

Tiera Kuningas -projektin ServiceMix sovellus

Visma Consulting Oy, Olli Pöyry <olli.poyry@vismaconsulting.fi>, Ossi Karhunen <ossi.karhunen@visma.com> – Version 0.4, 25.11.2016

Sisällysluettelo

1. Versiohistoria
2. Sovellus
 - 2.1. KaPA sovitinpalvelun tarkoitus
 - 2.2. Kuningas sovitinpalvelun REST/SOAP palveluina tarjoamat KaPA SOAP palvelut
 - 2.3. Virheen käsittely
 - 2.4. Palveluväylän metapalvelut
3. Testaus
 - 3.1. Apache JMeter integraatiotesti
 - 3.2. REST kutsujen suoritus komentoriviltä
 - 3.3. JUnit integraatiotesti joka suorittaa REST kutsut
 - 3.4. Funktionaalinen testaus
 - 3.5. Mock palvelut
4. Sovelluksen asennus ServiceMix palvelimelle
 - 4.1. Sovelluspalvelimen asennus
 - 4.2. Manuaalinen asennus Karaf shell komennoilla
 - 4.3. Konfigurointitiedostot
 - 4.4. Sovellusmodulit
5. Sovelluksen lähdekoodi
6. REST-palveluita käyttävien sovellusten autentikointi TIA ServiceMix palveluun
 - 6.1. X.509 mutual authentication
 - 6.2. RFC 7519 JSON Web Token (JWT)
7. Sovelluksen monitorointi

Tämä dokumentti on ainoa Kuningas -projektin ServiceMix sovitinpalvelun pilottiprojektin kuvaus. Tässä dokumentissa on siten sekä teknistä kuvausta ja asennusohjeita, että yleisemmän tason kuvausta projektin tarkoituksesta ja vaihtoehtoista.

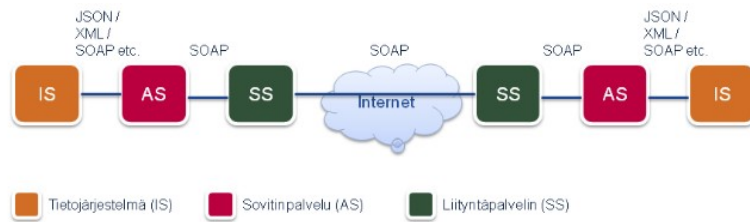
1. Versiohistoria

Päiväys	Versio	Kuvaus	Tekijä
8.11.2016	0.1	Ensimmäinen versio	Olli Pöyry
11.11.2016	0.2	Toinen versio	Olli Pöyry
23.11.2016	0.3	Kolmas versio	Olli Pöyry
25.11.2016	0.4	Neljäs versio	Olli Pöyry ja Ossi Karhunen

2. Sovellus

2.1. KaPA sovitinpalvelun tarkoitus

Sovitinpalvelu (<https://confluence.csc.fi/display/Palveluvayla/Sovitinpalvelu>) on X-Roadin liityntäpalvelimen (englanniksi 'Security Server') ja liitettävän järjestelmän väliin sijoittuva komponentti, joka sovittaa liitettävän järjestelmän tarjoamat palvelut X-Road-tiedonsiirto-protokollan mukaiseen muotoon.



Palveluiden liittäminen palveluväylään: ^[1]

- Palveluväylässä käytetään teknisenä alustana Virossa kehitettyä X-Roadia
- Tietojärjestemien liittäminen palveluväylään tapahtuu liityntäpalvelimien kautta.
- Liityntäpalvelin voi olla organisaation oma tai se voi olla jaettuna usean eri organisaation kesken
- Kaikki palveluväylässä kulkevat sanomat kulkevat suoraan liityntäpalvelimien välillä
- Eräs tapa liittää tietojärjestelmä osaksi palveluväylää on luoda tätä varten sovitinpalvelu, joka muuntaa liitettävän palvelun käyttämät viestit palveluväylässä käytettyyn SOAP-muotoon ja takaisin

Määrittämiä kansallisesta palveluväylän viitearkkitehtuurista: ^[2]

Liityntäpalvelin

Organisaation käytössä oleva julkisen palveluväylän palvelin

Sovitinpalvelu

Julkiseen palveluväylään liittyvän organisaation palvelu, jonka avulla muunnetaan palvelun sanomien tietosisältö palveluväylän käyttämään muotoon.

Julkisen palveluväylän sanoma

Varsinainen sanoma (tyypillisesti XML-viesti), jonka liityntäpalvelimet välittävät toisilleen.

Julkisen palveluväylän tukee vain synkronista viestintää. Sovitinpalvelu voi tarjota protokollan ja viestiformaattien muuntojen lisäksi asynkronisuuden toteutuksen.

SOAP-kyselyn valmistamiseen XRoad mukaiseen muotoon käytetään Väestörekisterikeskuksen järjestelmäpäällikkö Petteri Kivimäen kehittämää open source -kirjastoa <https://github.com/petkivim/xrd4j>.

Seuraava HTTP haku käyttää yhtä Kuningas sovitinpalvelun REST API:n tarjoamista kutsuista.

```
curl -X GET --header 'Accept: application/json' 'http://172.21.230.210:8080/ProxyWebApp/xroadexample/hello?name=Testi&clientId=2362180-3&requestId=12345'
```

BAT

Sovitinpalvelu muuntaa REST kutsun alla näkyväksi XRoad SOAP kutsuksi. XRoad-kirjaston edellyttämät SOAP Header elementit (<https://confluence.csc.fi/pages/viewpage.action?pageId=50873043#Suomi.fi-palveluväylä-X-Road-tiedonsiirtoprotokolla-4SOAP-otsikkotiedot>) on alustettu REST kutsun parametrien (mm. clientId) sekä konfigurointitiedoston sisällön mukaan.

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <xrd:client id:objectType="SUBSYSTEM" xmlns:xrd="http://x-road.eu/xsd/xr
      <id:xRoadInstance>FI-TEST</id:xRoadInstance>
      <id:memberClass>MUN</id:memberClass>
      <id:memberCode>2362180-3</id:memberCode>
      <id:subsystemCode>TIERA-SUBSI</id:subsystemCode>
    </xrd:client>
    <xrd:service id:objectType="SERVICE" xmlns:xrd="http://x-road.eu/xsd/xr
      <id:xRoadInstance>FI-TEST</id:xRoadInstance>
      <id:memberClass>GOV</id:memberClass>
      <id:memberCode>0245437-2</id:memberCode>
      <id:subsystemCode>TestService</id:subsystemCode>
      <id:serviceCode>helloService</id:serviceCode>
      <id:serviceVersion>v1</id:serviceVersion>
    </xrd:service>
    <xrd:id xmlns:xrd="http://x-road.eu/xsd/xroad.xsd">12345</xrd:id>
    <xrd:protocolVersion xmlns:xrd="http://x-road.eu/xsd/xroad.xsd">4.0</xrd
  </soap:Header>
  <soap:Body>
    <helloService xmlns="http://test.x-road.fi/producer" xmlns:ns2="http://x
      <request>
        <name>Test1</name>
      </request>
    </helloService>
  </soap:Body>
</soap:Envelope>

```

ServiceMix Kuningas sovelluksessa on toteutettu KaPA kutsujen lähetykset seuraavankaltaisista sisääntulevista pyynnöistä:

1. http GET kutsu jossa tarvittavat parametrit URL:n osina, esimerkiksi /ytj/GetCompany?lyTunnus=2362180-3&clientId=2362180-3&requestId=12345
2. http PUT tai POST jossa parametri sovelluksen omaa tietotyyppiä edustavana JSON tai XML dokumenttina (kuten tieraGetCompanyRequest, kts. alla)
3. http PUT tai POST jossa parametri KaPA palvelun mukaisena XML dokumenttina (kuten GetCompanyRequest, kts. alla)
4. SOAP kutsu johon Kuningas sovelluksessa täydennetään XRoad edellyttämät Header elementit.
5. Valmis SOAP kutsu joka välitetään sellaisenaan KaPA:an

Esimerkki JSON parametrilla YTJ GetCompany -hakuun. Tietotyyppi on Kuningas sovelluksessa määritetty "TieraGetCompanyRequest".

