



PAS-rajapinnat

PAS-koulutus 2019-10-29

Ohjelma

- 9:30-10:00 Aamukahvi
- 10:00-10:10 Tervetuloa
- 10:10-12:00 PAS-palvelun rajapintojen perusteet
- 12:00-13:00 Lounas (omakustanteinen)
- 13:00-14:00 REST-rajapinta käytännössä
- 14:00-14:30 Kahvitauko
- 14:30-16:00 REST-rajapinta käytännössä



PAS-palvelun rajapintojen perusteet



Määritykset saatavilla

Myös englanniksi

- <http://digitalpreservation.fi/specifications>
 - Aineistojen ja niiden metatietojen paketointi pitkäaikaissäilytykseen (v.1.7.1)
 - Säilytys- ja siirtokelpoiiset tiedostomuodot (v.1.7.0)
 - **PAS-palveluiden rajapinnat (v.2.1.0)**
- Ohjelmistokoodia @GitHub
 - <https://github.com/Digital-Preservation-Finland/>



PAS-rajapinnat perustuvat Asiakas-Palvelin-malliin

- Asiakas (client)
 - Järjestelmä, joka lähettää pyyntöjä palvelimelle
 - Ei ole jatkuvasti yhdistettyä palvelimeen
 - Resurssia muuttavassa pyynnössä pollaaminen, ei jatkuva kuunteleminen
 - (jatkuva kuuntelu olisi palvelu, joka aktiivisesti olisi aina valmis vastaanottamaan pyyntöjä, ja johon kohdistuisi merkittäviä vaatimuksia mm. turvallisuudesta ja toimintavarmuudesta)
 - Muutokset palvelimen koodiin eivät yleensä vaikuta asiakkaiden rajapintoihin
- Palvelin (server)
 - Järjestelmä, joka sisältää resursseja, esim. tietokannan ja aineistoa
 - Kuuntelee pyyntöjä, vastaa niihin
- Keskustelu tapahtuu rajapinnan vaatimuksien määräämällä tavalla

Asiakas-Palvelin-rajapinnat

- Palvelin ei voi lähettää omin pään viestejä asiakkaalle suoraan, koska **asiakas ei tarjoa jatkuvasti toimivaa ja turvallista palvelua**, joka ottaisi viestejä vastaan
 - Palvelin pystyy kyllä vastaamaan asiakkaan kysymykseen
 - Vain asiakas voi kysyä ja jäädä odottamaan vastausta, kunnes joko saa sen, tai kunnes odotus tai yhteys katkaistaan
- Asiakas voi keskeyttää asioinnin koska tahansa
 - Mikään ei saa mennä rajapinnassa rikki, vaikka istunto katkeaisi täysin odottamattomasti
- Palvelun toteuttaminen tuottaa aina kustannuksia mm. ylläpidon, aikavaatimusten ja turvallisuuden suhteen

Kauan kestävät prosessoinnit – Pollaus

- Joskus viestit voivat muuttaa jotakin palvelimella niin, että muutostyö kestää kauan
 - Vastausta muutoksen onnistumisesta ei voida suoraan toimittaa asiakkaalle sen jälkeen, kun kysely ehtii aikakatketa.
 - vrt. jäätelökioskilla valmistettua jäätelöä ei voida suoraan toimittaa asiakkaalle, jos asiakas ei enää ole tiskillä.
 - Viesti voidaan kuitenkin toimittaa asiakkaan saataville niin, että asiakaspää pollaa (seuraa) valmistumista omalla aikataulullaan (seuraaminen saa olla jatkuvaa, mutta ei tarvitse olla).
 - Pollaus on hyvin helppo automatisoida:

```
while jäätelö.status == keskeneräinen:  
    odota(3_minuuttia)  
    jäätelö = ota_jäätelö(jäätelön_tunniste)  
    makunautinto = onko_hyvää(jäätelö)
```

- (Lähes) kaikki rajapinnat maailmassa toimivat vastaavassa tilanteessa samalla tavalla
 - Sähköpostiohjelma pollaa uudet viestit palvelimella olevasta sähköpostilaatikosta
 - Sosiaalisen median sovellus pollaa uudet viestivirtaan tulleet viestit some-kanavalta (palvelimelta)
 - Uutissovellukset pollaavat uudet uutiset uutislähteistä (palvelimilta)

SFTP-rajapinta



SFTP

- SFTP – SSH File Transfer Protocol
- SSH-salauksella tunneloitu tiedostonsiirtoprotokolla
- Historiaa:
 - 1971: Ensimmäinen (salaamaton) FTP:n kaltainen versio
 - 1980: Salaamattoman FTP:n ensimmäinen RFC-määritys
 - 2001: Ensimmäinen SFTP:n RFC-määritys
 - 2006: Viimeisin SFTP:n RFC-määritys
 - Toki SSH-määritystä on päivitetty tämän jälkeenkin
- Edelleen tunnettu ja käytetty tiedostonsiirtoprotokolla

PAS-palvelun SFTP-rajapinta: Hakemistorakenne

PAS-rajapinnassa hyödyntäväällä organisaatiolla on kotihakemisto, jossa ovat seuraavat päähakemistot:

- ./transfer Hakemisto, jonne asiakas vie siirtopaketit.
- ./accepted Hakemisto, jonne säilytykseen hyväksyttyjen siirtopakettien tarkastusraportit viedään.
- ./rejected Hakemisto, jonne siirrossa hylättyt paketit sekä niiden tarkastusraportit viedään.
- ./disseminated Hakemisto, johon jakelupaketit syntyvät.

SFTP-rajapinta: Käyttäjätunnuksen valmistelut



- Hyödyntävä organisaatio luo käyttäjätunnukselleen erillisen ohjeen mukaan RSA-avainparin, jossa on julkinen ja yksityinen avain
 - Tunnusella pystyy kirjautumaan vain avainparia hyödyntämällä
 - Avainpari tehdään (tavallisesti) vain kerran
 - Tunnusella pystyy käsittelemään vain omaa aineistoaan
 - Aineiston siirto sopimustunnisteella, johon ei ole oikeuksia, johtaa paketin hylkäämiseen
- Julkinen avain toimitetaan PAS-palvelulle, yksityinen tallennetaan PAS-palvelua hyödyntävään järjestelmään
 - **Älä luovuta yksityistä avainta kenellekään!**
 - Jos yksityinen avain esimerkiksi häviää, on koko avainpari luotava uudelleen ja toimitettava sen julkinen avain PAS-palvelulle
- Ohjeet avainparin luomiseksi:
<http://digitalpreservation.fi/specifications/ohjeita-kayttoonotosta>
- Tarvittaessa käytöönnotoissa opastetaan tästä erikseen

