

KANTA Arkistopalvelu vanhojen asiakirjojen validointiohjelma, versio 1.2

Asennusohje

21.9.2023

Mika Suomalainen
Julkinen

Muutoshistoria

Versio	Muutos	Tekijä	PVM
1.0	Ensimmäinen julkaistu versio	Mika Suomalainen	17.11.2017
1.1	Tuki Java SE Runtime Environment 7(JRE) ajoympäristölle,koodistotarkistus kansallisestakoodistopalvelusta,xhtml ja text plain validointi,koontitiedoston validointi, tuki isoillekoontitiedostoille, lokikäsittelyn tehostaminen, HENnäkymän arkistoinninesto	Mika Suomalainen	25.4.2018
1.2	Päivitetty Java 11 tasolle	Mika Suomalainen	21.9.2023

Sisällys

Muutoshistoria	1
1 Asennuspaketin sisältö	3
2 Vaatimukset asennusympäristölle.....	3
3 Asentaminen	3
3.1 Arkisto-old-documents-validator-1.0.22.tar.gz paketin asennus.....	3
3.2 Windows-version asennus	4
4 Konfigurointi	4
4.1 olddocuments.properties -tiedoston asetukset	4
4.2 Lokitus.....	6
4.2.1 Tietokantalogi	6
4.2.2 Log4j	8
5 Asennuksen onnistumisen todentaminen	9
6 Latausohjelman käyttö ja lokit.....	9
6.1 Konfiguraatio.....	9
6.2 Ohjelman ensimmäinen ajo	10
6.3 Virheiden korjaaminen	11
6.4 Uudelleenajo	12
6.5 Koontitiedoston skeema.....	12

1 Asennuspaketin sisältö

Asennettava kokonaisuus koostuu seuraavista osista/tiedostoista:

Arkisto-old-documents-validator-1.0.22.tar.gz	Sisältää vanhojen asiakirjojen validointityökalun toteutusluokat, konfiguraatio-tiedostot ja tarvittavat kirjastot sekä ajoskriptit
Arkisto-old-documents-validator-1.0.22.zip	
zip-tiedosto on tarkoitettu windows:iin ja .tar.gz Linux:ille.	

2 Vaatimukset asennusympäristölle

KANTA Arkistopalvelu vanhojen asiakirjojen latausohjelma vaatii toimiakseen Java version 11.

Validoitaessa asiakirjoja KanTo-järjestelmän komponentteja ei tarvita. Koodistotarkistus tapahtuu kansallisesta koodistopalvelusta niin, että koodisto tarkistetaan ajonaikaisesti ja tallennetaan välimuistiin. Koodistotarkistuksen luonteesta johtuen validointiajtoa suorittavalta koneelta on oltava yhteys kansalliseen koodistopalveluun (<http://koodistopalvelu.kanta.fi>).

3 Asentaminen

3.1 Arkisto-old-documents-validator-1.0.22.tar.gz paketin asennus

Kirjautu koneelle, jossa on käytössä tar, java. Pura tar.gz-paketti hakemistoon, josta ohjelmaa ajetaan:

```
$ tar xvfz arkisto-old-documents-validator-1.0.22.tar.gz
```

Paketti sisältää ohjelman tarvitsemat jar-kirjastot, esimerkkikonfiguraatiot ja ohjelman ajoskriptit. Jos ohjelma on jo aiemmin konfiguroitu käyttöön, ota konfiguraatioista varmuuskopio ennen paketin purkamista.

3.2 Windows-version asennus

Windows-versio on pakattu zip-paketiksi, joka voidaan purkaa millä tahansa yhteensopivalla ohjelmalla. Kun zip-paketti on purettu hakemistorakenteeseen, etsi tiedosto olddocuments.properties tiedosto ja poista se. Muuta tiedoston olddocuments.properties-windows nimi muotoon olddocuments.properties.

