



OMPI

ORGANISATION MONDIALE
DE LA **PROPRIÉTÉ**
INTELLECTUELLE

Service Web WIPO Sequence Validator

Webinaire de formation

Ce que nous allons aborder aujourd'hui

- Objet du service Validator
- Comparaison avec l'outil bureautique
- Qu'est-ce qu'un service Web?
- Introduction aux points d'extrémité du logiciel Validator (principales fonctions)
- Fonctionnement et installation du service Web
- Paramètres de demande/réponse
- Configuration du service Web

WIPO Sequence Validator : objet

- **Objectif** : s'assurer que les listages des séquences déposés sont conformes à la norme ST.26 de l'OMPI
- **Mise en œuvre** : service Web développé pour les offices de propriété intellectuelle à des fins de validation des listages des séquences déposés
- **Offices de propriété intellectuelle** : l'incorporer dans leurs propres environnements informatiques
- **Déposants** : recevoir le même rapport indiquant les éventuels avertissements/erreurs

WIPO Sequence Validator vs l'outil bureautique WIPO Sequence

Service du logiciel Validator	Outil bureautique
Conçu pour les offices	Conçu pour les déposants
Service Web	Application bureautique
<u>PAS</u> d'interface	Interface
Intégré dans l'environnement informatique	Exécuté par l'utilisateur
Deux niveaux de validation	Validation complète uniquement
Plus d'une DTD	Uniquement la dernière DTD

Qu'est-ce qu'un service Web?

- Service opérant entre deux appareils électroniques communiquant sur l'Internet ou l'Intranet
- Utilise un format de demande/réponse normalisée XML/JSON
- N'a pas besoin d'être mis en œuvre dans un langage de programmation particulier (agnostique au niveau du langage)
- Modulaire, dynamique et distribué
- Un type d'API qui fonctionne grâce à un élément SOAP ou REST comme moyen de communication
- Un contrat de service Web qui définit "quoi"/"comment" et "où"

Spécification OAS de l'API

- Spécification pour les API REST complètes
- REST est l'abréviation de REpresentational State Transfer, une architecture qui utilise un sous-ensemble du protocole HTTP
- Grâce à cette spécification, les humains et les ordinateurs peuvent comprendre les capacités de l'API sans accéder au code source
- Agnostique au niveau du langage de programmation

WIPO Sequence Validator : fondamentaux (1)

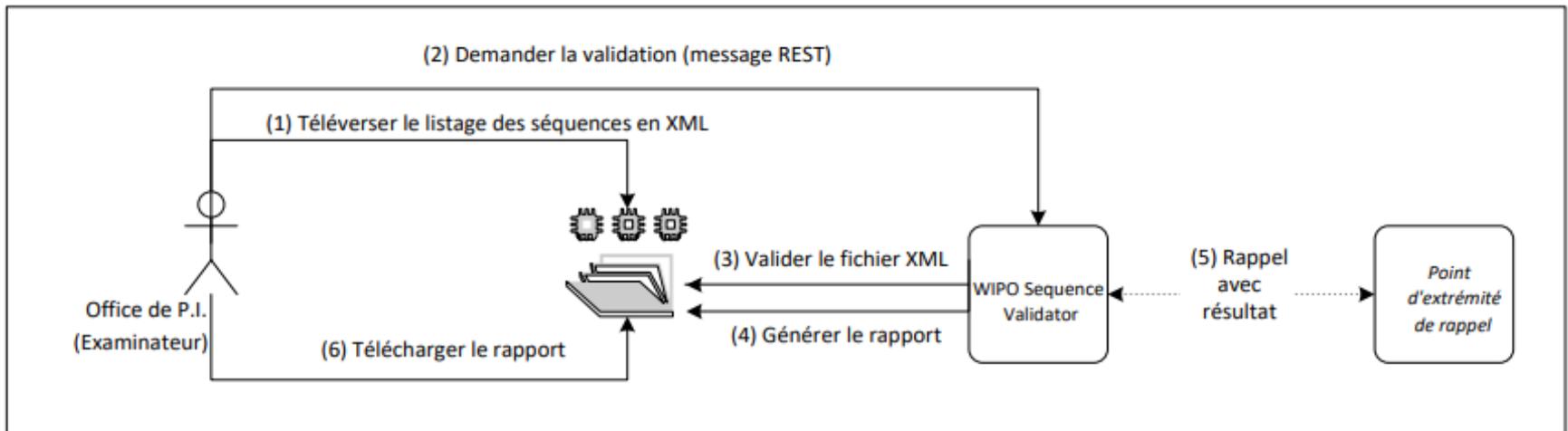
- Fonctions pour :
 - Valider un listage de séquence conforme à la norme ST.26
 - Indiquer le statut de la validation
 - Rappel du point d'extrémité avec des messages provenant du rapport de validation

