



WIPO Sequence Validator 网络服务

网络研讨会培训

培训内容

- Validator服务的目的
- 与桌面工具的对比
- 网络服务是什么？
- Validator终端简介（主要功能）
- 网络服务的运行和安装
- 请求/响应参数
- 网络服务的配置

WIPO Sequence Validator: 目的

- **目标:** 确保所提交的序列列表符合WIPO ST.26
- **实施:** 为知识产权局开发的用来验证所提交序列列表的网络服务
- **知识产权局:** 将该服务整合到其各自的信息技术环境
- **申请人:** 收到相同的错误/警告报告

WIPO Sequence Validator与 WIPO Sequence桌面工具

Validator服务	桌面工具
面向知识产权局	面向申请人
网络服务	桌面应用
无界面	有界面
整合到信息技术环境	由用户运行
两层验证	仅全面验证
多版本DTD	仅最新DTD

网络服务是什么？

- 在两台通过互联网或局域网进行通信的电子设备之间运行的服务
- 使用标准化XML/JSON请求/响应格式
- 无需以某种特定编程语言实施（语言无关）
- 模块化、动态化、分布化
- 使用SOAP或REST作为通信手段来运行的一种API
- 确定“内容”（what）/“方式”（how）以及“地点”（where）的网络服务协定

OAS API规范

- RESTful API的规范
- REST代表具象状态传输——一种使用HTTP子集的架构
- 在此规范下，人类和计算机无需获取源代码，就能理解API的功能
- 编程语言无关

WIPO Sequence Validator: 基础 (1)

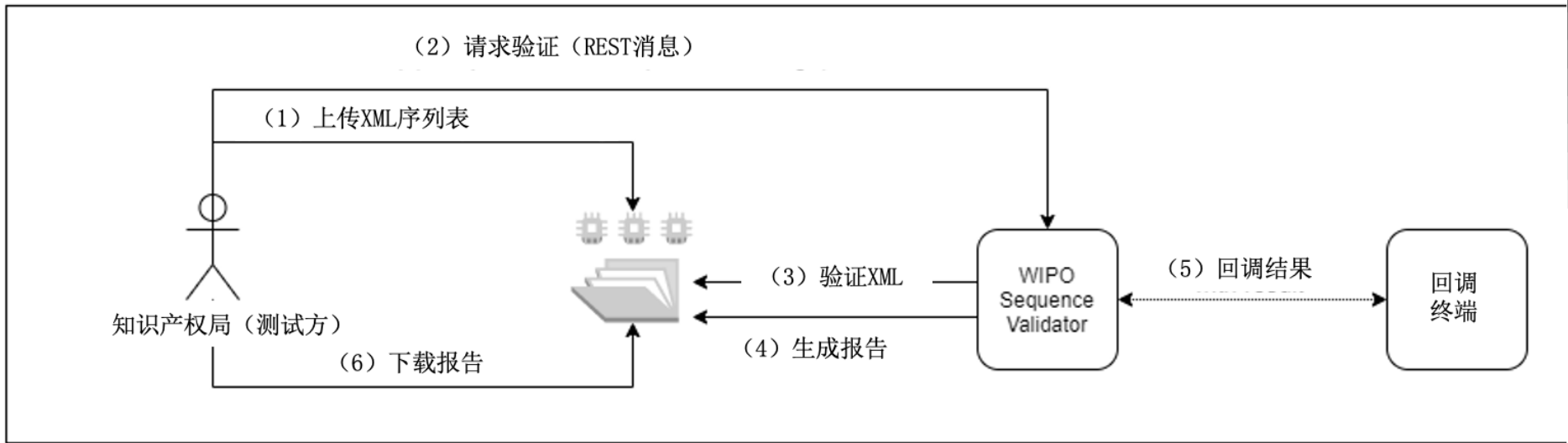
- 功能:
 - 验证ST.26序列表
 - 提供验证状态
 - 以验证报告中的信息回调终端

- 符合WIPO ST.90 (见参考资料)

- 网络API规范: OAS 2.0 (见参考资料)

- 并非为进行外部展示而设计

WIPO Sequence Validator: 基础 (2)



安装Validator

- 需要安装Java 8（未来将支持Java 11）
- 文件夹结构必须设在网络服务的父级位置
- 基础默认配置可以通过提供自己的application.properties文件进行定制
- 完整说明可见《WIPO Sequence Validator操作手册》：
https://www.wipo.int/export/sites/www/standards/en/sequence/wipo_sequence_validator_operations_manual.pdf