SFTP-rajapinta: Autentikointi

- Autentikointi = käyttäjän tunnistautuminen
- PAS-palveluiden SFTP-rajapinnassa käytetään SSH-suojausta
 - Vain rekisteröityneille käyttäjille on pääsy SFTP-rajapintaan
 - IP-rajattu pääsy
 - Vaaditaan käyttäjätunnus ja avainpari

SFTP: Luodun siirtopaketin vienti PAS-palveluun

- Avataan yhteys transfer-hakemistoon:

- sftp -oIdentityFile=~/ssh/avain tunnus@palvelin:transfer
 - Autentikointitiedot voi konfiguroida SSH-konfiguraatioon oletuksiksi, jolloin ne voi jättää yhteyden avaamisessa pois: sftp palvelin:transfer

- Viedään siirtopaketti transfer-hakemistoon:

- put siirtopaketti.tar

- PAS-palvelu kuuntelee ./transfer-hakemistoa

- Paketti häviää hakemistosta automaattisesti, kun se otetaan käsitteilyyn.
- **Paketti ei vielä tällöin ole päätynyt säilytykseen, vaan tarkastukseen.**

SFTP-esimerkki: Usean siirtopaketin vienti

```
#!/bin/bash

FILES=./siirtopaketit/*.tar           # Hakemiston kaikki .-tar-tiedostot

(for paketti in $FILES
do
    echo "put $paketti"               # Jokaiselle tiedostolle
    sleep 10                          # Vie siirtopaketti
done) | sftp palvelin:transfer       # Esimerkiksi tauko tarvittaessa
                                      # Välitetään SFTP-rajapintaan
```

*Syntaksi voi poiketa eri alustoilla.
Virheenkäsitteily huomioiden lienee mukavampi tehdä esim. sopivalla Python-kirjastolla.*

Tarkastusraportit

- Jokaisesta tarkastetusta siirtopaketista muodostuu tarkastusraportti
- Aineisto on säilytyksessä, kun siitä muodostuu raportti ./accepted-hakemistoon
 - Säilytysvastuu on siirtynyt PAS-palvelulle raportissa mainitussa ajanhetkessä
- Siirtopaketti on hylätty, kun siitä muodostuu raportti ./rejected-hakemistoon
 - Myös siirtopaketti löytyy ./rejected-hakemistosta
- Siirtopaketin käsittely on kesken, jos raporttia ei ole muodostunut.
- Voidaan noutaa myös REST-rajapinnasta

Tarkastusraportteja on kahdenlaisia

- XML-muotoinen koneluettava raportti
 - Kertoo, mitkä tarkastusvaiheet onnistuivat ja mitkä epäonnistuivat
 - Kertoo, miin objektiin mikäkin validointi kohdistui
 - Kertoo virheet tarkkaan
 - Perustuu PREMIS-standardiin, jolla myös mm. tapahtumahistoria luodaan siirtopaketteihin
- HTML-muotoinen ihmisieluettava raportti
 - Kertoo, mitkä tarkastusvaiheet onnistuivat ja mitkä epäonnistuivat
 - Kertoo, miin objektiin mikäkin validointi kohdistui
 - Ei kerro virhettä tarkkaan

SFTP-rajapinta: Raporttien sijainti

- Hyväksytty: *./accepted/<date>/<transfer>/<transfer-id>-ingest-report.xml* (ja *.html*)
- Hylätty: *./rejected/<date>/<transfer>/<transfer-id>-ingest-report.xml* (ja *.html*)
- Siirrettiin ja hyväksyttiin siirtopaketit **training-sip1.tar** ja **training-sip2.tar**:

./accepted/2019-10-29/:

training-sip1.tar:

training-sip1.tar-2306c82a-4041-42a6-8b75-d590106fefad-ingest-report.html

training-sip1.tar-2306c82a-4041-42a6-8b75-d590106fefad-ingest-report.xml

training-sip2.tar:

training-sip2.tar-61fc8262-0216-43db-9508-534845e685ff-ingest-report.html

training-sip2.tar-61fc8262-0216-43db-9508-534845e685ff-ingest-report.xml

Ihmisluettava HTML-raportti

Digital signature signature.sig

Event name	Outcome
Submission information package digital signature validation	success

Archival information package aineisto_28718-b09c6e37-e1ae-42d9-80ec-dba09b660981.tar-5c5ef23f-95e0-47bf-9f71-c8217e85f61d.tar

Event name	Outcome
Creation of archival information package	success
Preservation responsibility change to the digital preservation service	success

Submission information package /home/tpas/transfer/aineisto_28718-b09c6e37-e1ae-42d9-80ec-dba09b660981.tar

Event name	Outcome
Virus check of transferred files	success
Fixity check of digital objects in submission information package	success
Validation of service contract properties	success
Decompression of submission information package	success

Metadata object mets.xml

Event name	Outcome
METS schema validation	success
Additional METS validation of required features	success

Ihmisluettava HTML-raportti

Digital signature `signature.sig`

Event name	Outcome
Submission information package digital signature validation	Success

Metadata object `mets.xml`

Event name	Outcome
Additional METS validation of required features	Success
Additional METS validation of required features	Failure
METS schema validation	Success

XML on rakenteista tekstiä

- XML-kielessä on elementtejä. Elementti on tietokenttä, jonka sisältämä arvo ympäröidään elementin nimellä: <etunimi>Juha</etunimi>
- Elementti voi myös sisältää elementtejä ja attribuutteja
 - Attribuutti on elementin sisällä oleva tietokenttä (yleensä elementtiä koskeva lisätieto), joka on avain-arvo-pari erotettuna yhtäsuuruusmerkillä: <tarvike nro="1">
- <henkilo>

```
<etunimi>Juha</etunimi>
<sukunimi>Lehtonen</sukunimi>
<tarvikkeet>
  <tarvike nro="1">tietokone</tarvike>
  <tarvike nro="2">hiiri</tarvike>
  <tarvike nro="3">kännykkä</tarvike>
</tarvikkeet>
</henkilo>
```

Koneluettava PREMIS XML –raportti

- Raportissa on PREMIS-määritysten mukaisesti objekteja, tapahtumia ja agentteja
- Tapahtumista selviävät varsinaiset tarkastusvaiheet, niiden aikaleimat, lopputulokset ja tarkastuksen tuloste
- Tapahtumista on linkit tarkastuksen kohteeseen (objektiin) ja suorittajaan (agenttiin)
- Objektit viittaavat tarkastuksen kohteeseen: Ne siis ovat viittausobjekteja, jotka viittaavat esimerkiksi siirtopaketin tiedostoon, itse siirtopakettiin tai luotuun säilytyspakettiin
- Agentit ovat lyhyitä kuvauskia tapahtuman suorittajasta

Tarkastusraportissa kuvatut objektit

- Siirtopaketti
- Sähköinen allekirjoitus
- METS-dokumentti
- Dигитальный объект
- Säilytyspaketti

