

Validator APIには、次のエンドポイントでアクセスできます。

[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)

IPOは次の変更を加えてください。

- [host-name] はサーバーのホスト名に置き換えてください。
- [version] はAPIのバージョン（例：v1.0）に置き換えてください。

デフォルトでは、サーバーはポート8080で稼働します。別のポートに変更するには、以下に記載の手順に従ってTomcat設定ファイルを変更する必要があります。

[https://tomcat.apache.org/tomcat-8.5-doc/config/http.html#Common\\_Attributes](https://tomcat.apache.org/tomcat-8.5-doc/config/http.html#Common_Attributes)

デフォルトでは、ValidatorはJVMのデフォルトのメモリ設定を使用します。デフォルトの最大ヒープサイズは、使用可能な物理メモリの4分の1です。

最大ヒープサイズを変更するには、上記のセクション2.1で示したように、コマンドラインを使用して実行するときに[-Xmx]オプションを使用してください。

### 3. 検証レポート

検証レポートを作成できる形式には、XML と HTML(またはその両方)の 2 つがあります。

このツールによって作成される検証レポートはXML形式です。使用されるテンプレートを次に示します。

```
<?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>
    ...
```

ルートレベルで示される属性は次のとおりです。

- 'applicationNumberText': この配列表に関連付けられたアプリケーション
- 'productionDate': 検証が行われた日付
- 'filingDate': 出願日
- 'softwareBuildVersion': 検証に使用されるValidatorのバージョン
- 'softwareVersion': 配列表の作成に使用されるWIPO Sequenceのバージョン
- 'sourceFileName': 配列表XMLインスタンスの名称

検証レポートの例を、このマニュアルのAnnex Iとして、Annex IIIで提供されるコンポーネントに対する使用可能な値とともに示しています。

示された重大度に関しては、以下の分類に留意してください。

- ERROR - 「完全」検証中に返されたエラー

- WARNING - 「完全」検証中に返された警告
- XML\_ERROR - 「形式」検証中に返されたエラー
- XML\_WARN - 「形式」検証中に返された警告

検証レポートは、WIPOシーケンスで使用されるのと同じスタイルシートを用いて、HTML形式でも作成されます。検証レポートの例を、このマニュアルのAnnex IVに示しています。

## 4. コールバックエンドポイントリクエスト

コールバックエンドポイントからValidatorへのリクエストは、ファイルの場所と検証プロセスの詳細を示す次のパラメータを

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

含みます。

検証リクエストの[seqInputLocation]フィールドを、検証するXML配列表のパスを示すように設定する必要があります。知的財産庁がこのフィールドを空のままにした場合、ツールは、デフォルトの[未処理]フォルダにあるファイル名[nameFile]のXMLファイルを検証しようとします。[nameFile]パラメータで、検証する配列表ファイルを識別します。

リクエスト内の[verificationReportOutputPath]は、ツールによって作成されたapplication.propertyファイルの設定に応じて、検証レポート（.xml及び／または.html）のファイルの場所を示します。ユーザーがこのフィールドを空白のままにするか、無効なファイルパスを入力した場合、検証レポートはデフォルトの[Reports]フォルダに保存されます。

### 4.1. コールバックエンドポイントリクエストの形式

プロパティ[api.URL]が設定されている場合、Validatorは、指定されたURLのエンドポイントに検証結果を送信しようとします。

Validatorと通信するには、コールバックエンドポイントが次のWebサービス契約（YAML）に準拠していなければなりません：[https://www.wipo.int/standards/en/sequence/callback\\_end\\_point\\_web\\_service\\_contract.yml](https://www.wipo.int/standards/en/sequence/callback_end_point_web_service_contract.yml)

さらに、リクエストは次の構造を持つJSONオブジェクトである必要があります。

```
{
  "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"
}
```