```

{
  "clientId" : "2362180-3",
  "lyTunnus" : "12345",
  "requestId" : "R1234"
}

```

JSON

Samana kyselypyynnön voi lähettää myös XML-muodossa, kuten alla.

```

<tieraGetCompanyRequest>
  <clientId>2362180-3</clientId>
  <lyTunnus>1988069-3</lyTunnus>
  <requestId>R1234</requestId>
</tieraGetCompanyRequest>

```

XML

Toiseen HTTP osoitteeseen Kuningas sovelluksessa voi tehdä vastaavan kyselyn lähettämällä jossa parametrina annetaan PRH:n tarjoaman SOAP WSDL:n mukainen GetCompany -dokumentti. Tässä tapauksessa clientId ja requestId välitetään HTTP-kutsun Header elementeissä.

```

<GetCompany xmlns="http://bis/dataservices/companyquery/v1">
  <request>
    <companyQuery>
      <BusinessId>2362180-3</BusinessId>
    </companyQuery>
  </request>
</GetCompany>

```

XML

Vastaus palvelusta voi olla joko XML tai JSON -muodossa, riippuen kutsun Header tiedoissa välitetystä "Accept" parametrilla "application/xml" tai "application/json". Vastaus on yleensä kutsun KaPA palvelun mukainen dokumentti, kuten

```
<GetCompanyResponse xmlns="http://bis/dataservices/companyquery/v1"
xmlns:ns2="http://schemas.microsoft.com/2003/10/Serialization/Arrays" xmlns:ns3="http://x-road.eu/xsd/identifiers"
xmlns:ns4="http://x-road.eu/xsd/xroad.xsd" xmlns:ns5="http://schemas.microsoft.com/2003/10/Serialization/">
...
</GetCompanyResponse>
```

XML

Tai vastaus on sovelluksen oma tietotyyppi, kuten:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<tieraRovaResponse>
  <authorization>ALLOWED</authorization>
  <delegateIdentifier>010180-953U</delegateIdentifier>
  <principalIdentifier>010215A951T</principalIdentifier>
</tieraRovaResponse>
```

XML

2.2. Kuningas sovitinpalvelun REST/SOAP palveluina tarjoamat KaPA SOAP palvelut

Kukin sovitinpalvelun kautta kutsuttava palvelu edellyttää esimerkiksi Java luokkien generointia palvelun WSDL:stä ja sitä käyttävän REST API:n lisäystä sovitinpalveluun. Kuningas ServiceMix sovelluksen konfigurointitiedostossa listataan KaPA palvelut joita sen kautta on mahdollista kutsua. Tällä hetkellä se tarjoaa kolme eri palvelua:

- VRK:n huoltajan ja huollettavien hetujen tarkistus ("rovaAuthorizationService")
- PRH:n YTJ yrityksen perustietojen haku Y-tunnuksella ("GetCompany")
- XRoad testisovelluksen "helloService"

VRK:n palvelu löytyy Liityntäkatalogista (Beta) kohdasta Liitynnät Suomi.fi-asiointivaltuudet rovaAuthorizationService.v1 (<https://liityntakatalogi.suomi.fi/fi/dataset/rolesauthsservice/resource/0d7d6c6d-bcc7-473b-b912-7ea02f94137c>).

PRH:n YTJ GetCompany -hakua ei ole liityntäkatalogissa. Sen sijaan PRH:lta on liityntäkatalogissa "xroadvirre-services" (tunniste palveluväylässä: FI.GOV.0244683-1.xroadvirre-services). Alla selitys "xroadvirre-services" ja "GetCompany" palvelut eroavat toisistaan.

“YTJ-palvelu muodostaa yrityksistä joukon, jonka osajoukko löytyy kaupparekisteristä ja siten Virre-palveluista. Kaupparekisterin sovelluspalveluista löytyy sitten monin osin tarkempia tietoja (pois lukien verohallinnon ylläpitämät tiedot, kuten ennakonperintärekisteriä koskevat tarkastuspäivät tai muut verohallinnon rekisteröinnit) mukaan lukien yritysten henkilöihin liittyvää tietoa. Vain osa yksityisistä elinkeinonharjoittajista löytyy kaupparekisteristä. GetCompany on hyvä perustietojen hakemiseen (osoitetieto, tila rekisterissä/lakannut, tilanne konkurssi/selvitystila). Tarkemmat, esimerkiksi yrityksen henkilöitä tai kaupparekisteriotetta koskevat kyselyt voi sitten tehdä Virren:n puolelta.

— Mikko Pitkänen (PRH)

2.3. Virheen käsittely

Jos KaPA liityntäpalvelin ei vastaa tulee REST kutsulle HTTP-virhe 500 Server Error esimerkiksi selitteellä "org.apache.cxf.interceptor.Fault: Could not send Message.". Kutsun lähetys saattaa kestää 20 sekuntia.

Jos kutsun parametreissa annettava clientId on väärä, eli sellainen jota ei ole lisätty konfigurointitiedostoon fi.tiera.integraatioalusta.kapaconsumer.cfg, tulee virhe 500 selitteellä "Check properties for client 2362180-3".

Jos parametreista puuttuu KaPA kutsun yksilöivä requestId, tulee virhe 500 selitteellä "com.pkrete.xrd4j.common.exception.XRd4JException: "id" can't be empty.".

Jos GetCompany kyselyssä on parametrinä väärän muotoinen Y-tunnus tulee virhe 500 selitteellä "org.apache.cxf.binding.soap.SoapFault: Business ID is incorrect." Jos Y-tunnus on oikean muotoinen mutta sitä ei löydy, tulee virhe 500 selitteellä "org.apache.cxf.binding.soap.SoapFault: Company not found.".

2.4. Palveluväylän metapalvelut

Metapalveluilla (<https://esuomi.fi/palveluntarjoajille/palveluvayla/tekninen-aineisto/rajapintakuvaukset/metapalvelut/>) voi kysellä palveluväylään liittyneistä organisaatioista ja niiden tarjoamista palveluista. Metapalveluiden kuvaus löytyy myös täältä (<https://confluence.csc.fi/display/Palveluvayla/Metapalvelut>).