4 Konfigurointi

4.1 olddocuments.properties -tiedoston asetukset

Esimerkki on Linux-ympäristön asetuksin, jakelupaketissa on mukana myös Windows-yhteensopiva konfiguraatitiedosto.

```
# xpathconfig-polku. Konfiguraatiot asiakirjan tietojen validointiin.  
xpathconfigpath=config/xmlconfig
```

```
# skeema asiakirjojen validointiin  
schemapath=config/schema/CDA_Fi.xsd
```

```
# skeema xhtml validointiin  
xhtmlschemapath=config/schema/xhtml11-strict.xsd
```

```
# skeema koontitiedoston validointiin  
koontitiedostoschemapath=config/schema/koontitiedosto.xsd
```

```
#validoitaanko koontitiedosto  
validatekoontitiedosto=true
```

```
#kansallisen koodistopalvelimen tiedot  
thl_koodisto_namespace=urn:codeapi:Codeservice  
thl_koodisto_service=CodeService  
thl_koodisto_wsdl=http://koodistopalvelu.kanta.fi/codeserver/ws/services/CodeService?wsdl  
thl_koodisto_endpoint=http://koodistopalvelu.kanta.fi/codeserver/ws/services/CodeService
```

```
#Kuinka monta vuorokautta koodistotarkistuksen tuloksia cachetetaan.  
thl_koodisto_cachen_ttl=14
```

```
# Levylokitiedoston sijainti  
logpath=logs/runlogs/
```

```
# ajon virhelokitiedoston sijainti  
error_logpath=./logs/
```

```
# lokimerkinnän tunniste  
logId=laturi_1.x
```

```
#koontitiedoston perusnimi 1. ajokerralla  
manifestfile =koontitiedosto.xml
```

```
#uudelleenajo-koontitiedoston nimen alkuosa  
rerunmanifestfileprefix =rerun_koontitiedosto_
```

```
# ajokierrostieto  
ajokierros=1
```

```
# aineiston juuripolku  
filepath=/data/cdar2-signed/180370-900X/2006
```

```
# Käytettävien säikeiden lukumäärä hakemistojen käsittelyyn  
threadcount_directory = 5
```

```
# Käytettävien säikeiden lukumäärä asiakirjojen käsittelyyn  
threadcount = 50
```

```
# Yksittäisen asiakirjan käsittelysäikeen timeout millisekunteina  
threadtimeout= 60000
```

```
# kuinka monta sekuntia asiakirjan luontiaika saa olla tulevaisuudessa  
offset=5
```

```
# Vaadittu palvelunantaja
organizationid=1.2.246.99.9999999.88.1

# Liityntäpiste
pluginsystemid=1.2.246.99.9999999.88.01

# Vaadittu rekisterinpitäjä asiakirjoilla
custodianid=1.2.246.99.9999999.88.1

subsystem=Validointiajo

checksignature=false
validatepdf=true
checkpdf=true
checktextplain=true
checkxhtml=true
checkcodes=true
testmode=false

# Ajonaikana käytettävän tietokannan jdbc-url. Tietokannan polku voidaan
muuttaa tarvittaessa.
jdbcurl=jdbc:hsqldb:file:/tmp/hsqldb/validoija/db;hsqldb.write_delay=false

# jdbc-ajuri, ei tarvitse muuttaa ellei tietokanta alustaa vaihdeta
toiseksi jdbcdriver=org.hsqldb.jdbcDriver

# generoidaanko edellisen version mukainen rerun-loki. Voi olla hyödyksi
backup:ina. rerunlog=false
logAll=true
sqllog=/tmp/redolog
```

4.2 Lokitus

4.2.1 Tietokantalogi

Sovellus tuottaa tietokantaan lokia onnistuneesti käsitellyistä hakemistoista, palvelutapahtumista ja asiakirjoista. Tietokanta koostuu seuraavista tauluista kuvauksineen:

ASIAKIRJA		
CDA_ID	asiakirjan oid	
TILA	asiakirjan tila	Onko asiakirja käsitelty onnistuneesti (OK/ERROR)
TIEDOSTOPOLKU	asiakirjan polku	Sijainti tiedostojärjestelmässä

HAKEMISTOTAULU		
HAKEMISTO	hakemistopolku	Onnistuneesti käsitelty hakemistot lisätään tähän tauluun. Uudelleenajossa näitä hakemistoja ei enää lueta.