- Conforme à la norme ST.90 de l'OMPI (voir la section Références)

- Spécification OAS de l'API : OAS 2.0 (voir la section Références)

- N'a pas été conçu pour être exposé à l'extérieur

WIPO Sequence Validator : fondamentaux (2)



Installation du logiciel Validator

- Nécessite l'installation de Java 8 (Java 11 sera pris en charge dans le futur)
- La structure des dossiers doit être établie dans l'emplacement parent où se trouve le service Web
- La configuration de base par défaut peut être adaptée grâce à l'utilisation d'un fichier application.properties personnalisé
- Des instructions complètes sont disponibles dans le Manuel d'utilisation du logiciel WIPO Sequence Validator :
https://www.wipo.int/export/sites/www/standards/fr/sequence/wipo_sequence_validator_operations_manual_FR.pdf

Structure des dossiers par défaut

temp/st26

temp/st26/inbox

temp/st26/outbox

temp/st26/params

temp/st26/process

temp/st26/reports

Déploiement du service Web

- Deux types de formats sont proposés : WAR et JAR
 - JAR : serveur intégré
 - WAR : les offices devront avoir un serveur Tomcat installé, ou tout serveur d'applications compatible avec Spring Boot 2 et Servlet Spec 3.1+

- L'application Swagger UI peut être utilisée pour fournir les paramètres de base/exécuter les essais; elle est disponible à l'adresse : [http://\[host-name\]:8080/swagger-ui.html](http://[host-name]:8080/swagger-ui.html)

- **Les offices doivent fournir leur point d'extrémité de rappel car celui-ci ne fait pas partie de l'outil**

Déploiement du Service en tant que JAR

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

■ Par défaut, le serveur se lance sur le port 8080 mais il est possible de modifier ce numéro de port de la manière suivante :