Kuningas ServiceMix palvelimella on REST palvelu, joka kutsuu edelleen palveluväylän `allowedMethods` ja `listMethods` metapalveluita. REST url on `/allowedMethods/{clientId}/{requestClass}/` jossa:

- `clientId` on esim. Tieran Y-tunnus
- `requestClass` on SOAP palvelun WSDL:stä generoidun SOAP body:n Java luokan nimi polkuineen

`http://172.21.230.210:8080/ProxyWebApp/meta/allowedMethods/2362180-3/bis.dataservices.companyquery.v1.GetCompany`

Vastauksena tulevien palveluiden kutsuminen on sallittua Kuningas ServiceMix palvelimelta FI-TEST ympäristössä.

```
[
  {
    "memberClass": "GOV",
    "memberCode": "0244683-1",
    "subsystemCode": "xroadytj-services",
    "objectType": "SERVICE",
    "namespacePrefix": null,
    "namespaceUrl": null,
    "serviceCode": "GetCompany",
    "serviceVersion": "v1",
    "xroadInstance": "FI-TEST"
  }
]
```

JSON

Kyselyn yhteydessä haetaan asynkronisesti kunkin KaPA palvelun WSDL-määrittystiedosto liityntäpalvelimen osoitteesta `/wsdl`.

```
servicemix@kuningas-sma1 ~/instances/c2 $ pwd
/opt/servicemix/instances/c2
servicemix@kuningas-sma1 ~/instances/c2 $ ls -l *.wsdl
-rw-r--r-- 1 servicemix servicemix 67555 Nov 22 15:14 GetCompany.wsdl
-rw-r--r-- 1 servicemix servicemix 33451 Nov 22 15:01 getPTVOrganizationList.wsdl
-rw-r--r-- 1 servicemix servicemix 33451 Nov 22 15:01 getPTVOrganization.wsdl
-rw-r--r-- 1 servicemix servicemix 8878 Nov 22 15:01 getRandom.wsdl
-rw-r--r-- 1 servicemix servicemix 8878 Nov 22 15:01 helloService.wsdl
-rw-r--r-- 1 servicemix servicemix 8173 Nov 22 15:19 rovaAuthorizationService.wsdl
```

BAT

3. Testaus

Testauksella varmistetaan että toimitettava sovellus vastaa määrittämiä, Kuningas projektissa testausta on suoritettu kolmella tasolla:

- Yksikkötestaus käyttäen JUnit työkalua joka tarjoaa työkalut testien automatisointiin.
- Integraatiotestit joilla testataan komponenttien välisiä rajapintoja
- Järjestelmätestaus jossa järjestelmä testataan käyttöliittymän kautta.

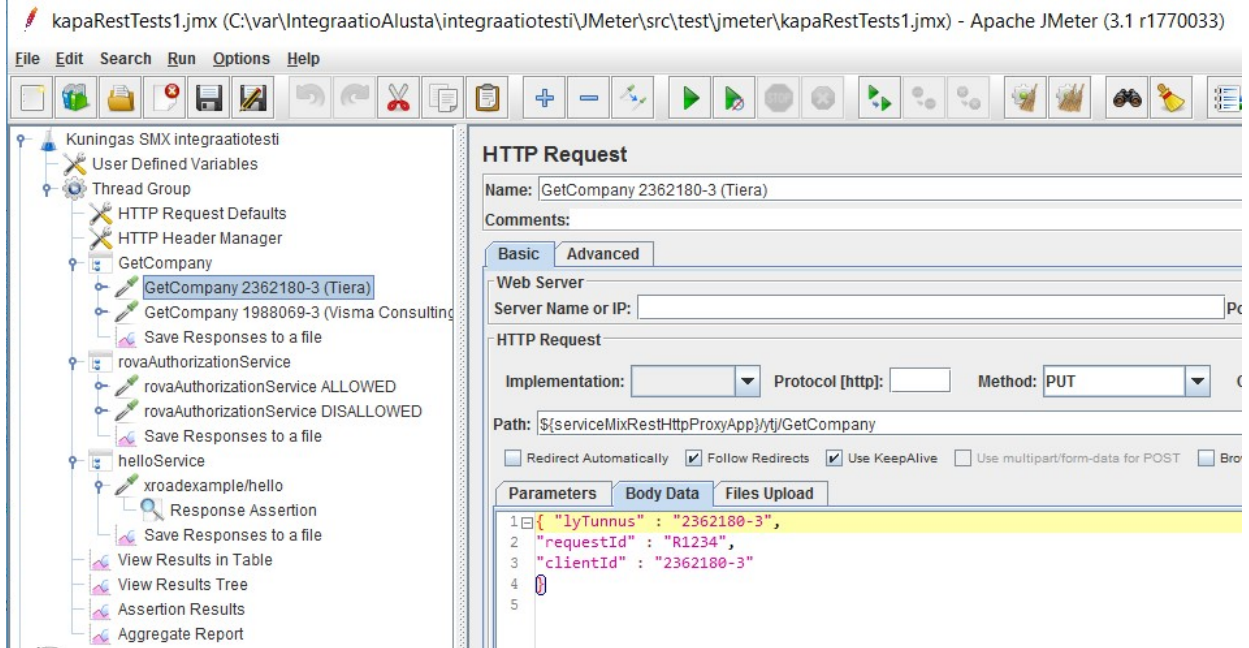
Seuraavissa kappaleissa kuvataan tarkemmin testausta erityisesti integraatiotestauksen näkökulmasta. Testaus on rajattu pelkästään kuningas projektissa toteutettaviin ominaisuuksiin, sähköisen asioinnin asioinnin testaus ei kuulu kokonaisuuteen. Testauksen tulokset testitapauksineen on kirjattu liitteeseen 1.

3.1. Apache JMeter integraatiotesti

Modulin integraatiotesti/JMeter alahakemistossa `src/test/jmeter` on Apache JMeter (<http://jmeter.apache.org/>) integraatiotesti tiedostossa `kapaRestTests1.jmx`. JMeter kutsuu REST API:a ja saa vastauksena KaPA vastaukset JSON:ina. JMeter testin voi suorittaa joko interaktiivisesti JMeter GUI:ssa, tai komentoriviltä Maven:illä komennolla:

mvn clean verify

Maven konfigurointitiedosto käyttää [JMeter Maven Plugin:ia](https://github.com/jmeter-maven-plugin/jmeter-maven-plugin/wiki) (https://github.com/jmeter-maven-plugin/jmeter-maven-plugin/wiki)



Huomioi että JMeter testin alussa on kohta "User Defined Variables" jonka arvoja muuttamalla saat määritettyä kutsuttavan ServiceMix instanssin.