默认文件夹结构

temp/st26

temp/st26/inbox

temp/st26/outbox

temp/st26/params

temp/st26/process

temp/st26/reports

网络服务部署

- 提供两种类型的服务：WAR和JAR
 - JAR——嵌入在服务器中
 - WAR——各局需安装Tomcat服务器，或任何与Spring Boot 2和Servlet Spec 3.1+兼容的应用服务器
- 可以用Swagger UI来提供基本参数/进行测试：[http://\[host-name\]:8080/swagger-ui.html](http://[host-name]:8080/swagger-ui.html)
- 各局必须提供其回调终端，因为工具中不含这一部分

JAR的服务部署

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

■ 默认端口是8080，但服务器端口可以使用以下语句进行更改：

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

WAR的服务部署

- Tomcat 8.5:
 - 停止服务器：“\$TOMCAT_ROOT\bin\catalina.bat stop”
 - 将WAR复制到“\$TOMCAT_ROOT\webapps\wipo-sequence-validator.war”
 - 启动服务器：“\$TOMCAT_ROOT\bin\catalina.bat start”
- “\$TOMCAT_ROOT”指的是Tomcat应用服务器的根文件夹
- 启动应用服务器时，必须设置文件编码为UTF-8
- 默认端口还是8080。要进行更改，必须设置Tomcat配置文件中的“port”属性。参考资料：https://tomcat.apache.org/tomcat-8.5-doc/config/http.html#Common_Attributes

验证功能

- **说明：** 请求验证inbox文件夹中已有的ST.26文件。根据请求返回唯一的verificationID，以供获取验证请求的状态。
- **终端（JAR）：** [http://\[host-name\]:8080/api/\[version\]/validate](http://[host-name]:8080/api/[version]/validate)
- **终端（WAR）：** [http://\[host-name\]:8080/wipo-sequence-validator/api/\[version\]/validate](http://[host-name]:8080/wipo-sequence-validator/api/[version]/validate)
- **生成：** application/json
- **输入：** nameFile和验证类型
- **输出：** verificationID和验证报告

“全面”验证与“形式”验证

- Validator可以进行两种验证：**全面验证**与**形式验证**
 - **形式验证**：XML文件是否合格并且符合ST. 26的DTD（严重性：XML_WARN | XML_ERROR）
 - **全面验证**：形式验证和验证是否符合源于WIPO ST.26的验证规则（严重性：ERROR | WARN）
- 形式验证可在用户提交申请时同步提供。全面验证并非同步进行。

验证之后

- 序列列表文件移入本地“Outbox”文件夹
- 在本地“Reports”文件夹中生成验证报告
- 本地“Process”文件夹中的临时文件夹被删除
- 有一个确认具体验证过程的验证ID，所创建的文件夹以此ID进行标识
- 可选择回调终端提供包含验证报告的响应（在知识产权局进行了配置的情况下）

Swagger UI (演示)

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 "[Base URL: localhost:8080/]" and a link to the API documentation: <http://localhost:8080/v2/api-docs?group=api-infos>. The description reads "API for the WIPO Sequence Validator". A section for "validation-controller" is visible, labeled "Validation Controller" with a right-pointing arrow. Below this, a "Models" section is also visible with a right-pointing arrow.

[hostname]:8080/swagger-ui.html

(检查) 状态功能

- 说明：请求特定ST.26文件的验证状态
- 终端 (**JAR**) : [http://\[host-name\]:8080/api/\[version\]/status](http://[host-name]:8080/api/[version]/status)
- 终端 (**WAR**) : [http://\[host-name\]:8080/wipo-sequence-validator/api/\[version\]/status](http://[host-name]:8080/wipo-sequence-validator/api/[version]/status)
- 生成： application/json
- 输入： verificationID
- 输出： 状态
(RUNNING/FINISHED_VALID/FINISHED_INVALID/NOT_FOUND/VERIFICATION_ID_ERROR)

回调终端

- 回调终端：从您的网络服务接收信息的单独API
- 也可以由回调终端呼叫网络服务来进行验证
- 非同步进行
- 必须符合操作手册中提供的网络服务协议
- 请求和响应的格式必须以所定义的标准格式提供

请求格式

■ 验证功能

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

■ 状态功能

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

所提供的响应 (1)

- **processID**: 已由WIPO Sequence Validator提供
- **seqIType**: 固定值: 'ST.26'
- **httpStatus**: 来自服务的HTTP状态代码: 'Success'/'Failure' 以及HTTP错误代码和相应说明
- **applicationNumber**: 序列为其组成部分的专利申请号, 例如PCTUS1234567或23456789
- **currentSQLVersionNumber**: 该序列的版本号 (由知识产权局内部分配), 例如1.1
- **parentApplicationNumber**: 任何相关的主申请, 例如12345678
- **parentSQLVersionNumber**: 父级序列的修订版本号 (由知识产权局内部分配, 例如3.2)
- **verificationReportOutputPath**: 提供报告的完整路径
- **startTime**: 服务开始验证的时间, 例如2019-07-29 15:59:37.784
- **endTime**: 服务停止验证的时间, 例如2019-07-29 15:59:37.859
- **elapsedTime**: 提供验证序列所需的总时长, 例如0 hr(s) 0 min(s) 0 sec(s) 75 ms
- ...