Tarkastusraportissa kuvatut tapahtumat

- Siirtopaketin siirto
- Siirtopaketin purku
- Virustarkastus
- Dигитaalisen allekirjoituksen tarkastus
- Sopimuksen tai säilytyspäätöksen tarkastus
- METS-skeemavalidointi
- METS-lisävalidointi
- Digitaalisten objektiien eheyystarkastus
- Digitaalisten objektiien validointi
- Siirtopaketin tarkastuksen yhteenvetö
- Säilytyspaketin luominen
- Säilytysvastuun siirtyminen

Tarkastusraportissa kuvatut agentit

- Aineiston siirtänyt käyttäjä
- PAS-ohjelmistokomponentti (useita)

Määritysdokumentti vs. Tarkastusraportti – Esimerkki Agentista

Elementti	Määritys
<agent>	
-<agentIdentifier>	Agentin tunniste
--<agentIdentifierType>	“preservation-agent-id” (tunnisteen tyyppi)
--<agentIdentifierValue>	Tunnisteen arvo
-<agentName>	Ohjelmakoodin nimi
-<agentType>	“software” (agentin tyyppi)

```

<premis:agent>
  <premis:agentIdentifier>
    <premis:agentIdentifierType>preservation-agent-id</premis:agentIdentifierType>
    <premis:agentIdentifierValue>preservation-agent-check_sip_digital_objects.py-v0.0-d259624d-c712-41e1-a3b7-e879351797af</premis:agentIdentifierValue>
  </premis:agentIdentifier>
  <premis:agentName>check_sip_digital_objects.py-v0.0</premis:agentName>
  <premis:agentType>software</premis:agentType>
</premis:agent>

```

Siirtopaketti tarkastusraportissa

Elementti	Määritys
<object>	
@xsi:type	"representation" (PREMIS-objektiin tyyppi)
-<objectIdentifier>	Siirtopaketin tunniste tarkastusraportissa
--<objectIdentifierType>	"preservation-sip-id" (tunnisten tyyppi)
--<objectIdentifierValue>	Tunnisten arvo
-<originalName>	ZIP- tai TAR-tiedoston nimi, jossa siirtopaketti alkuaan sijaitsi
-<environment>	
--<dependency>	Riippuvuussuhde siirtopakettiin
---<dependencyIdentifier>	Siirtopaketin tunniste METS-dokumentissa
----<dependencyIdentifierType>	"mets:OBJID"
----<dependencyIdentifierValue>	Siirtopaketin tunnisten arvo
--<dependency>	Riippuvuussuhde sopimustunnisteeseen
---<dependencyIdentifier>	Sopimustunniste METS-dokumentissa
----<dependencyIdentifierType>	preservation-contract-id
----<dependencyIdentifierValue>	Sopimustunnisten arvo

Tapahtumat tarkastusraportissa (tapahtumarunko)



Elementti	Määritys
<event>	
-<eventIdentifier>	Tapahtuman tunniste
--<eventIdentifierType>	“preservation-event-id” (tunnisten tyyppi)
--<eventIdentifierValue>	Tunnisten arvo
-<eventType>	Tapahtuman tyyppi, ks. tarkennus tapahtumista
-<eventDateTime>	Tapahtuman aikaleima
-<eventDetail>	Tapahtuman nimi, ks. tarkennus tapahtumista
-<eventOutcomeInformation>	
--<eventOutcome>	Tapahtuman lopputulos: “success” tai “failure”
--<eventOutcomeDetail>	
--<eventOutcomeDetailNote>	Tekstimuotoinen lisätieto tapahtuman lopputuloksesta
--<eventOutcomeDetailExtension>	XML-muotoinen lisätieto tapahtuman lopputuloksesta
-<linkingAgentIdentifier>	Viittaus agenttiin tarkastusraportissa, ks. tarkennus tapahtumista
--<linkingAgentIdentifierType>	Agentin tunnisteen tyyppi
--<linkingAgentIdentifierValue>	Agentin tunnisteen arvo
-<linkingObjectIdentifier>	Viittaus objektiin tarkastusraportissa, ks. tarkennus tapahtumista
--<linkingObjectIdentifierType>	Objektiin tunnisteen tyyppi
--<linkingObjectIdentifierValue>	Objektiin tunnisteen arvo

Täsmennykset tapahtumarunkoon: Esimerkkejä

Siirtopaketin siirto (täsmennys tapahtumarunkoon)

Elementti	Määritys
-<eventType>	"transfer"
-<eventDetail>	"Transfer of submission information package"
-<linkingAgentIdentifier>	Viittaus aineiston siirtäneeseen käyttäjään
-<linkingObjectIdentifier>	Viittaus siirtopakettiin

Digitaalisten objektien validointi (täsmennys tapahtumarunkoon)

Elementti	Määritys
-<eventType>	"validation"
-<eventDetail>	"Digital object validation"
-<linkingAgentIdentifier>	Viittaus validointisovellukseen
-<linkingObjectIdentifier>	Viittaus digitaaliseen objektiin

(... ja niin edelleen kaikille tapahtumille...)

Agentit tarkastusraportissa

Aineiston siirtänyt käyttäjä:

Elementti	Määritys
<agent>	
-<agentIdentifier>	Agentin tunniste
--<agentIdentifierType>	“preservation-user-id” (tunnisten tyyppi)
--<agentIdentifierValue>	Tunnisten arvo
-<agentName>	Aineiston siirtäjän käyttäjätunnus
-<agentType>	“organization” (agentin tyyppi)

PAS-ohjelmistokomponentti:

Elementti	Määritys
<agent>	
-<agentIdentifier>	Agentin tunniste
--<agentIdentifierType>	“preservation-agent-id” (tunnisten tyyppi)
--<agentIdentifierValue>	Tunnisten arvo
-<agentName>	Ohjelmakoodin nimi
-<agentType>	“software” (agentin tyyppi)

Esimerkki: Digitaalisen objektiin validointi raportissa



```
<premis:event>
  <premis:eventIdentifier>
    <premis:eventIdentifierType>preservation-event-id</premis:eventIdentifierType>
    <premis:eventIdentifierValue>7a9cdcba-186a-4231-bdfa-8cdc83a549ef</premis:eventIdentifierValue>
  </premis:eventIdentifier>
  <premis:eventType>validation</premis:eventType> <- Tyyppi
  <premis:eventDateTime>2019-10-22T13:33:04.164262</premis:eventDateTime> <- Aikaleima
  <premis:eventDetail>Digital object validation</premis:eventDetail> <- Kuvaus
  <premis:eventOutcomeInformation>
    <premis:eventOutcome>success</premis:eventOutcome> <- Lopputulos (success/failure)
    <premis:eventOutcomeDetail>
      <premis:eventOutcomeDetailNote>Detected mimetype: text/plain, version: None
METS mimetype: text/plain
Validating as mimetype: text/plain
[PredefinedDetector] User-supplied file format used
[MagicScraper] MIME type not scraped, using user-supplied value. Lisätietoa tuloksesta (validoinnin tuloste)
[MagicScraper] The file was analyzed successfully.
[JHoveUtf8Scraper] Well-Formed and valid
[MetadataComparator] METS metadata matches scraper metadata.
The digital object will be preserved as mimetype: text/plain</premis:eventOutcomeDetailNote>
</premis:eventOutcomeDetail>
jatkuu . . .
```