`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.annota
tionConfigServletWebServerApplicationContext@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.se
rvices.async.AsyncConfiguration$$EnhancerBySpringCGLIB$$c683b031] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for
```

Déploiement du Service en tant que WAR

- Pour Tomcat 8.5 :
 - Arrêter le serveur : “\$TOMCAT_ROOT\bin\catalina.bat stop”
 - Copier le fichier WAR dans “\$TOMCAT_ROOT\webapps\wipo-sequence-validator.war”
 - Lancer le serveur : “\$TOMCAT_ROOT\bin\catalina.bat start”

- “\$TOMCAT_ROOT” désigne le dossier racine du serveur Tomcat

- Au démarrage du serveur d’applications, indiquer “UTF-8” dans le fichier de propriétés système “file.encoding”

- Le port par défaut est également 8080. Pour changer le numéro de port, il faut configurer l’attribut “port” de la configuration de la Tomcat. Référence : https://tomcat.apache.org/tomcat-8.5-doc/config/http.html#Common_Attributes

Fonction Valider

- **Description** : demander la validation d'un fichier conforme à la norme ST.26 existant dans le dossier inbox. Donne un identifiant de vérification unique pour extraire le statut de la demande de validation sur demande
- **Point d'extrémité (JAR)** : [http://\[host-name\]:8080/api/\[version\]/validate](http://[host-name]:8080/api/[version]/validate)
- **Point d'extrémité (WAR)** : [http://\[host-name\]:8080/wipo-sequence-validator/api/\[version\]/validate](http://[host-name]:8080/wipo-sequence-validator/api/[version]/validate)
- **Produit en sortie** : application/json
- **Entrée** : nameFile (nom de fichier) et type de validation
- **Sortie** : identifiant verificationID et rapport de vérification

Validation “complète” versus validation “de formalité”

- Il existe deux types de validation que le logiciel Validator peut exécuter : la validation *complète* et *de formalité*
 - **De formalité** : indique si le fichier XML est valide et conforme à la DTD de la norme ST.26 (gravité : XML_WARN | XML_ERROR)
 - **Complète** : validation de formalité ET règles de vérification de la conformité déduites de la norme ST.26 de l’OMPI (gravité : ERREUR | AVERTISSEMENT)
- La validation de formalité peut être proposée de manière synchrone aux utilisateurs au moment du dépôt. La validation complète doit être effectuée de manière asynchrone.

Après la validation

- Le fichier de listage des séquences est déplacé vers le dossier “Outbox” (boîte de sortie)
- Le rapport de vérification est généré dans le dossier local “Reports” (rapports)
- Les dossiers temporaires dans le dossier local “Process” (traitement) sont supprimés
- Un identifiant de vérification identifie le processus de vérification concerné et le fichier qui est créé est étiqueté au moyen de cet identifiant
- Point d’extrémité de rappel facultatif pour envoyer une réponse qui comprend un rapport de vérification (s’il est configuré par les offices de propriété intellectuelle)

Swagger UI (d emo)

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, the text "swagger", 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 reads "API for the WIPO Sequence Validator". A section for "validation-controller" (Validation Controller) is visible, along with a "Models" section, both with right-pointing chevron icons indicating expandable content.

[hostname]:8080/swagger-ui.html

Fonction statut (vérification)

- **Description** : demande le statut de la validation pour un fichier particulier conforme à la norme ST.26
- **Point d'extrémité (JAR)** : [http://\[host-name\]:8080/api/\[version\]/status](http://[host-name]:8080/api/[version]/status)
- **Point d'extrémité (WAR)** : [http://\[host-name\]:8080/wipo-sequence-validator/api/\[version\]/status](http://[host-name]:8080/wipo-sequence-validator/api/[version]/status)
- **Produit en sortie** : application/json
- **Entrée** : identifiant de vérification (verificationID)
- **Sortie** : statut
(RUNNING/FINISHED_VALID/FINISHED_INVALID/NOT_FOUND/VERIFICATION_ID_ERROR)

Point d'extrémité de rappel

- Point d'extrémité de rappel : une API distincte qui reçoit des informations émanant du service Web
- La validation peut également être effectuée par un point d'extrémité de rappel qui appelle le service Web
- Sera exécuté de manière asynchrone
- Doit être conforme au contrat de service Web indiqué dans le manuel d'utilisation
- Les formats des demandes et des réponses doivent également être identiques au format standard défini

Format de la demande

■ Fonction Valider

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

■ Fonction statut

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

Réponse obtenue (1)

- **processID** : déjà fournie par le logiciel WIPO Sequence Validator
- **seqType** : une valeur fixe : 'ST.26'
- **httpStatus** : code statut HTTP du service : 'Succès' / 'Échec' de concert avec le code Erreur HTTP et la description correspondante
- **applicationNumber** : pour la demande dont le listage des séquences fait partie, par ex., PCTUS1234567 ou 23456789.
- **currentSQLVersionNumber** : le numéro de version pour ce listage de séquences (attribué en interne par un office, par ex., 1.1.
- **parentApplicationNumber** : toute demande parent associée, par ex., 12345678.
- **parentSQLVersionNumber** : la modification du numéro du listage de séquences (attribué en interne par un office, par ex., 3.2.
- **verificationReportOutputPath** : indique le chemin d'accès complet pour accéder au rapport
- **startTime** : l'heure à laquelle le service a commencé la validation, par ex. 2019-07-29 15:59:37.784
- **endTime** : l'heure à laquelle le service a achevé la validation, par ex. 2019-07-29 15:59:37.859
- **elapsedTime** : indique le temps total nécessaire pour la validation des listages des séquences, par ex., 0 h
(s) 0 min(s) 0 sec(s) 75 ms

...

Réponse obtenue (2)

- **totalWarningQuantity** : somme de tous les avertissements du rapport de vérification, par ex., 2.
- **totalErrorQuantity** : somme de toutes les erreurs du rapport de vérification, par ex., 0.
- **seqInputQuantity** : nombre total de séquences figurant dans la section des informations générales du projet indiqué par le déposant, par ex., 10.
- **seqIDQuantity** : étape de la vérification destinée à vérifier le nombre de séquences dans le listage des séquences par rapport au nombre indiqué par le déposant, par ex. 10.
- **errorSummary** : un résumé des erreurs signalées dans le rapport de vérification, avec les informations détaillées suivantes :