これは、Validatorに呼び出しを行った外部エンドポイントに送信されるJSONインスタンスの例です。

```
{
  "processID": "1608194222169dvVE",
  "seqType": "ST.26",
  "httpStatus": "SUCCESS",
  "currentApplicationNumber": "string",
  "currentSEQLVersionNumber": "string",
  "parentApplicationNumber": "string",
  "parentSEQLVersionNumber": "string",
  "verificationReportOutputPath":
  "C:/temp/report.xml", "startTime": "2020-12-
  17 09:36:54.000000",
  "endTime": "2020-12-17 09:37:26.000607",
  "elapsedTime": 32607,
  "totalWarningQuantity": 1,
  "totalErrorQuantity": 2,
  "seqInputQuantity": 0,
  "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. 検証レポート

セクション 3 で説明したように、検証後、作成された検証レポートは `verificationReportOutputPath` に置かれ、デフォルトでは、 `/temp/st26/reports/[verificationID]/report.xml` (XML レポートファイルの場合) に、あるいは `/temp/st26/reports/[verificationID]/report.html` (HTML レポートファイルの場合) に置かれます。

`application.properties` ファイルでは、HTML レポートの作成を有効または無効にすることができます。値を `[true]` に設定することでレポート作成を有効にし、値を `[false]` に設定することで無効にします。

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

このレポートの内容は、`ServiceRequest` の `errorSummary` フィールド内のコールバックエンドポイントに送信されます。このフィールドの例は、前述のセクション4に示したリクエストの例に記載されています。

## 5. 設定

### 5.1. デフォルト設定

Validatorは、プロパティファイルを使用して設定されます。デフォルトの`application.properties`ファイルは次の値を有します。<sup>3</sup>

```
##### 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
```

<sup>3</sup> 2023年2月より有効

```
app.property.html=true
#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 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
```

ここで示したパラメータの値を変更するには、代替の[application.properties]ファイルを使用してください。Spring Bootのドキュメンテーションに詳述されているように、いくつかのオプションがあります：<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>

最もシンプルなオプションは、新しいapplication.propertiesファイルを指定することです。このファイルは、操作の順に、次の場所で検索されます：

- a) 現在のディレクトリ内の[/config]フォルダ[注：ValidatorがWARファイルとしてTomcatにデプロイされている場合、このフォルダは[lib]フォルダにあります（例えば、/opt/apache-tomcat/lib/config）]；
- b) 現在のディレクトリ [注：ValidatorがWAR ファイルとしてTomcatにデプロイされている場合、このフォルダは [lib]フォルダ（例：/opt/apache-tomcat/lib/）にあります]；
- c) \$classpath/configパッケージ；次いで
- d) \$classpathのルート。

また、設定ファイルのパスおよび名前は、ツールの起動時にコマンドラインにパラメータを設定することによって指定できます。

JARデプロイの場合：

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

TomcatでのWARデプロイの場合は、CATALINA\_OPTSに次のエントリを追加する必要があります。

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

WARデプロイを使用する場合、新しい`application.properties`ファイルをウェブアプリケーションの[WEB-INF/classes]フォルダにコピーすることも可能です。或いは既存のものが編集されます。

## 5.2. Configuring verification rule VXQV\_49

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

ユーザーがルールVXQV\_49を有効にしたい場合は、`application.properties`ファイルの`optionalEnglishQualifierValue`値を[true]に設定し、`optionalRuleType`の値を[ERROR]または[WARNING]に更新することでルールの重大度を設定できます。これら2つのプロパティのデフォルト値は上記の通りです。

## 5.3. エンドポイントの正常性を確認する

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

Validator サービスは、アプリケーションの「正常性」に関する基本情報を提供する /health エンドポイントを実装しています。

/health エンドポイントを調べるためのURLは、URL <http://localhost:8080/wipo-sequence-validator/actuator/health> です。エンドポイントには、次の情報が表示されます。

- アプリケーションが正常であれば、ステータスは「UP」です。
- データベースとの接続やディスク容量の不足などの問題によりアプリケーションに問題が生じると、「DOWN」と表示されます。

/health エンドポイントは、単純な[UP]または[DOWN]ステータスのみを表示します。

`application.properties` ファイルの次のプロパティは、正常性チェックプロセスの一部としてチェックされたすべ

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

ての正常性インジケータのステータスを含む全詳細を提供します。

/health エンドポイントに、正常性チェックプロセスの一部として実行されるDiskSpaceHealthIndicatorの詳細が含まれるようになりました。

/health エンドポイントは、このように一連のキーと値のペアを表示し、さらに詳細を含みます。以下に例を示します。

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

#### 5.4. ローカライズされたメッセージ

Validatorは、例えば検証レポートで、PCTの公式の10の言語（アラビア語、中国語、英語、フランス語、ドイツ語、ポルトガル語、日本語、韓国語、ロシア語、スペイン語）にローカライズされたメッセージを提示することができます。

デフォルトでは、メッセージは英語で提示されます。メッセージを他の言語で示すようにValidatorを設定するには、`application.properties`ファイルの`validator_locale`パラメータを適切な言語コードに設定してください。

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

い。

注：新しい「`application.properties`」ファイルで設定されたプロパティを適用するには、Validatorを再起動する必要があります。

#### 5.5. ユーザー生物名

知的財産庁は、既定の初期生物名リストに含まれない独自のユーザー生物名を提供するために、`[custom_organism.json]`という名前の新しいファイルを`[alternativeResourceBasePath]`フォルダ内

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

に作成することによってユーザー生物のリストを提供することができます。このファイルは、次の構造である必要があります。

注：既定の生物名リストとは異なり、すべての生物をアルファベットの各文字ごとにJSONファイルに分けるのではなく、1つのJSONファイルに含めます。

#### 5.6. ST.26 DTDファイルを参照する

デフォルトでは、ValidatorはST.26 DTDの最新バージョンを参照します。WIPO Sequence Validatorの現在のバージョンは、WIPO ST.26 DTD<sup>3</sup>のバージョン1.3に基づいています。

ST.26 DTDの最新バージョンのコピーは、ソースコードの`[/src/main/resources]`フォルダにあるValidatorライブラリに含まれています（これはJARまたはWARファイル内で参照される定義されたファイルパスです）。これは、次に示すよう