Esimerkki: Dигиталisen objektiin validointi raportissa

...jatkoa

```
</premis:eventOutcomeInformation>
<premis:linkingAgentIdentifier> Viittaus agenttiin
  <premis:linkingAgentIdentifierType>preservation-agent-id</premis:linkingAgentIdentifierType>
  <premis:linkingAgentIdentifierValue>preservation-agent-check_sip_digital_objects.py-v1</premis:linkingAgentIdentifierValue>
</premis:linkingAgentIdentifier>
<premis:linkingObjectIdentifier> Viittaus objektiin
  <premis:linkingObjectIdentifierType>preservation-object-id</premis:linkingObjectIdentifierType>
  <premis:linkingObjectIdentifierValue>6a5a0250-264f-40dd-95f2-7c958bd0c0fe</premis:linkingObjectIdentifierValue>
</premis:linkingObjectIdentifier>
</premis:event>
```

Esimerkki: Dигитални объект в отчете

```
<premis:object xsi:type="premis:representation">
  <premis:objectIdentifier>
    <premis:objectIdentifierType>preservation-object-id</premis:objectIdentifierType>
    <premis:objectIdentifierValue>6a5a0250-264f-40dd-95f2-7c958bd0c0fe</premis:objectIdentifierValue>
  </premis:objectIdentifier> ^Tunniste
  <premis:originalName>contents/Machine-readable metadata/metadata.txt</premis:originalName>
  <premis:environment>
    <premis:dependency>
      <premis:dependencyIdentifier>
        <premis:dependencyIdentifierType>UUID</premis:dependencyIdentifierType>
        <premis:dependencyIdentifierValue>6cbf6da5-ef63-404f-abad-7b0486f5c7e5</premis:dependencyIdentifierValue>
      </premis:dependencyIdentifier> ^ Tiedostonimi
    </premis:dependency>
  </premis:environment>
  <premis:relationship>
    <premis:relationshipType>structural</premis:relationshipType>
    <premis:relationshipSubType>is included in</premis:relationshipSubType> <- Sisältyy...
    <premis:relatedObjectIdentification>
      <premis:relatedObjectIdentifierType>preservation-sip-id</premis:relatedObjectIdentifierType>
      <premis:relatedObjectIdentifierValue>2306c82a-4041-42a6-8b75-d590106fefad</premis:relatedObjectIdentifierValue>
    </premis:relatedObjectIdentification> ... ^ Viittaus отчету объекту,
  </premis:relationship> который описывает SIP:н
</premis:object>
```

Esimerkki: Agentti raportissa

```
<premis:agent>
  <premis:agentIdentifier>
    <premis:agentIdentifierType>preservation-agent-id</premis:agentIdentifierType>
    <premis:agentIdentifierValue>preservation-agent-check_sip_digital_objects.py-v0.0-d259624d-c712-41e1-a3b7-e879351797af</premis:agentIdentifierValue> ^ Tunniste
  </premis:agentIdentifier>
  <premis:agentName>check_sip_digital_objects.py-v0.0</premis:agentName>
  <premis:agentType>software</premis:agentType> ^ Nimi
</premis:agent> ^ Tyyppi
```

SFTP-esimerkki: Edellisen päivän tarkastusraporttien nouto

```
#!/bin/bash
yesterday=$(date -d "yesterday" '+%Y-%m-%d')          # Esim. "2019-10-28"

mkdir "./accepted/$yesterday"                         # Luo paikallisesti pvm-hakemistot
mkdir "./rejected/$yesterday"

(echo "lcd ./accepted/$yesterday"                      # Siirry luotuun hakemistoon
 echo "cd accepted"                                    # Siirry palvelimella accepted-hakemistoon
 echo "get $yesterday/*/*.xml"                         # Nouda raportit palvelimen pvm-hakemistosta

echo "lcd ../../rejected/$yesterday"      # Tee sama rejected-hakemistossa
echo "cd ../../rejected"
echo "get $yesterday/*/*.xml") | sftp palvelin
```

*Syntaksi voi poiketa eri alustoilla.
Virheenkäsitteily huomioiden lienee mukavampi tehdä esim. sopivalla Python-kirjastolla.*

SFTP-rajapinta: Luodun jakelupaketin nouto PAS-palvelusta

- Avataan yhteys disseminated-hakemistoon:
➤ sftp palvelin:disseminated
- Haetaan jakelupaketti disseminated-hakemistosta:
➤ get dip-tunniste/dip-tunniste.zip
- Jakelupaketin tunnisten (dip-tunniste) saa jakelupaketin luonnissa tulleesta vastauksesta
 - Jakelupaketti luodaan REST-rajapinnassa
- Jakelupaketin voi noutaa myös REST-rajapintaa hyödyntämällä
- REST-rajapintaa harjoitellaan iltapäivällä

REST-rajapinta



REST-rajapinta

- REST = Representational state transfer
- Arkkitehtuurimalli, perustuu HTTP-protokollaan
 - Pyyntö - Vastaus
 - Resurssipohjainen, resurssit määritellään URI:na pyynnöissä
 - Hyperlinkkejä viesteissä
 - Vakiintuneita metodeita (GET, POST, PUT, DELETE)
- REST määrittelee ominaisuuksia rajapinnoille, jotka ovat "RESTful"

Historiaa

- Vuonna 2006 Google ilmoitti SOAP-rajapintansa vanhentuneeksi ja siirtyi REST:iin
 - Maailma seurasi perässä.
 - Syy muutokseen oli, että XML-ratkaisut koettiin kaikkien kannalta liian työläiksi (ja siten kalliiksi):
 - SOAP XML:ssä mikä tahansa pienikin muutos vaatii kaikilta osapuolilta korjauksia koodiin.
 - REST mahdollistaa valmiin ja laajan automatiikan mukaan ottamisen, kun se rakennetaan HTTP(S)-protokollan päälle: sisäänrakennettu virheenkäsittely, välimuisti, kerrostettu arkkitehtuuri...
 - Vuonna 2009 Googlen SOAP-rajapinnat suljettiin lopullisesti.
- Tänä päivänä 80% kaikista maailman aktiivisena olevista koneluettavista julkisista verkkorajapinnoista on toteutettu REST:llä
 - Osuus kasvaa koko ajan.