Name	Value
serviceMixRestHttpIp	172.21.230.210
serviceMixRestHttpPort	8080
serviceMixRestHttpProxyApp	/ProxyWebApp

Testi tallettaa vastaukset tiedostoihin alihakemistoon target/jmeter/testFiles :

```
23.11.2016 14.02          47 370 GetCompany1.json
23.11.2016 14.02          59 926 GetCompany2.json
23.11.2016 14.02          114 rovaAuthorizationService3.json
23.11.2016 14.02          117 rovaAuthorizationService4.json
23.11.2016 14.02          130 xroadexample5.json
```

BAT

Tiedostoon target/jmeter/testFiles/assertionResults.txt kirjoitetaan JMeter "Response Assertion" tulokset, sarakkeessa "success" tulee olla "true" kaikilla riveillä.

```
timestamp,elapsed,label,responseCode,responseMessage,threadName,dataType,success,failureMessage,bytes,sentBytes,grpThread
s,allThreads,Latency,IdleTime,Connect
1479902209727,60201,GetCompany 2362180-3 (Tiera),500,Internal Server Error,Thread Group 1-1,text,false,"Test failed:
text expected to contain / "Name" : ""Kuntien Tiera Oy""/",1962,297,1,1,60200,0,73
1479902269950,60106,GetCompany 1988069-3 (Visma Consulting Oy),500,Internal Server Error,Thread Group
1-1,text,false,"Test failed: text expected to contain /"Name" : ""Visma Consulting Oy""/",1962,297,1,1,60106,0,13
1479902330064,3253,rovaAuthorizationService ALLOWED,200,OK,Thread Group 1-1,text,true,,551,394,1,1,3253,0,13
1479902333324,2097,rovaAuthorizationService DISALLOWED,200,OK,Thread Group 1-1,text,true,,554,394,1,1,2096,0,15
1479902335431,373,xroadexample/hello,200,OK,Thread Group 1-1,text,true,,567,289,1,1,372,0,17
```

BAT

3.2. REST kutsujen suoritus komentoriviltä

Alla olevat curl -kutsut on kopioitu Swagger UI:sta sivulta <http://172.21.230.210:8080/swagger-ui/index.html> Kutsut käyttävät Tomcat palvelimelle asennettua proxy sovellusta.

```
curl -X GET --header 'Accept: application/json' 'http://172.21.230.210:8080/ProxyWebApp/ytj/GetCompany?lyTunnus=2362180-3&clientId=2362180-3&requestId=12345'

curl -X PUT --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{"lyTunnus": "1753580-2", "clientId": "2362180-3", "requestId": "R1234"}' 'http://172.21.230.210:8080/ProxyWebApp/ytj/GetCompany'

curl -X GET --header 'Accept: application/json' 'http://172.21.230.210:8080/ProxyWebApp/xroadexample/hello?name=World&clientId=2362180-3&requestId=12345'

curl -X PUT --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ \
  "clientId": "2362180-3", \
  "name": "test", \
  "requestId": "1234" \
}' 'http://172.21.230.210:8080/ProxyWebApp/xroadexample/hello'
```

BAT

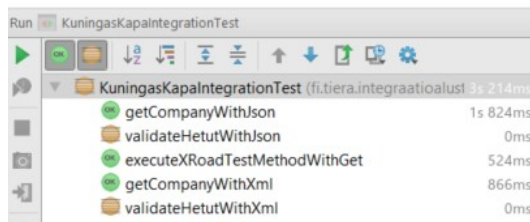
Kuningas palvelimella voi käyttää myös ServiceMix instanssin suoraa osoitetta kuten:

```
curl -X PUT --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{"clientId": "2362180-3", "name": "test", "requestId": "1234"}' 'http://172.21.230.210:8122/xroadexample/hello'
```

BAT

3.3. JUnit integraatiotesti joka suorittaa REST kutsut

Jenkins työ <http://ci-dev.dev.tia/job/Integraatiotesti/> suorittaa testit moduliasta `integraatiotesti/KuningasJerseyClient/pom.xml`. Tällä hetkellä Jenkins -palvelimelta ei ole pääsyä ServiceMix palvelimelle eli testi ei toimi sieltä. Sen voi ajaa kuitenkin omalta kehityskoneelta VPN-yhteydellä jolloin testit näyttäivät vihreää.



VRK:n RovaAuthorizationService palvelua kutsuvat testit on merkitty `@Ignore` annotaatiolla kunnes VRK yhteys on avattu.

3.4. Funktionaalinen testaus

Toimijarekisterin kautta voi suorittaa yrityksen perustietojen haku -kyselyn.

Päivähoitoanomuslomakkeen Orbeon prototyypin kautta voi suorittaa huoltajan ja huollettavan henkilötietojen tarkistuksen.

Käytettävissä olevat REST-kyselyt näkyvät Swagger UI -selainkäyttöliittymän kautta osoitteessa <http://172.21.230.210:8080/swagger-ui/index.html>.

Kuningas KaPA API

GetCompany

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET /ytj/GetCompany Hae yrityksen tiedot Y-tunnuksella

Response Class (Status 200)
Output type

Model | Example Value

```

{
  "getCompanyResponse": {
    "request": {
      "companyQuery": {
        "businessId": "string"
      }
    },
    "response": {
      "getCompanyResult": {
        "company": {

```

Response Content Type [application/json](#)

Parameters

Parameter	Value	Description	Parameter Type	Data Type
lyTunnus	2362180-3	Haettavan yrityksen Y-tunnus	query	string
clientId	2362180-3	KaPA consumer (Tiera) Y-tunnus	query	string
requestId	12345	KaPA haun tunniste	query	string

[Try it out!](#)

POST /ytj/GetCompany Hae yrityksen tiedot Y-tunnuksella

PUT /ytj/GetCompany Hae JSON:illa kutsu { "lyTunnus": "1753580-2", "clientId": "2362180-3", "requestId": "R1234" }

hello [Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

rovaAuthorizationService [Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

3.5. Mock palvelut

Kehityksessä ja testauksessa voi käyttää oikean liityntäpalvelimen sijaan mock -versioita:

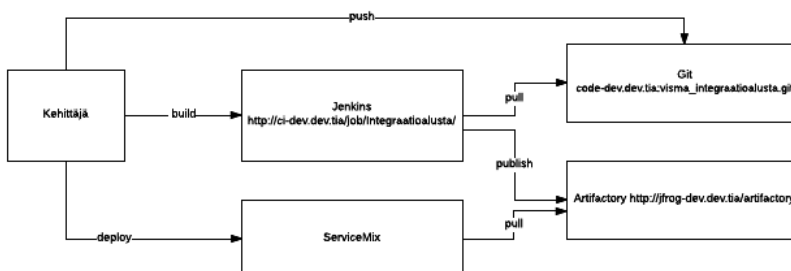
1. KapaMockServer
2. X-Road Adapter Example

KapaMockServer on Kuningas sovelluksen osana kehitetty moduli

IntegraatioAlusta/integraatiotesti/KapaMockServer/pom.xml . Se on Spring Boot -sovellus, joka käynnistyy komennolla `mvn spring-boot:run`. KapaMockServer palauttaa staattisen vastauksen GetCompany ja RovaAuthorizationService SOAP pyyntöihin.