所提供的响应 (2)

- **totalWarningQuantity:** 验证报告中所有警告的总数，例如2
- **totalErrorQuantity:** 验证报告中所有错误的总数，例如0
- **seqInputQuantity:** 来自申请人项目输入中一般信息部分的序列总数，例如10
- **seqIDQuantity:** 对照申请人输入的数字检查序列表中序列数量的验证步骤，例如10
- **errorSummary:** 验证报告中所报告的错误总结，包含以下具体内容：

```
[
  {errorIndexID (所报告的错误代码的序列号，例如1、2、3等) : , errorCode: (错误的数字或字母数字标识，例如错误为1(####)或警告为2(####)), description (匹配的错误代码-说明) :, sequenceNumber (SEQ ID), XPath: (可以用来浏览XML文件中的元素和属性，例如/ ST26SequenceListing/SequenceData[1]/INSDSeq/INSDSeq_sequence) , characterRange: (例如10-20，在INSDSeq_sequence元素中从字符位置10到20)
} , { (为下个错误报告相同信息) }, { . . . . }
```

验证报告

```

<?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>

```

验证报告：样例

```

<?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>

```


参数配置——默认

- 其余文件夹使用的基本路径=**../temp/st26/**
- 存放待处理文件的文件夹=**Base Path+/inbox/**
- 保存验证过的ST.26文件的文件夹=**Base Path+/outbox/**
- 保存验证报告的文件夹=**Base Path+/reports/**
- 完整和形式验证文件夹的父文件夹=**Base Path+/process/**
- 保存参数的文件夹=**Base Path+/params/**
- 替代资源基本路径=**Base Path+/alt_resources**
- 验证报告中本地化消息所使用的本地化= **en**
- 将用于通知验证结果的回调终端的
URL=**http://callbackservice/api/endpoint**
- 还有日志记录等。

替代配置（1）

- 可以通过提供新的`application.properties`文件进行配置
- 如果提供了`application.properties`替代文件，工具将会按以下顺序检索替代项：
 - 当前目录中的“/config”文件夹
 - 当前目录
 - `$classpath`或`config`包；然后
 - `$classpath`根，或
 - 在命令行用参数表明的地点
- 注：在新配置投入应用之前，需重启**Validator**

替代配置（2）

- 特别是，可以进行以下两种定制：
 - 验证报告的消息语言——`application.properties` 文件的“`validator_locale`”参数必须设置为适当的语言代码，例如“`validator_locale=es`”将语言改为西班牙文。
 - 定制生物体名称：提供并不构成打包列表一部分的单独的生物体名称 JSON 文件：

```
[  
  {"value": "Custom Organism Sample"},  
  {"value": "Custom Organism Sample 2"}  
]
```
 - 新的 `custom_organism.json` 文件应在替代资源基本路径（`alternativeResourceBasePath`）所指明的位置

替代配置： DTD （1）

- 在默认配置下， Validator服务将根据最新DTD进行验证： 根据“/src/main/resources”（目前为1.3版）
- 还可以使用两种不同方法， 根据旧版本的DTD进行验证：
 - 方法一：
 - 解压JAR文件， 在“src/main/resources”文件夹中为附加或替代的ST.26 DTD文件添加参考资料
 - 通过为附加的ST.26 DTD加入新条目或编辑已有条目， 修改“ catalog.xml”文件

替代配置： DTD （2）

■ 方法二，与修改JAR文件不同：

- 将“catalog.xml”和所有DTD复制到本地文件夹；
- 修改“catalog.xml”，为附加的ST.26 DTD添加参考资料； 和
- 在启动时设置此Java系统属性：
“xml.catalog.files=<path_to_catalog.xml>”

■ 注：这只对形式检查有效，因为可能需要对源于标准主体的业务验证规则进行更新

问答环节

standards@wipo.int

参考资料

- WIPO Sequence主页（Validator二进制及操作手册英文版）：

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

- WIPO标准ST.26:

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

- 实施WIPO ST.26的常问问题:

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

- WIPO标准ST.90:

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

- OAS 2.0规范:

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