REST

- REST on suosittu ja yleisesti käytössä koska:
 - Sitä on helppoa kehittää, koska se perustuu olemassa olevaan protokollaan, HTTP
 - Se ei vaadi asiakasjärjestelmiltä erillistä jatkuvasti ylläpidettävää softaohjelmaa
 - HTTP-metodit (GET, POST jne.) ovat yksiselitteisiä
 - REST on joustava, koska jokainen resurssi on yksilöity
 - REST skaalautuu hyvin ja on tehokas viestinnässä koska sisältö ei ole riippuvainen käytetystä taustajärjestelmien teknologiasta
 - REST-rajapinta on helposti muokattavissa ja laajennettavissa

REST-yhtenäisen rajapinnan vaatimus

- REST on resurssipohjainen
 - Vs. toimintapohjainen
 - Pyyntöjen kohde on resurssi, esimerkiksi tietokanta, jota käsitellään HTTP-komennoilla
 - Pyynnöt koostuvat URI-merkkijonoista, jotka yksilöivät resurssin
- REST-viestit kuvaavat itseään
 - esim. MIME-tyyppiillä viestin otsikossa

REST: Yhtenäisen rajapinnan vaatimus

- Dataa käsitellään representaationa
 - Esimerkiksi metatietoa JSON-muodossa
 - Ei riippuvuutta taustajärjestelmän teknologiasta
- REST-viestit ovat hypertekstiä
 - Viestit sisältävät otsikon, rungon, kyselyn, palautuskoodin, hyperlinkkejä
- Esimerkki: GET <http://myrest/people/juhalehtonen>

RESTful-rajapinta on tilaton

- Tilattomassa arkkitehtuurissa palvelin ei tallenna dataa istunnoista
 - Pyyntö on itsenäinen ja tuottaa aina saman tuloksen
 - Pyynnön tulos ei ole riippuvainen edellisistä pyynnöistä
 - Asiakas hallinnoi tilaa, haluttu tila ilmaistaan viesteissä
- Jokainen viesti sisältää kaiken tarvittavan tiedon mitä palvelin tarvitsee voidakseen käsittää viestin
 - Ei riippuvuutta ulkoiseen tietoon
- Esimerkiksi: GET <http://myrest/people/juhalehtonen/email>

Eli:

- Asiakkaat lähettävät REST-rajapintaan kyselyitä, joihin palvelin vastaa
- Vastaukset johtavat asiakkaita eteenpäin
 - Vastaukset sisältävät linkkejä eteenpäin, tai aineistoa
 - Asiakkaan ei tarvitse tietää rajapinnan kaikkia ominaisuuksia etukäteen
- Pitkään kestävä resurssin muutos, esimerkiksi jakelupaketin muodostaminen, selvitetään pollaamalla palvelinta
 - Asiakkaan ei tarvitse toteuttaa kuuntelupalvelua PAS-palvelimelle
 - Omaa palvelua merkittävine ylläpito- yms. vastuineen ei tarvita
 - Asiakas on aina aloitteentekijä keskustelussa
 - PAS-palvelun REST-rajapinnassa jakelupaketin muodostaminen on toistaiseksi ainoa toiminto, jota tarvitsee pollata

Hyötyjä



- Kaikki tarvittavat tiedot resurssien muokkaamiseen ja palauttamiseen sisältyvät viesteihin
- Pyyntö ja vastaus liittyvät automaattisesti toisiinsa
- Viestit sisältävät valmiiksi metatietoja otsikossa HTTP-protokollan myötä, kuten keskustelun osapuolet HTTPS-yhteydessä
- Mahdollistaa sisäänrakennetun virheenkäsittelyn
 - Jos jokin ei toimi, tavallisesti riittää yrittää uudestaan
 - Mitään ei tarvitse ”resetoida”, jos jokin keskeytyy
 - Pyynnöt ja pollaamiset eivät riko rajapintaa eivätkä vaikuta suorituskykyyn
- Yleisesti tunnetuissa ohjelmointikielissä on tarvittavat kirjastot valmiina
- PAS-palveluiden rajapinnat perustuvat GET-pyyntöihin, jotka eivät muokkaa resurseja
 - Poikkeuksena jakelupaketin luontipyyntö, joka on POST

HTTP-viestit



HTTP-metodit

- CRUD-pohjaisia metodeita resurssien hallintaan
 - CRUD = Create, Read, Update, Delete
- GET – resurssin haku; harmiton metodi, koska se ei muuta resurssia
- POST – tietojen lähetäminen tai päivittäminen palveluun
- Lisäksi DELETE, PUT, HEAD, OPTIONS jne.

HTTP-pyyntöviestien rakenne

- Pyyntö kohdistuu resurssiin
 - Metodi ja URL
- Otsake kuvaaa viestiä
 - Sisältää tietoja pyyntöviestistä, kuten viestin sisällön tyyppi, koko, aikaleima
 - Tiedot ilmaistaan avain-arvo-parina
- Tyhjä rivi
- Runko sisältää tarvittavan datan
 - Voidaan käyttää esim. PUT-pyynnössä
 - GET-pyyntö ei sisällä viestirunkoa
- Esimerkkejä:
 - GET /example.html HTTP/1.1
Date: Mon, 13 Nov 2017 08:12:31 GMT
Host: localhost

```
PUT /newpage.html HTTP/1.1
Host: localhost
Content-type: text/html
Content-length: 16

<p>New Page</p>
```

HTTP-vastausviestien rakenne

- Statusrivi ilmaisee tuloksen
 - Palautuskoodi
- Otsake kuvaaa viestiä
 - Sisältää tietoja viestistä kuten viestin sisällön tyyppi, koko, aikaleima
 - Tiedot ilmaistaan avain-arvo-parina
- Tyhjä rivi
- Runko sisältää palautusviestin
 - Yleisiä formaatteja on JSON, HTML, XML
 - Voi joissakin tilanteissa olla myös tyhjä

HTTP-vastausviestien rakenne, esimerkki

HTTP/1.1 200 OK

Date: Mon, 23 May 2005 22:38:34 GMT

Content-Type: text/html; charset=UTF-8

Content-Length: 138

Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT

Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)

ETag: "3f80f-1b6-3e1cb03b"

Accept-Ranges: bytes

Connection: close

```
<html>
  <head>
    <title>An Example Page</title>
  </head>
  <body>
    <p>Hello World, this is a very simple HTML document.</p>
  </body>
</html>
```