MANIFESTVIRHE		
TIEDOSTO	manifest-tiedoston polku	Virheelliset manifest-tiedostot lokitetaan tähän tauluun

PALVELUTAPAHTUMA		
CDA_ID	palvelutapahtuman oid	
ASIAKIRJATARKISTOITU	onko asiakirjat käsitelty onnistuneesti	true jos palvelutapahtuman kaikki asiakirjat on arkistoitu / validoitu onnistuneesti. Käytetään uudelleenajossa

VIRHETAULU		
CDA_ID	asiakirjan oid	asiakirjan oid
KUVAUS	virheen kuvaus	Virheen kuvaus
TARKKAKUVAUS	virheen tarkempi kuvaus	Virheen tarkka kuvaus
KOODI	virheen koodi	Virhekoodi laturista tai ulkoisesta järjestelmästä
KORJATTU		Käytetään ajonaikaisesti, jos asiakirjassa on virhe joka on uudelleenajossa korjattu asetetaan arvoksi "true"

VIRHEHAKEMISTOTAULU		
---------------------	--	--

HAKEMISTO	virheellinen hakemisto	mm. hakemistot joihin ei ole oikeutta logitetaan tähän tauluun
-----------	------------------------	--

Tietokantaan saa ajon jälkeen yhteyden HSQLDB käyttöliittymätyökalulla. Koska tietokantamoottori pyörii ajonaikaisen jvm:n sisällä kaksi java-instanssia ei voi käyttää yhtä aikaa samoja tietokantatiedostoja. Eli käyttöliittymää ja validaattoria ei voi ajaa yhtä aikaa.

4.2.2 Log4j

Sovellus lokittaa omaan erityiseen lokiinsa edellä mainitun properties-tiedoston mukaan Log4j-pohjainen palveluloki voidaan konfiguroida seuraavasti: Ohjelman polkuun annetaan log4j.properties-tiedosto. Sovelluspaketin mukana on seuraavanlainen esimerkkikonfiguraatio, jota voidaan tarpeen mukaan muokata.

```
#----- KANTA ARKISTOERAJOJEN LOKI -----  
log4j.logger.fi.kanta=DEBUG,eraajolog  
log4j.appender.eraajolog=org.apache.log4j.RollingFileAppender  
log4j.appender.eraajolog.File=/tmp/olddocuments.log  
log4j.appender.eraajolog.MaxFileSize=50MB  
log4j.appender.eraajolog.MaxBackupIndex=5  
log4j.appender.eraajolog.layout=org.apache.log4j.PatternLayout  
log4j.appender.eraajolog.layout.ConversionPattern=%d{dd.MM.yyyy  
HH:mm:ss.SSS} %-5p [%t] : %c{1}::%M %m%n
```

Sinisellä merkityjä kohtia on mahdollista muuttaa tarpeen mukaan. Oman lokinsa säätöviuilla logAll säätää sitä, lokitetaanko onnistuneet tapahtumat vai pelkästään epäonnistuneet.

Tuotantoasetukset

Tehokkuussystä täytyy isojen asiakirjamassojen käsittelyssä kääntää lokitus kirjoittamaan korkeintaan INFO -tasoiset viestit, mielellään vain WARN ja ERROR - tasoiset. Näin tekninen lokitus ei hidasta asiakirjojen tallennuksia arkistoon.