<https://github.com/petkivim/x-road-adapter-example> sisältää saman sovelluksen joka on kutsuttavissa myös KaPA väljän kautta Tierasta. x-road-adapter-example käynnistetään antamalla hakemistossa x-road-adapter-example/src komento `mvn tomcat7:run`

4. Sovelluksen asennus ServiceMix palvelimelle



(<http://www.flickr.com/photos/javh/5448336655>)

Figure 1. Sovelluksen CI kulku ServiceMix:iin.^[3]

1. Sovelluksen lähdekoodi talletetaan Git repositoryyn `code-dev.dev.tia:visma_integraatioalusta.git`
2. Jenkins job `http://ci-dev.dev.tia/job/Integraatioalusta/` kääntää sovellusmodulit ja
3. lataa ne Artifactory Maven repositoryyn `http://jfrog-dev.dev.tia/artifactory`
4. ServiceMix konfiguroidaan siten että käännettyt modulit haetaan Artifactorystä

Sovelluksen asennus ServiceMix:iin voidaan tehdä joko eksplisiittisesti antamalla asennuskomennot Karaf-shell käyttöliittymässä (kts. alla [Manuaalinen asennus]), tai seuraavan kappaleen ohjeen mukaan luomalla uusi ServiceMix instanssi siten että sovellus määritetään instanssin konfigurointitiedostoihin ja se asentuun instanssin luonnin yhteydessä.

4.1. Sovelluspalvelimen asennus

Sovelluksen voi asentaa joko Apache Karaf (<http://karaf.apache.org/>) tai Apache ServiceMix (<http://servicemix.apache.org/>) asennuksen päälle.



Apache Karaf yhtä hyvin kuin Apache ServiceMix

"ServiceMix kernel was pulled out of ServiceMix into its own project - Apache Karaf."



Instanssin saa luotua uudelleen poistamalla kokonaan data -alahakemiston, esimerkiksi `rm -Rf /opt/servicemix/instances/c1/data` Sen jälkeen instanssi on samanlainen kuin ennen ensimmäistä käynnistystä.

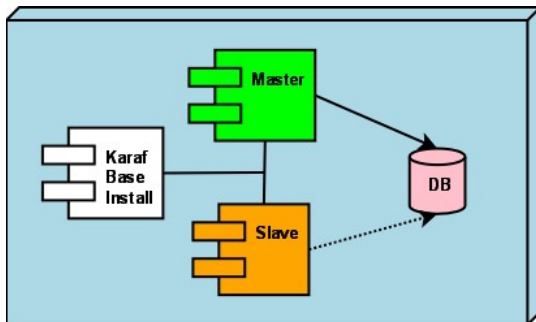
Karaf -shell komennolla `instance:create` voi luoda sovelluspalvelimesta alainstanssin joka toimii omana JVM prosessinaan. Ennen instanssin käynnistystä täytyy muuttaa Jetty http palvelimen portti oletuksesta (8181), joka saattaisi mennä root-instanssin Jettyyn kanssa päällekkäin. Editoi tiedoston `etc/org.ops4j.pax.web.cfg` riviä siten että porttinumero 8181 vaihdetaan esimerkiksi 8182.

```
org.osgi.service.http.port = 8181
```

Karaf child instance kuvaus:

“ A Karaf child instance is a copy of Karaf that you can launch separately (its own system process) and deploy applications into. An instance does not contain the full copy of Karaf, but only a copy of the configuration files and data folder which contains all the runtime information, logs and temporary files. Each of the Karaf child instances will "live" in your `KARAF_HOME/instances` folder.

Instanssin käyttö hot tai cold standby instanssina, kts. [Karaf Failover Deployments](http://servicemix.apache.org/docs/7.x/users-guide/failover.html) (<http://servicemix.apache.org/docs/7.x/users-guide/failover.html>)



Sovelluksen omat modulit asentuvat ServiceMix -palvelimeen Tieran Artifactory repositorystä. Alla näkyvässä tiedoston `etc/org.ops4j.pax.url.mvn.cfg` asetuksessa `org.ops4j.pax.url.mvn.repositories` on lisättyä kaksi ensimmäistä `http://jfrog-dev.dev.tia/artifactory` -alkavaa riviä.

```

org.ops4j.pax.url.mvn.repositories= \
  http://jfrog-dev.dev.tia/artifactory/visma@id=libs-snapshot@snapshots@noreleases, \
  http://jfrog-dev.dev.tia/artifactory/visma@id=libs-release, \
  http://repo1.maven.org/maven2@id=central, \
  http://repository.springsource.com/maven/bundles/release@id=spring.ebr.release, \
  http://repository.springsource.com/maven/bundles/external@id=spring.ebr.external, \
  http://zodiac.springsource.com/maven/bundles/release@id=gemini, \
  http://repository.apache.org/content/groups/snapshots-group@id=apache@snapshots@noreleases, \
  https://oss.sonatype.org/content/repositories/snapshots@id=sonatype.snapshots.deploy@snapshots@noreleases, \
  https://oss.sonatype.org/content/repositories/ops4j-
snapshots@id=ops4j.sonatype.snapshots.deploy@snapshots@noreleases, \
  http://repository.springsource.com/maven/bundles/external@id=spring-ebr-repository@snapshots@noreleases

```

YAML

Artifactoryn käyttäjätunnus/salasana täytyy lisätä ~/.m2/settings.xml -tiedostoon.

Sovellus saadaan asentumaan instanssin luonnin yhteydessä editoimalla ServiceMix:in konfigurointitiedostoa etc/org.apache.karaf.features.cfg joka on instanssin alahakemistossa, esimerkiksi /opt/servicemix/instances/c2/etc. Ennen instanssin ensimmäistä käynnistystä muutetaan kahta asetusta: featuresRepositories sekä featuresBoot.

```

featuresRepositories =
mvn:org.apache.karaf.features/spring/4.0.7/xml/features,mvn:org.apache.karaf.features/framework/4.0.7/xml/features,mvn:org
g.apache.karaf.features/enterprise/4.0.7/xml/features,mvn:org.apache.karaf.features/standard/4.0.7/xml/features,mvn:org.a
pache.camel.karaf/apache-camel/2.16.3/xml/features,mvn:org.apache.cxf.karaf/apache-
cxf/3.1.5/xml/features,mvn:io.hawt/hawtio-
karaf/1.4.66/xml/features,mvn:fi.tiera.integraatioalusta/IntegraatioAlustaDependencyFeatures/1.0-SNAPSHOT/xml/features,mv
n:fi.tiera.integraatioalusta/IntegraatioAlustaFeatures/1.0-SNAPSHOT/xml/features

```

YAML

Verrattuna oletukseen asetuksen lopussa on kaksi lisäystä:

1. mvn:fi.tiera.integraatioalusta/IntegraatioAlustaDependencyFeatures/1.0-SNAPSHOT/xml/features
2. mvn:fi.tiera.integraatioalusta/IntegraatioAlustaFeatures/1.0-SNAPSHOT/xml/features

Tiedostoon org.apache.karaf.features.cfg lisätään myös uuden ServiceMix -instanssin ensimmäisessä käynnistyksessä automaattisesti asennettavat featuret, Tieran lisäykset lopussa:

```

featuresBoot = (instance,package,log,ssh,aries-blueprint,framework,system,feature,shell,management,service,jaas,shell-
compat,deployer,diagnostic,wrap,bundle,config,kar,camel,cxf,camel-cxf,camel-jaxb,camel-blueprint,camel-xstream,war),
(hawtio),(IntegraatioalustaDependencies),(IntegraatioalustaJwt),(Integraatioalusta),XRoadExample,Ytj,Rova

```

YAML

Verrattuna ServiceMix:in oletusasetukseen yllä olevasta on jätetty pois ActiveMQ JMS-palvelin ja asennettavat featuret on ryhmitelty sulkeilla siten että asennus tehdään vaiheissa.