HTTP – yleisiä palautuskoodeja

- Viestit sisältävät kolminumeroisen koodin, joka kertoo miten pyyntö on vastaanotettu ja käsitelty
 - 1xx-alkuiset koodit – palvelin on vastaanottanut viestin
 - 2xx-alkuiset koodit – onnistunut pyyntö
 - 200 – OK, 202 – hyväksytty
 - 3xx-alkuiset koodit – edelleenohjaus
 - 4xx-alkuiset koodit – virhetilanne
 - 400 – vääränlainen pyyntö, 401 – ei auktorisoitu pyyntö, 404 – resurssia ei löydy, 405 – väärä menetelmä
 - 5xx-alkuiset koodit – palvelinvirhe (palvelu tai sen osa on alhaalla)

PAS-palveluiden rajapinnan palautusviestit ovat JSON-muodossa

- JSON on kompakti ja avoin standardi tiedonvälitykseen
- Syntaksi perustuu avain-arvo-pareille
 - Tietueet erotellaan aaltosulkeilla
 - Avain-arvo-parit erotellaan pilkulla
 - Listat ympäröidään hakasulkeilla
- Esimerkki:

```
{  
    "etunimi": "Juha",  
    "sukunimi": "Lehtonen",  
    "tarvikkeet": ["tietokone", "hiiri", "känykkä"]  
}
```

Jäätelökioski vs. HTTP-viestit

- Jussi näkee torilla avoimen kioskin ja saapuu tiskille
 - Jussi: Anteeksi, minkälainen kioski tämä on?
 - Asiakas: *GET https://kiosk*
 - Myyjä: Tämä on jäätelökioski nimeltään "Liisan Jätski". Myymme täällä jäätelöitä.
 - Palvelin: *Status 200 ... {"name": "Liisan Jätski", "type": "ice-creams", "seemenu": "http://kiosk/icecreams"}*
 - Jussi : Minkälaisia jäätelöitä täältä saa?
 - Asiakas: *GET https://kiosk/icecreams*
 - Myyjä: Meiltä saa mansikka- ja suklaajäätelöitä.
 - Palvelin: *Status 200 ... {"icecreams": ["https://kiosk/icecreams/strawberry", "https://kiosk/icecreams/chocolate"]}*
 - Jussi : Mansikkajäätelö, kiitos.
 - Asiakas: *GET https://kiosk/icecreams/strawberry*
 - Myyjä: Tässä ole hyvä, se maksaa kolme euroa.
 - Palvelin: *Status 200 ... {"icecream": ..., "payment": "https://kiosk/payment?id=id123&euros=3"}*
 - Jussi : Tässä kolme euroa.
 - Asiakas: *POST https://kiosk/payment?id=id123&euros=3*
 - Myyjä: Kiitos. Tässä vielä kuitti.
 - Palvelin: *Status 200 ... {"receipt": ...}*

Jäätelökioski vs. HTTP-viestit

- Seuraavalla kerralla Jussi jo tietää kioskin olevan jäätelökioski ja myös valikoiman, joten tämä voi suoraan pyytää haluamansa jäätelön. Hän kuitenkin pohtii, voisiko sieltä ostaa muutakin, ja kokeilee ostaa myös sohvaan:
 - Jussi : Mansikkajäätelö, kiitos.
 - Asiakas: *GET https://kiosk/icecreams/strawberry*
 - Myyjä: Tässä ole hyvä, se maksaa kolme euroa.
 - Palvelin: *Status 200 ... {"icecream": ..., "payment": "https://kiosk/payment?id=id234&euros=3"}*
 - Jussi : Tässä kolme euroa.
 - Asiakas: *POST https://kiosk/payment?id=id234&euros=3*
 - Myyjä: Kiitos. Tässä vielä kuitti.
 - Palvelin: *Status 200 ... {"receipt": ...}*
 - Jussi : Saisinko vielä sohvan, kiitos.
 - Asiakas: *GET https://kiosk/furnitures/sofa*
 - Myyjä: Myymme vain jäätelöä.
 - Palvelin: *Status 400 ... {"message": "Only ice-cream available."}*

Jäätelökioski vs. Pollaus

- Valikoimiin on tullut erikoisjäätelö, mutta se tehdään paikan päällä ja kioskilla on ruuhkaa. Jussi ei kiireidensä vuoksi pysty odottamaan valmistumista riittävän pitkään, joten hän poistuu paikalta muihin tehtäviin, mutta käy aina välillä kysymässä tilannetta.
 - Jussi: Haluaisin erikoisjäätelön, jossa on suklaata, vaniljakastiketta ja nonpareilleja, kiitos.
 - Asiakas: *POST https://kiosk/makespecial?chocolate=true&vanillasauce=true&nonpareils=true*
 - Myyjä: Alan tekemään jäätelöä. Jäätelön tunniste on 7.
 - Palvelin: *Status 202 ... {"location": "https://kiosk/icecreams/special?id=7"}*
(...Jussi poistuu, aikaa kuluu, Jussi tulee takaisin...)
 - Jussi: Saisinko jäätelöni tunnisteella 7, kiitos.
 - Asiakas: *GET https://kiosk/icecreams/special?id=7*
 - Myyjä: Ei ole (vielä valmis).
 - Palvelin: *Status 404*
(...Jussi poistuu, aikaa kuluu, Jussi tulee takaisin...)
 - Jussi: Saisinko jäätelöni tunnisteella 7, kiitos.
 - Asiakas: *GET https://kiosk/icecreams/special?id=7*
 - Myyjä: Tässä ole hyvä, se maksaa viisi euroa.
 - Palvelin: *Status 200 ... {"icecream": ..., "payment": "https://kiosk/payment?id=id345&euros=5"}*

PAS-palveluiden REST-rajapinta – määritykset



REST-rajapinnan toiminnot

- PAS-palveluiden REST-rajapinnan kautta asiakas voi hakea, hallinnoida ja noutaa omaa aineistoaan
 - Haku: Haun avulla asiakas löytää aineistonsa, jota hän haluaa hallinnoida
 - Hallinnointi: Asiakas pystyy muodostamaan aineistostaan jakelupaketin
 - Noutaminen: Asiakas voi ladata muodostettuja jakelupaketteja
- Lisäksi asiakas voi:
 - hakea ja ladata siirtopaketin tarkastusraportin ja
 - saada tilastotietoa

REST-rajapinnan komentoketju

- Komennot muodostavat katkeamattoman ketjun:
 - Aineiston haku palauttaa linkit säilytyspakettien hallintaan
 - Säilytyspaketin hallinta palauttaa linkin jakelupaketin muodostamiseen
 - Jakelupaketin muodostus palauttaa linkin jakelupaketin noutamiseen
- Koko ketjua ei ole tarpeen käydä läpi, mikäli tarvittava osoite on tiedossa
- Esimerkiksi jakelupaketin noutokomennon voi antaa heti, mikäli jakelupaketti on ennestään olemassa ja sen tunniste on tiedossa

PAS-palveluiden REST-rajapinnan kyselyt

- Resurssi yksilöidään pyynnön URlla, joka muodostetaan kauttaviivoilla erotetuilla määreillä
- Joillekin pyynnöille on mahdollista antaa tarkentavia parametreja avain-arvo-pareina
 - ? -merkki ilmaisee parametrien osuuden pyynnöstä
 - & -merkki käytetään parametrien erotinmerkinä
 - Esimerkki: <http://myrest/people?find=all&limit=100>

PAS-palveluiden REST-rajapintakyselyiden muoto

- PAS-palveluiden REST-rajapinnan kyselyt ovat muotoa:

{GET, POST} <base>/<contract>/<term> ...