```
[
{ (il s'agit de la numérotation séquentielle de chaque code erreur
signalé, par ex. 1, 2, 3, etc.) : , errorCode: (identification
numérique ou alphanumérique de l'erreur, par ex., 1 (####) pour les
erreurs, ou 2 (####) pour les avertissements) , description (paire
code erreur-description) :, sequenceNumber (SEQ ID), XPath: (peut être
utilisé pour parcourir les éléments et les attributs dans un document
XML, par ex., / ST26SequenceListing/SequenceData [1
]/INSDSeq/INSDSeq_sequence) , characterRange: (par ex., 10-20 à partir
de la position de caractère 10 jusqu'à la position 20 dans l'élément
INSDSeq_sequence)
} , { (même information signalée pour l'erreur suivante) }, { . ... ..
}
```

Rapport de validation

```

<?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>

```

Rapport de validation : exemple

```

<?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>

```

Configuration des paramètres – par défaut

- Chemin d'accès de base (Base path) à utiliser pour le reste des dossiers = **../temp/st26/**
- Dossier où placer les fichiers à traiter = Base Path+/**inbox/**
- Dossier pour stocker les fichiers conformes à la norme ST26 une fois validé =Base Path+/**outbox/**
- Dossier pour stocker les rapports de validation = Base Path+/**reports/**
- Dossier parent pour les fichiers de validation complète et de formalité =Base Path+/**process/**
- Dossier pour stocker les paramètres =Base Path+/**params/**
- AlternativeResourceBasePath=Base Path+/**alt_resources**
- Localisation utilisée pour les messages localisés figurant dans le rapport de vérification = **en**
- URL du point d'extrémité de rappel qui sera utilisée pour informer des résultats de la validation =**http://callbackservice/api/endpoint**
- **Également connexion (logging) etc.**

Autre configuration possible (1)

- La configuration est possible au moyen d'un nouveau fichier application.properties

- Si un autre fichier application.properties est utilisé, l'outil recherchera une alternative dans l'ordre suivant :
 - Un dossier “/config” dans le répertoire en cours
 - Le répertoire en cours
 - Un chemin de classe \$ classpath ou un paquetage de configuration (config); puis
 - Dans la racine \$ classpath OU
 - À l'emplacement précisé au moyen du paramètre dans la ligne de commande

- **Note : l'outil Validator devra être redémarré pour que la nouvelle configuration soit prise en compte.**

Autre configuration possible (2)

- Il est en particulier possible de personnaliser les deux éléments suivants :
 - La langue des messages du rapport de vérification : le paramètre “`validator_locale`” du fichier `application.properties` doit être défini dans le code langue approprié, par ex., “`validator_locale=es`” pour changer la langue en espagnol.
 - Noms d’organismes personnalisés : fournir un seul fichier JSON de noms d’organismes qui ne font pas partie de la liste prédéfinie, qui doit avoir la structure suivante :

```
[  
  {"value": "Custom Organism Sample"},  
  {"value": "Custom Organism Sample 2"}  
]
```
 - Le nouveau fichier `custom_organism.json` devrait se trouver à l’emplacement indiqué par `alternativeResourceBasePath`.

Autre configuration possible : DTD (1)

- Par défaut, le service Validator effectuera la validation par rapport à la version la plus récente de la DTD : fournie dans le dossier `“/src/main/resources”` (actuellement version 1.3)

- Il est également possible d’effectuer une validation par rapport à une DTD plus ancienne, au moyen de deux méthodes :
 - La première consiste à :
 - Décompresser le fichier JAR et ajouter une référence à la DTD supplémentaire ou alternative de la norme ST.26 dans le dossier `“src/main/resources”`;
 - Modifier le fichier `“catalog.xml”` en ajoutant une nouvelle entrée définissant la DTD supplémentaire de la norme ST.26 ou en modifiant l’entrée existante.

Autre configuration possible : DTD (2)

- La deuxième méthode, au lieu de modifier le fichier JAR, consiste à :
 - Copier le fichier “catalog.xml” et toutes les DTD dans un dossier local;
 - Modifier le fichier “catalog.xml” pour ajouter une référence à la DTD de la norme ST.26 supplémentaire; et
 - Définir cette propriété système Java au lancement : “xml.catalog.files=<path_to_catalog.xml>”

■ **Note : cela fonctionnera uniquement pour la validation de formalité car des mises à jour peuvent être nécessaires pour les règles de vérification officielles déduites du corps de la norme.**

Séance de Questions-Réponses

standards@wipo.int

Références

- Page d'accueil WIPO Sequence (fichiers binaires du logiciel Validator et manuel d'utilisation en anglais) :

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

- Norme ST.26 de l'OMPI :

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

- Mise en œuvre de la norme ST.26 FAQ :

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

- Norme ST.90 de l'OMPI :

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

- Spécification OAS 2.0 :

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