4.2. Manuaalinen asennus Karaf shell komennoilla

Saman lopputuloksen kuin yllä kuvattu org.apache.karaf.features.cfg -tiedoston editointi ennen ensimmäistä käynnistystä saa antamalla Karaf shell:issä komennot:

```

feature:repo-add mvn:fi.tiera.integraatioalusta/IntegraatioAlustaDependencyFeatures/1.0-SNAPSHOT/xml/features
feature:repo-add mvn:fi.tiera.integraatioalusta/IntegraatioAlustaFeatures/1.0-SNAPSHOT/xml/features

feature:install IntegraatioalustaDependencies
feature:install IntegraatioalustaJwt
feature:install Integraatioalusta
feature:install XRoadExample
featexture:install Ytj
feature:install Rova

```

BAT

4.3. Konfigurointitiedostot

Tiedostossa /opt/servicemix/instances/c1/etc/fi.tiera.integraatioalusta.kapaconsumer.cfg on määritetty esimerkiksi XRoad testipalvelun kutsun SOAP header tiedot seuraavasti. Huomioi että liityntäpalvelimen http-osoite, XRoad instanssin nimi (FI-TEST) ja palvelun versio (v1) on kaikille kolmelle palvelulle sama, ja se on määritetty niille yhteisesti.

```

# Service defaults
service.serviceVersion = v1
service.url = http://172.21.230.226:8080
service.xRoadInstance = FI-TEST

# x-road-adapter-example helloService
service.fi.x_road.test.producer.HelloService.memberClass = GOV
service.fi.x_road.test.producer.HelloService.memberCode = 0245437-2
service.fi.x_road.test.producer.HelloService.serviceCode = helloService
service.fi.x_road.test.producer.HelloService.subsystemCode = TestService

#
service.bis.dataservices.companyquery.v1.GetCompany.memberClass = GOV
service.bis.dataservices.companyquery.v1.GetCompany.subsystemCode = xroadytj-services
service.bis.dataservices.companyquery.v1.GetCompany.memberCode = 0244683-1
service.bis.dataservices.companyquery.v1.GetCompany.operationName = GetCompany
service.bis.dataservices.companyquery.v1.GetCompany.operationNamespace = http://bis/dataservices/companyquery/v1
service.bis.dataservices.companyquery.v1.GetCompany.resultClass = bis.dataservices.companyquery.v1.GetCompanyResponse
service.bis.dataservices.companyquery.v1.GetCompany.serviceCode = GetCompany

#
service.fi.vrk.xml.ws.rova.authorization.entities.RovaAuthorizationService.memberClass = GOV
service.fi.vrk.xml.ws.rova.authorization.entities.RovaAuthorizationService.memberCode = 0245437-2
service.fi.vrk.xml.ws.rova.authorization.entities.RovaAuthorizationService.serviceCode = rovaAuthorizationService
service.fi.vrk.xml.ws.rova.authorization.entities.RovaAuthorizationService.subsystemCode = RolesAuthsService

service.eu.x_road.xsd.xroad.AllowedMethods.memberClass = GOV
service.eu.x_road.xsd.xroad.AllowedMethods.memberCode = 0245437-2
service.eu.x_road.xsd.xroad.AllowedMethods.serviceCode = allowedMethods
service.eu.x_road.xsd.xroad.AllowedMethods.subsystemCode = TestService
service.eu.x_road.xsd.xroad.AllowedMethods.operationName = allowedMethods
service.eu.x_road.xsd.xroad.AllowedMethods.operationNamespace = http://metadata.x-road.eu/

service.eu.x_road.xsd.xroad.ListMethods.memberClass = GOV
service.eu.x_road.xsd.xroad.ListMethods.memberCode = 0245437-2
service.eu.x_road.xsd.xroad.ListMethods.serviceCode = listMethods
service.eu.x_road.xsd.xroad.ListMethods.subsystemCode = TestService
service.eu.x_road.xsd.xroad.ListMethods.operationName = listMethods
service.eu.x_road.xsd.xroad.ListMethods.operationNamespace = http://metadata.x-road.eu/

```

Kuningas sovitinpalvelun tarjoamissa REST-kutsuissa kerrotaan yhdellä parametrilla kutsun esittäjän Y-tunnus. Sen perusteella haetaan konfigurointitiedostosta SOAP header -elementteihin lisättävät tiedot. Konfigurointitiedostossa on annettu alijärjestelmän tunniste (subsystemCode) palvelu- ja käyttäjäkohtaisesti client.{clientId}.{requestClass}.subsystemCode .

```

# Client defaults:
client.memberClass = MUN
client.xRoadInstance = FI-TEST

# Kuntien Tiera Oy
client.2362180-3.memberCode = 2362180-3

# Espoon kaupunki
client.0101263-6.memberCode = 0101263-6

#
client.2362180-3.fi.x_road.test.producer.HelloService.subsystemCode = TIERA-SUBS1
client.2362180-3.fi.vrk.xml.ws.rova.authorization.entities.RovaAuthorizationService.subsystemCode = TieraVRKclient
client.2362180-3.bis.dataservices.companyquery.v1.GetCompany.subsystemCode = TieraYTJclient

```

Vertailun vuoksi <https://github.com/vrk-kpa/roles-auths-client> "A client library for Asiointivaltuudet -services" sovelluksen käyttämä .properties -tiedosto olisi tällainen:

```
roles-auths-client.client_xroad_instance = FI-TEST
roles-auths-client.client_member_class = MUN
roles-auths-client.client_member_code = 2362180-3
roles-auths-client.client_subsystem_code = TieraVRKclient
roles-auths-client.client_version = v1

roles-auths-client.service_xroad_instance = FI-TEST
roles-auths-client.service_member_class = GOV
roles-auths-client.service_member_code = 0245437-2
roles-auths-client.service_subsystem_code = RolesAuthsService
roles-auths-client.service_version = v1
```

YAML

Tiedostossa `etc/fi.tiera.integraatioalusta.rest.cfg` kerrotaan missä portissa Kuningas REST API palvelee. Liityntäpalvelimen osoite kerrotaan KaPA meta-tietoja hakevaa modulia varten. Parametri `useMethodReferences` liittyy testiin jossa Camel viestinvälitys haarautuu käyttämään joko `direct-vm`: -viestinvälitystä OSGi modulien välillä, tai sama asia tehdään OSGi interface kutsulla modulista toiseen.

```
tia.rest.host = 172.21.230.210
tia.rest.port = 8123

securityServerUrl = http://172.21.230.226:8080

useMethodReferences = false
```