```
<?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>
```

に、同じフォルダ内の「catalog.xml」ファイルで参照されています。

新しいDTDを含める方法について、以下に詳述します。検証時には、XMLファイルのDOCTYPE宣言で設定されたDTDのバージョンが使用されます。まず、使用するDTDファイルの場所を識別するために、publicIdが使用されます。publicIdがカタログに含まれていない場合、システムは、Javaプロセスが実行されているルートフォルダ内のDTDファイルの検索を試みます。

---

<sup>3</sup> 2023年2月より有効

### 5.6.1. 検証用の代替DTDバージョンを指定する方法

古いバージョンのST.26 DTDを参照するWIPO ST.26ファイルを検証できるようにするには、このST.26 DTDファイルをValidatorで使用できるようにして、適切な検証を可能にする必要があります。

これを実行するには、2つの代替アプローチがあります。

- JARファイルを解凍し、追加または代替ST.26 DTDファイルへの参照を[src/main/resources]フォルダに含めます。
- catalog.xmlファイルを変更して、追加のST.26 DTDの新たなエントリを追加するか、既存のエントリを編集します。

以下はその例です。

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

JARファイルを変更する代わりに、次の手順に従う必要があります。

- a. catalog.xmlとすべてのDTDをローカルフォルダにコピーします。
- b. catalog.xmlを修正して、追加のST.26 DTDへの参照を含めます。
- c. 起動時に次のJavaシステムプロパティを設定します：  
xml.catalog.files=<path\_to\_catalog.xml>

注：Windows用のTomcatでは、次の環境変数を追加することによって行うことができます：

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

[重要：異なるバージョンのST.26 DTDを含めると、参照されたST.26 DTDに対するXMLファイルの「形式」検証が可能になりますが、「完全」検証では、検証ルールを実装するためにソースコードの変更が必要になる可能性があります。したがって、複数のDTDの使用は、「形式」検証を行う場合にのみ使用することが推奨されます。]



## 6. Validator REST API

このセクションでは、Validator APIのユースケースについて説明します。次の3つのサービスまたはユースケースがあります。

- [未処理]フォルダ内のファイルを検証; (セクション 6.1)6.1
- リクエストの一部としてファイルを検証 (セクション6.1);
- 検証のステータスを要求 (セクション 6.2)。

この (OAS 3.0API [YAML ファイル]に記載) サービスのAPI仕様は、Annex IIにすべて示されています。

### 6.1. WIPO ST.26ファイルを検証

リクエストマッピング	/api/v1/validate
メソッド	POST
Consumes	application/json
Produces	application/json
動作	[未処理]フォルダ内の既存のWIPO ST.26ファイルの検証をリクエストします。 検証リクエストのステータスを取得するため固有の“verificationID”を返します。
リクエスト	<pre>{   "currentApplicationNumber": "string",   "currentSQLVersionNumber": "string",   "parentApplicationNumber": "string",   "parentSQLVersionNumber": "string",   "sqlInputLocation": "string" (Location of   Input.xml file),   "verificationReportOutputPath": "文字列" (report.xmlの宛   先) , "nameFile": "file.xml",   "type": "string" (Possible values: full   formality), }</pre>
レスポンス	<ul style="list-style-type: none"> <li>• '202': 「受理しました」。WIPO ST.26ファイルは形式検証に通過し、ビジネスルール検証が開始されました。このコードには、検証レポートを取得するための固有のコード（「verificationID」）を示す応答メッセージも含まれます。WIPO ST.26ファイルは、処理のために[プロセス]フォルダに移動されません。</li> <li>• '400': [不正なリクエスト]。RESTリクエストの形式が正しくないか、WIPO</li> </ul>