Esim.

```
log4j.logger.fi.kanta=INFO,eraajolog  
log4j.logger.fi.kanta=WARN,eraajolog
```

Virhelokitus

Ohjelma kerää mahdolliset ajonajanaikaiset virheet errors.log tiedostoon, jonka hakemisto annetaan konfigurointitiedostossa. Virhelokista käy ilmi tiedoston polku ja nimi sekä virheen seliteteksti. Virheet listautuvat lokille siinä järjestyksessä, kun niitä ilmenee, joten useamman hakemiston latauksissa järjestys ei ole hakemistokohtainen.

Tiedosto kannattaa poistaa ennen uutta ajokierrosta, jolloin näkee helposti ajon onnistumisen. Jos virheloki jää tyhjäksi on ajo mennyt onnistuneesti läpi. Jos tiedostoa ei poisteta, lokitus jatkuu tiedoston lopusta

```
log4j.logger.errorlog=ERROR,errorlog
log4j.appender.errorlog=org.apache.log4j.FileAppender
log4j.appender.errorlog.File=./logs/errors.log
log4j.appender.errorlog.layout=org.apache.log4j.PatternLayout
log4j.appender.errorlog.layout.ConversionPattern=%d{dd.MM.yyyy HH:mm:ss.SSS}
%-5p : %m%n
```

5 Asennuksen onnistumisen todentaminen

Aja pieni koe-erä asiakirjoja ja katso lokilta onnistuiko niiden validointi. Ohjelmalle olevista testeistä on erillinen testausohje.

6 Latausohjelman käyttö ja lokit

Validointiohjelma ajetaan käynnistämällä paketissa mukana oleva validoi

Palvelu- ja virhelokille kirjoitetaan log4j-määritysten mukaisesti tietoa ohjelmiston toiminnasta ja suorituksesta sekä virhetilanteista laajemmat java-ilmoitukset.

Ennen ohjelman ajoa on tärkeää konfiguroida oikein olddocuments.properties -tiedosto. Tiedostoon konfiguroidaan mm. ajettavan aineiston sijainti levyllä.

6.1 Konfiguraatio

Ennen latausajoa tarkastetaan, että ajolle on konfiguroitu aineiston mukaiset asetukset

- palvelunantaja

Kela / Suomalainen Mika

21.9.2023

Julkinen

- rekisterinpitäjä sekä
- liityntäpiste ja palvelupyyntötyyppi.

Suoritusoikeus tarkistus konfiguraatiosta

Konfiguraatiossa määritetyn palvelunantajan täytyy vastata ladattavan aineiston koontitiedosto.xml palveluntantajaa, muuten hakemiston suoritus päättyy virheeseen.

```
organizationid=1.2.246.537.10.15675350.10.0  
<palvelutapahtumat palvelujenantaja="1.2.246.537.10.15675350.10.0" .. />
```

Suoritusoikeuden tarkistus palvelulta

Suoritusoikeus arkistointiin tarkastetaan hakemistokohtaisesti, jolloin luetaan aineiston palvelunantaja ja konfiguraatiossa määritetyt liityntäpiste ja palvelupyyntötyyppi.

```
pluginssystemid=1.2.246.99.9999999.88.01
```

Rekisterinpitäjätarkistus

Rekisterinpitäjän vastaavuus tarkastetaan asiakirjakohtaisesti lukemalla tiedot konfiguraatiosta sekä asiakirjasta.

```
custodianid=1.2.246.537.10.15675350.19.0
```

6.2 Ohjelman ensimmäinen ajo

Tarkista että lokihakemisto on tyhjä, jos sieltä löytyy runlogs hakemisto, ajo päättyy virheeseen.