YAML

4.4. Sovellusmodulit

Sovelluksen onnistuneen käynnistymisen jälkeen ServiceMix Karaf -shell komento `camel:context-list` näyttää kuusi Camel kontekstia:

```
karaf@c2>camel:context-list
Context              Status          Total #      Failed #      Inflight #      Uptime
-----
KapaConsumerContext Started         172          2             0      1 day
MetaService          Started          0            0             0      1 day
RovaConsumerContext Started          72            0             0      1 day
XRoadExampleConsumerContext Started          40            0             0  19 hours 20 minutes
YtjConsumerContext  Started         168          23            0      1 day
YtjRestService       Started          0            0             0      1 day
karaf@c2>
```

BAT

Karaf shell (<http://camel.apache.org/karaf.html>) -komentoriviltä voi kysyä modulikohtaisesti, esimerkiksi:

```
camel:endpoint-list KapaConsumerContext
camel:endpoint-list YtjConsumerContext
camel:endpoint-list RovaConsumerContext
```

BAT

5. Sovelluksen lähdekoodi

Sama lähdekoodi on kahdessa Git repositoryssä, Tierassa ja Visma Consultingissa:

- `git@code-dev.dev.tia:visma_integraatioalusta.git`
- `git@git-ssh.vismaconsulting.fi:poyryoll/Integraatioalusta.git`

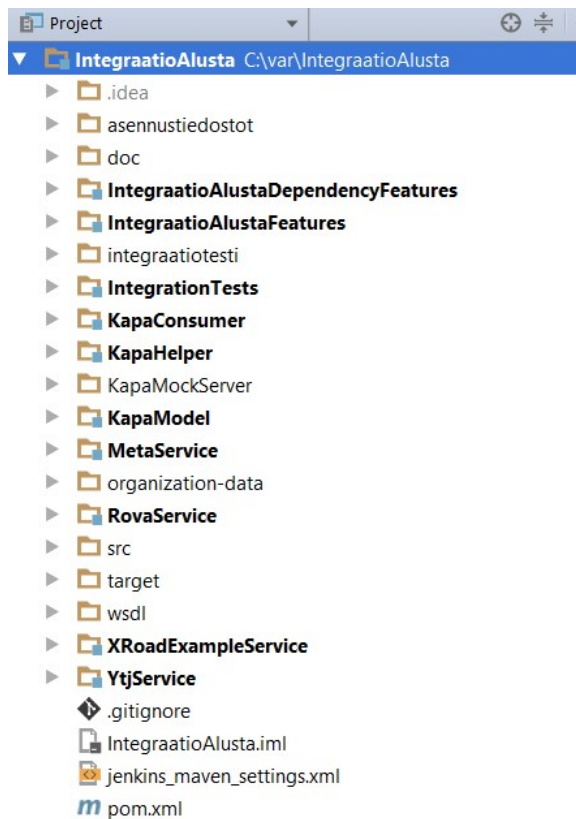


Table 1. Lähdekoodin alajakemistot

Hakemiston nimi	ServiceMix OSGi bundle	Selite
IntegraatioAlustaDependencyFeatures		Asennuksessa käytettävä feature-paketti sovelluksen vaatimille ServiceMix ominaisuuksille
IntegraatioAlustaFeatures		ServiceMix asennuksessa käytettävä feature-paketti itse sovellukselle
IntegrationTests		Yksikkötestejä
KapaConsumer	K	Valmistele SOAP kysely ennen lähetystä Liityntäpalvelimelle
KapaHelper	K	OSGi service bean malli
KapaMockServer		Mock implementaatio YTJ GetCompany ja VRK rovaAuthorizationService SOAP palveluista
KapaModel	K	Sovelluksen yhteiskäyttöisiä POJO-luokkia
MetaService	K	Kutsu KaPA listMethods ja allowedMethods palveluita
RovaService	K	VRK rovaAuthorizationService client
XRoadExampleService	K	XRoad mallisovelluksen helloService palvelun client

Hakemiston nimi	ServiceMix OSGi bundle	Selite
YtjService	K	YTJ GetCompany palvelun client
asennustiedostot		ServiceMix asennuksessa käytettäviä tiedostoja
doc		Tämä dokumentti
integraatiotesti		Jersey REST client kirjastoa käyttävä Unit integraatiotesti
organization-data		jar Maven projekti käytettäväksi service2 Tomcat sovelluksessa Orbeon lomakkeen taustalla. Kutsuu rovaAuthorizationService palvelua.
wSDL		Sovelluksen kutsumien KaPA SOAP palveluiden WSDL tiedostot

Tämä dokumentti on kirjoitettu käyttäen kirjastoa <http://asciidoctor.org/>. Maven pom.xml modulissa `IntegraatioAlusta/doc` käyttää [Maven pluginia](https://github.com/asciidoctor/asciidoctor-maven-plugin) (https://github.com/asciidoctor/asciidoctor-maven-plugin) joka tuottaa dokumentin ASCII lähdekoodista HTML ja PDF versiot.

6. REST-palveluita käyttävien sovellusten autentikointi TIA ServiceMix palveluun

6.1. X.509 mutual authentication

REST kutsut sallitaan vain jos asiakassovelluksella on hyväksytyt sertifikaatti.

6.2. RFC 7519 JSON Web Token (JWT)

Palvelua käytävällä sovelluksella ja ServiceMix TIA sovelluksella yhteinen avain. REST kutsuihin lisätään avaimella generoitu token, jonka voimassaolo voidaan laittaa esim. tunniksi.

[Java JWT: JSON Web Token for Java and Android](https://github.com/jwt/jwt) (https://github.com/jwt/jwt)

“*The compacted representation of a signed JWT is a string that has three parts, each separated by a .
 eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJKb2UifQ.ipevRNuRP6Hf1G8cFKnmUPtypruRC4fb1DwtoLL62SY Each
 section is base 64 encoded. The first section is the header, which at a minimum needs to specify the
 algorithm used to sign the JWT. The second section is the body. This section has all the claims of this
 JWT encoded in it. The final section is the signature. It's computed by passing a combination of the
 header and body through the algorithm specified in the header.*

7. Sovelluksen monitorointi

JMX konsolin (JConsole, JVisualVM, Hawtio) kautta `mbean org.apache.camel:context=YtjConsumer,type=routes,name="kapa-soap-call-GetCompany"` kertoo esimerkiksi:

Table 2. Camel viestinvälitysväylän (route) keskeiset valvontatiedot

Attribuutin nimi	Selite
LastProcessingTime	Viimeisimmän KaPA:an lähetetyn GetCompany SOAP-kyselyn suoritus aika millisekuntia

Attribuutin nimi	Selite
ExchangesCompleted	GetCompany -kyselyiden kokonaismäärä
ExchangesFailed	Epäonnistuneiden GetCompany -kyselyiden määrä

Java Monitoring & Management Console - pid: 17204 org.apache.karaf.main.Main server