	<p>ST.26ファイルがXML検証を通過しませんでした。このコードにはエラーの詳細を示す応答メッセージが補足されます</p> <ul style="list-style-type: none"> <li>• '400' : [不正なリクエスト]-[FILENAME_NOT_VALID]。このファイル名には無効な文字が含まれています。ファイル名にラテン文字以外を含むことは可能ですが、句読点（punctuation mark）によっては無効と見なされるものがあります</li> <li>• '404' : 「ファイルが見つかりません」。WIPO ST.26ファイルが[未処理]フォルダに見つかりませんでした</li> <li>• '500' : 「内部サーバーエラー」。内部エラーが発生しました。このコードは、エラーの詳細を示す応答メッセージで補足されます</li> </ul>
前提条件	WIPO ST.26ファイルをユーザーが指定した[未処理]フォルダ内に置く必要があります。
事後条件	<p>WIPO ST.26ファイルを、verificationReportOutputPathあるいは次の場所の[処理済み]フォルダに移します。</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>

注：検証ファイル名にラテン文字以外を含むことは可能ですが、予約文字は使用できません。

## 6.2. 検証ステータスを要求する

リクエストマッピング	/api/v1/status
メソッド	POST
Consumes	application/json
Produces	application/json
動作	特定のWIPO ST.26ファイルの検証ステータスをリクエストします
リクエスト	<pre>{   検証ID: {type: 文字列} }</pre>
レスポンス	<ul style="list-style-type: none"> <li>● '200': 成功。このコードには、以下から選択される検証プロセスのステータスとともに固有のIDを示す応答メッセージも含まれます。 <ul style="list-style-type: none"> <li>○ "RUNNING": ファイルは処理中です</li> <li>○ "FINISHED-VALID": ファイルは形式検証を正常に通過し、検証結果はレポートフォルダにあります</li> <li>○ "FINISHED-INVALID": プロセスは完了しましたが、ファイルが形式検証を通過しませんでした。検証結果は、レポートフォルダにあります</li> <li>○ "NOT_FOUND": 「検証ID」が見つかりません</li> <li>○ VERIFICATION_ID_ERROR: "verificationID"がリクエストに含まれていません</li> </ul> </li> <li>● '400': リクエストが正しくありません。RESTリクエストが適切に作成されませんでした</li> <li>● '500': サーバーエラー。内部エラーが発生しました。このコードは、エラーの詳細を示す応答メッセージも含みます</li> </ul>
事後条件	Validatorは、検証のステータスを提供します。
前提	-

以下Annex I

## Annex I : XML検証レポートの例

```
<?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>
```

このXMLインスタンスは以下からもダウンロードできます

す:[https://www.wipo.int/standards/en/sequence/example\\_verification\\_report.xml](https://www.wipo.int/standards/en/sequence/example_verification_report.xml)

以下Annex II

## Annex II：完全なAPI仕様（YAML）

完全なAPI仕様を、以下からダウンロードできます：

[https://www.wipo.int/standards/en/sequence/complete\\_validator\\_api\\_specification.yml](https://www.wipo.int/standards/en/sequence/complete_validator_api_specification.yml)

以下Annex III

## Annex III：プロパティ名（JSON）

ツール内で使用されるプロパティ名は、以下からダウンロードできます。

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

これはソースコードの一部であり、実装されると必要に応じて更新されることに留意してください。

以下Annex IV

## Annex IV：HTML検証レポートの例

### 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	

以上