- Kyselyissä:
 - <base> ilmaisee REST-rajapinnan
 - <contract> ilmaisee sopimustunnisteen, joka toimii kyselyiden aineiston rajausena
 - <term> ilmaisee resurssin tai työkalun

PAS-palveluiden REST-rajapinnan kyselyiden muoto

{GET, POST} base/<contract>/<term> ...

- base on URL-merkkijono, osoite, joka on ohjaa kyselyt PAS-palveluiden rajapinnalle:
 - <https://pas.csc.fi/api/2.0>
- Hyväksyttyt metodit kyselyviesteille ovat GET (useimmat kyselyt) ja POST (jakelupaketin muodostus)

Sopimustunniste

{GET, POST} <base>/<**contract**>/<term> ...

- <contract> on aineiston sopimustunniste

- Aineistoa haetaan ja hallinnoidaan REST-rajapinnassa sopimustunnisteen (ei käyttäjätunnuksen) perusteella
- Jokainen säilytyksessä oleva säilytyspaketti on osa aineistoa, jotka määritellään PAS-palvelusopimuksissa
- Sopimuksen ja hyödyntävän organisaation suhde ei ole 1:1
- Hyödyntäville organisaatioille ilmoitetaan heidän sopimuksiensa tunnisteet
- Sopimustunniste (CONTRACTID) on ollut pakollinen metatieto PAS-skeemakatalogin versiosta 1.7.0 lähtien
- Kaikille "vanhoille" säilytyspaketeille on lisätty sopimustunniste hakuindeksissä

PAS-palveluiden REST-rajapinnan resurssit

{GET, POST} <base>/<contract>/<**term**> ...

- <term> on olemassa oleva resurssi tai työkalu:
 - search – aineiston haun työkalu
 - preserved – säilytyspaketin resurssi
 - disseminated – jakelupaketin resurssi
 - ingest – siirtopaketin resurssi
 - statistics - tilastoresurssi
- Esimerkki:
 - <https://pas.csc.fi/api/2.0/urn:minunsopimukseni/search>

PAS-palveluiden REST-rajapinnan palautusviestit

```
{  
  "status": "success",  
  "data": { ... }  
}
```

- status – mahdolliset arvot: "success" ja "fail"
- data – sisältää palautusviestin

Autentikointi ja auktorisointi



REST-rajapinta on avoin vain PAS-palveluiden käyttäjille

- PAS-palveluiden REST-rajapinnassa käytetään TSL/SSL-suojausta ja HTTP Basic Access –autentikointia
 - Vain rekisteröityneille käyttäjille on pääsy REST-rajapintaan
 - IP-rajattu pääsy
 - Vaaditaan käyttäjätunnusta ja salasanaa REST-rajapinnan pyynnöille

REST-rajapinnan kautta on pääsy vain omalle aineistolle

- Auktorisointi = käyttäjän oikeuksien hallinta resursseihin
- PAS-palvelussa tarkastetaan jokaisen pyynnön yhteydessä, onko kyseisellä käyttäjällä oikeus hakea kyseisellä sopimustunnisteella aineistoa
 - Pyynnöt hyväksytään vain, jos käyttäjällä on riittäviä oikeuksia sopimustunnisteelle
 - Pyynnöissä tarkastetaan, kuuluuko haettu säilytyspaketti kyseiselle sopimustunnisteelle
- Poikkeuksena jakelupaketin julkisen allekirjoitusavaimen nouto

Aineiston hakutyökalu



Aineiston haku

GET <base>/<contract>/**search**?<parameters>

- Aineistoa pystyy hakemaan metatietotietokannasta METS-dokumentin sisällön perusteella
- Hakuindeksinä toimii Solr-tietokanta
 - Myös Finna käyttää Solria
- Tyhjä haku (ei parametreja) palauttaa luettelon kaikista kyseisen sopimustunnisteen alaisista paketeista

Aineiston haku

GET <base>/<contract>/search?<parameters>

- Pyynnölle pystyy antamaan valinnaisia parametreja, joista "q" ("query") ilmaisee hakuehdon
- Esimerkki:
 - <https://pas.csc.fi/api/2.0/mycontract/search?q=title:MyTitle>

Aineiston haku – kyselyt metatietotietokantaan

GET <base>/<contract>/search?q=XXX:XXX

- Hakuehdot ovat muotoa q=<hakukenttä>:<arvo>
 - Hakukenttä on METS-dokumentin elementin tai attribuutin nimi
 - Mahdollisuus antaa koko XML-polku haun tarkkaan rajaukseen
 - Apache Lucene-syntaksi käytettäväissä
- Esimerkkejä:
 - q=subject:food
 - q=OBJID:sip-001
 - q=mets_dmdSec_mdWrap_xmlData_subject:food
 - Myös monimutkaisemmat haut mahdollisia: AND/OR-operaatiot, jokerimerkit, jne.

Aineiston haku – vastausviesti



```
{  
  "data": {  
    "links": {  
      "self": "/api/2.0/mycontract/search?q=OBJID%3Ab82112ce-f3ea-4789-afd7-  
43651a374b0f&limit=20&page=1"  
    },  
    "results": [  
      {  
        "createdate": "2018-04-10T11:48:20Z",  
        "id": "aip-001", <- Paketin tunniste  
        "location": "/api/2.0/mycontract/preserved/aip-001",  
        "match": {  
          "mets_OBJID": [ "b82112ce-f3ea-4789-afd7-43651a374b0f" ]  
        },  
        "pkg_type": "AIP"  
      }  
    ]  
  },  
  "status": "success"  
}
```

^ Osoite aineiston hallintaan

^ Hakuehdon osuma

Säilytyspakettien resurssi



Säilytyspaketin kohdistuvat komennot

GET <base>/<contract>/**preserved**/<aip-id>

- Aineiston hakutoiminto palauttaa tietoja halutusta aineistokokonaisuudesta ja linkit sitä vastaavien säilytyspaketin hallintaan
- Säilytyspakteja hallinnoidaan preserved-termillä
 - <aip-id> on pakollinen määre, joka yksilöi säilytyspaketin
- Esimerkki:
 - <https://pas.csc.fi/api/2.0/mycontract/preserved/aip-001>