Ajonaikana käytettävän tietokannan tulee olla alussa tyhjä, tarkista olddocument.properties tiedoston jdbcurl:

```
jdbcurl=jdbc:hsqldb:file:/tmp/hsq1/validoija/db;hsqldb.write_delay=false
```

jdbcurl muuttujassa osoitettu hakemisto tulee olla tyhjä, viimeinen osio polussa kuvaa tietokantatiedostojen nimeä, esim:

Kela / Suomalainen Mika

21.9.2023

Julkinen

```
$ pwd && ls  
/tmp/hsq1/laturi  
db.properties          db.script
```

Aja validointi komennolla: **./validoi.sh** (Windows: validoi.cmd)

Jos ajo menee loppuun asti kaatumatta, tulostuu yhteenveto:

```
Asiakirjoja arkistoitu / validoitu  
433 OK  
2 ERROR  
Virheellisiä manifest tiedostoja  
1  
Palvelutapahtumat (lukumäärä, onko kaikki asiakirjat arkistoitu/validoitu)  
143 TRUE  
2 FALSE  
Hakemistoja käsitelty onnistuneesti  
27  
Eräajo päättyi: Thu Oct 26 14:02:31 EEST 2017, kesto 19372 ms
```

Esimerkissä on kaksi dokumenttia jotka eivät ole valideja eikä niitä ole arkistoitu. Lisäksi mukana oli virheellinen manifest (koontitiedosto.xml), joten ko. hakemiston kaikki asiakirjat ovat arkistoitumatta. Jos ajossa ei tullut virheitä on validointi valmis. Jos ajossa oli virheitä tai ajo kaatui virheeseen josta latausohjelma ei voi jatkaa, virheet tulee korjata ja suorittaa uudelleenajo. Katso ohjeita virheenselvitykseen kappaleesta 6.3

6.3 Virheiden korjaaminen

Virheet löytyvät helposti errors.log –tiedostosta, mutta jos virheitä on huomattava määrä, kaikki asiakirjavirheet saadaan tietokannasta sh-skriptillä **./raportti.sh** (Windows: raportti.cmd)

Jos ajo kaatui ajonaikana muuhun kuin asiakirjavirheeseen tutki errors.log ja olddocuments.log tiedostoja. Syy pitäisi löytyä java-exception:in muodossa. Korjaa virhe ja suorita uudelleenajo.

6.4 Uudelleenajo

Ota varmuuskopio logs/runlogs hakemistosta jos hakemisto löytyy ja poista se.

Uudelleenajo tapahtuu komennolla: `./validointi_uudelleenajo.sh` (Windows-version: `validointi_uudelleenajo.cmd`)

Tietokantaa ei saa tyhjentää uudelleenajoon koska kanta sisältää tiedon jo onnistuneesti validoiduista hakemistoista, palvelutapahtumista ja asiakirjoista.

6.5 Koontitiedoston skeema

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- PDF/A -aineiston koontitiedoston XML Schema V 1.0 17.10.2013. Kansaneläkelaitos. -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="palvelutapahtumat" type="palvelutapahtumatTyyppi"/>
  <xs:complexType name="palvelutapahtumatTyyppi">
    <xs:sequence>
<xs:element maxOccurs="unbounded" ref="palvelutapahtuma"/>
    </xs:sequence>
    <xs:attribute name="palvelujenantaja" type="oidTyyppi" use="required"/>
  </xs:complexType>
  <xs:element name="palvelutapahtuma" type="palvelutapahtumaTyyppi"/>
  <xs:element name="asiakirja" type="asiakirjaTyyppi"/>
  <xs:complexType name="asiakirjaTyyppi">
    <xs:attribute ref="id" use="required"/>
  </xs:complexType>
  <xs:complexType name="palvelutapahtumaTyyppi">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" ref="asiakirja"/>
    </xs:sequence>
    <xs:attribute ref="id" use="required"/>
    <xs:attribute name="saved" type="xs:boolean" use="optional"/>
  </xs:complexType>
  <xs:attribute name="id" type="oidTyyppi"/>
  <xs:simpleType name="oidTyyppi">
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-2](\.(0|[1-9][0-9]*)*)*"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```