Säilytyspaketin kohdistuvat komennot

- Palautusviestissä ilmoitetaan säilytyspaketille kohdistuvat komennot linkkinä

- Tällä hetkellä vain jakelupaketin muodostus

```
{  
  "status": "success",  
  "data": {  
    "disseminate": "<base>/<contract>/preserved/<aip-id>/disseminate"  
  }  
}
```

- Esimerkki:

- "disseminate": "https://pas.csc.fi/api/2.0/mycontract/preserved/aip-001/disseminate"

Säilytyspaketin muodostus jakelupaketiksi

POST <base>/<contract>/preserved/<aip-id>/**disseminate?**<parameters>

- Säilytyspaketista muodostetaan jakelupaketti POST-komennolla
- Valinnaiset parametrit ovat:
 - catalog – haluttu skeemakatalogin versio, johon jakelupaketti muodostetaan
 - format – haluttu jakelupaketin formaatti
- Esimerkkejä:
 - <https://pas.csc.fi/api/2.0/mycontract/preserved/aip-001/disseminate>
 - [https://pas.csc.fi/api/2.0/mycontract/preserved/aip-001/disseminate?catalog="1.7.0"](https://pas.csc.fi/api/2.0/mycontract/preserved/aip-001/disseminate?catalog='1.7.0')
 - [https://pas.csc.fi/api/2.0/mycontract/preserved/aip-001/disseminate?catalog="1.7.0"&format="tar"](https://pas.csc.fi/api/2.0/mycontract/preserved/aip-001/disseminate?catalog='1.7.0'&format='tar')

Jakelupaketin muodostus

- Palautusviestissä ilmoitetaan jakelupaketin hallinnan osoite

```
{  
  "status": "success",  
  "data": {  
    "disseminated": "<base>/<contract>/disseminated/<dip-id>"  
  }  
}
```

- <dip-id> on muodostetun jakelupaketin tunniste

- Esimerkki:

- "disseminated": "https://pas.csc.fi/api/2.0/mycontract/disseminated/dip-001"

Jakelupaketit



Jakelupaketin seuranta ja niihin kohdistuvat komennot

GET <base>/<contract>/**disseminated**/<dip-id>

- Komenolla seurataan jakelupaketin muodostusta
- Palautusviestissä ilmoitetaan jakelupaketin muodostuksen status sekä jakelupakettiin kohdistuvat komennot
- Komento löytyy tuloksena edeltävästä jakelupaketin muodostus-komennosta
- Esimerkki:
 - <https://pas.csc.fi/api/2.0/mycontract/disseminated/dip-001>

Jakelupaketin muodostamisen seuranta

- Palautusviesti on seuraava

```
{  
  "status": "success",  
  "data": {  
    "complete": <complete>,  
    "actions": <actions>  
  }  
}
```

- *complete* ilmaisee jakelupaketin muodostumisen statusta, arvo voi olla joko "true" (jakelupaketti on muodostettu) tai "false" (jakelupaketin muodostus on käynnissä)
- *actions* ilmaisee jakelupaketille kohdistuvat komennot
- Mikäli jakelupaketin muodostus on kesken, *actions* on tyhjä {}

Jakelupakettiin kohdistuvat komennot

- Muodostuneen jakelupaketin palautusviesti on seuraava

```
{  
  "status": "success",  
  "data": {  
    "complete": "true",  
    "actions": {  
      "download": "<base>/<contract>/disseminated/<dip-id>/download"  
      "metadata": "<base>/<contract>/disseminated/<dip-id>/metadata"  
      "history": "<base>/<contract>/disseminated/<dip-id>/history"  
    }  
  }  
}
```

Jakelupaketin noutaminen

GET <base>/<contract>/disseminated/<dip-id>/**download**

- Komentolla noudetaan jakelupaketti
- Jakelupaketti annetaan palautusviestissä
- Muita komentoja on metatietojen ja tapahtumahistorian noutaminen
- Esimerkki:
 - <https://pas.csc.fi/api/2.0/mycontract/disseminated/dip-001/download>

Allekirjoitusavaimen noutaminen

GET <base>/**public_key/dip**

- Komennolla noudetaan jakelupaketin allekirjoitusavaimen julkinen osa
- Avaintiedosto annetaan palautusviestissä

Siirtopaketin tarkastusraporttien haku

GET <base>/<contract>/**ingest/report**/**<objid>**

- Siirtopaketteja hallinnoidaan ingest-termillä
 - Toistaiseksi ainoa toiminto on tarkastusraporttien haku ja nouto
 - <objid> on pakollinen määre, joka yksilöi siirtopaketin
- Esimerkki:
 - <https://pas.csc.fi/api/2.0/mycontract/ingest/report/sip-001>

Siirtoraportin tarkastusraportin haku – palautusviesti

```
{  
  "status": "success",  
  "data": {  
    "results": [  
      {  
        "download": {  
          "html": "<report-url>?type=html", <- Tarkastusraportin URL:it (html ja xml)  
          "xml": "<report-url>?type=xml"  
        },  
        "id": <transfer-id>, <- Siirron tunniste  
        "date": <date>, <- Tarkastusraportin luontiaika  
        "status": <status> <- "accepted" (paketti hyväksytty), "rejected" (paketti hylätty)  
      },  
      ... ] <- Voi olla useita  
    }  
  }
```

Siirtoraportin tarkastusraportin nouto

GET <report-url>?type=html

GET <report-url>?type=xml

- Palauttaa tarkastusraportin suoraan HTML- tai XML-muotoisena
 - <report-url> saadaan raporttien haussa

GET <base>/<contract>/**statistics/overview**

- Tilastoja hallinnoidaan statistics-termillä

- Toistaiseksi ainoa toiminto on yleiskatsaus

- Palautusviesti:

```
{  
  "status": "success",  
  "data": {  
    "capacity": {  
      "used": <used>,  
      "available": <available>,  
      "total": <total>  
    },  
    "key_figures": {  
      "sips_accepted": <accepted>,  
      "objects_preserved": <objects>  
    }  
  }  
}
```

Sopimuksen kapasiteetti:

- Käytetty
- Vapaana
- Yhteensä

Sopimuksen avaintilastot:

- Siirtopaketteja hyväksyty
- Digitaalisia objekteja säilytyksessä



pas-support@csc.fi

<http://digitalpreservation.fi/>

Twitter @dpres_fi

<https://github.com/Digital-Preservation-Finland/>

15.11.2019: Koulutus datatuuen henkilöstölle (mm. IDA/Fairdata-yhteyshenkilöt) – ks. <https://fairdata.fi/>

26.11.2019: Paketointikomponentti-koulutus – seuraa: <http://digitalpreservation.fi/>