Connection Window Help

Overview Memory Threads Classes VM Summary MBeans

- JMImplementation
- com.sun.management
- connector
- hawtio
- java.lang
- java.nio
- java.util.logging
- javax.perl
- jolokia
- org.apache.aries.blueprint
- org.apache.camel
 - components
 - consumers
 - context
 - dataformats
 - endpoints
 - errorhandlers
 - eventnotifiers
 - processors
 - producers
 - routes
 - KapaConsumer
 - RovaConsumer
 - "kapa-soap-call-RovaAuthorizationService"
 - Attributes
 - Operations
 - Notifications
 - "route18"
 - "route19"
 - "route20"
 - "rovaAuthorizationService"
 - "rovaAuthorizationServiceGet"
 - XRoadExampleConsumer
 - "helloService"
 - "helloServiceGet"
 - "kapa-soap-call-XRoadExample"
 - Attributes
 - Operations
 - Notifications
 - "route11"
 - "route12"
 - "route13"
 - YtyConsumer
 - "kapa-soap-call-GetCompany"
 - Attributes
 - Operations
 - Notifications

Name	Value
CamelId	YtyConsumerContext
CamelManagementName	YtyConsumer
DeltaProcessingTime	-31
Description	
EndpointUri	direct://kapa-soap-call-GetCompany
ExchangesCompleted	5
ExchangesFailed	0
ExchangesInflight	0
ExchangesTotal	5
ExternalRedeliveries	0
FailuresHandled	0
FirstExchangeCompletedExchangeId	ID-LAPTOP-56SO150D-60430-147884
FirstExchangeCompletedTimestamp	Fri Nov 11 09:36:36 EET 2016
FirstExchangeFailureExchangeId	
FirstExchangeFailureTimestamp	
InflightExchanges	0
LastExchangeCompletedExchangeId	ID-LAPTOP-56SO150D-60430-147884
LastExchangeCompletedTimestamp	Fri Nov 11 09:40:34 EET 2016
LastExchangeFailureExchangeId	
LastExchangeFailureTimestamp	
LastProcessingTime	116
Load01	
Load05	
Load15	
MaxProcessingTime	320
MeanProcessingTime	190
MessageHistory	true
MinProcessingTime	116
OldestInflightDuration	
OldestInflightExchangeId	
Redeliveries	0
ResetTimestamp	Fri Nov 11 09:36:20 EET 2016
RouteId	kapa-soap-call-GetCompany
RoutePolicyList	
StartTimestamp	Fri Nov 11 09:36:20 EET 2016
State	Started
StatisticsEnabled	true
TotalProcessingTime	953
Tracing	false
Uptime	4 minutes

Vastaava näkymä Hawtio (<http://hawt.io/>) selainsovelluksesta

localhost:8181/hawtio/jmx/attributes?nid=root-org.apache.camel-XRoadExampleConsumer-routes-"kapa-soap-call-XRoadExample"

hawtio

Camel Connect Dashboard Jetty JMX Logs Maven OSGi Terminal Threads

Attributes Operations Chart

Start Pause Stop Delete

Property	Value
Camel	XRoadExampleConsumerContext
Camel management name	XRoadExampleConsumer
Delta processing time	-246
Description	
Endpoint uri	direct://kapa-soap-call-XRoadExample
Exchanges completed	3
Exchanges failed	1
Exchanges inflight	0
Exchanges total	4
External redeliveries	0
Failures handled	0
First exchange completed exchange	ID-LAPTOP-56SO15OD-60430-1478847159547-8-4
First exchange completed timestamp	2016-11-11T09:38:55+02:00
First exchange failure exchange	ID-LAPTOP-56SO15OD-60430-1478847159547-8-1
First exchange failure timestamp	2016-11-11T09:38:40+02:00
Inflight exchanges	0
Last exchange completed exchange	ID-LAPTOP-56SO15OD-60430-1478847159547-8-14
Last exchange completed timestamp	2016-11-11T09:39:19+02:00
Last exchange failure exchange	ID-LAPTOP-56SO15OD-60430-1478847159547-8-1
Last exchange failure timestamp	2016-11-11T09:38:40+02:00
Last processing time	319

Apache Karaf Decanter (<https://karaf.apache.org/manual/decanter/latest-1/>) avulla voidaan esimerkiksi lähettää kerran minuutissa JMX statistiikkaa Kafka jonoon. Alla on esimerkki Camel Karaf collector:in (https://karaf.apache.org/manual/decanter/latest-1/#_camel_jmx) lähettämän Kafka viestin sisällöstä:


```

1 { "@timestamp" : "2016-11-11T11:51:39,148Z",
2   "CamelId" : "YtjConsumerContext",
3   "CamelManagementName" : "YtjConsumer",
4   "DeltaProcessingTime" : 0,
5   "EndpointUri" : "direct://kapa-soap-call-GetCompany",
6   "ExchangesCompleted" : 0,
7   "ExchangesFailed" : 1, 2
8   "ExchangesInflight" : 0,
9   "ExchangesTotal" : 1, 1
10  "ExternalRedeliveries" : 0,
11  "FailuresHandled" : 0,
12  "FirstExchangeFailureExchangeId" : "ID-LAPTOP-56S0150D-65413-1478865045507-3-1",
13  "FirstExchangeFailureTimestamp" : "Fri Nov 11 13:51:32 EET 2016",
14  "InflightExchanges" : 0,
15  "LastExchangeFailureExchangeId" : "ID-LAPTOP-56S0150D-65413-1478865045507-3-1",
16  "LastExchangeFailureTimestamp" : "Fri Nov 11 13:51:32 EET 2016", 3
17  "LastProcessingTime" : 0,
18  "Load01" : "",
19  "Load05" : "",
20  "Load15" : "",
21  "MaxProcessingTime" : 0,
22  "MeanProcessingTime" : 0,
23  "MessageHistory" : true,
24  "MinProcessingTime" : 0,
25  "ObjectName" : "org.apache.camel:context=YtjConsumer,type=routes,name=\"kapa-soap-call-GetCompany\"",
26  "Redeliveries" : 0,
27  "ResetTimestamp" : "Fri Nov 11 13:50:49 EET 2016",
28  "RouteId" : "kapa-soap-call-GetCompany",
29  "RoutePolicyList" : "",
30  "StartTimestamp" : "Fri Nov 11 13:50:49 EET 2016",
31  "State" : "Started",
32  "StatisticsEnabled" : true,
33  "TotalProcessingTime" : 0,
34  "Tracing" : false,
35  "Uptime" : "49.967 seconds",
36  "component_id" : 1,
37  "component_name" : "org.apache.karaf.decanter.collector.jmx",
38  "decanter_collector_name" : "jmx",
39  "event_topics" : "decanter/collect/jmx/jmx-local/org/apache/camel",
40  "felix_fileinstall_filename" : "file:/C:/apache-
41  karaf-4.0.7/instances/c3/etc/org.apache.karaf.decanter.collector.jmx-local.cfg",
42  "hostAddress" : "172.21.6.210",
43  "hostName" : "LAPTOP-56S0150D",
44  "karafName" : "c3",
45  "object_name" : "org.apache.camel:context=YtjConsumer,type=routes,name=\"kapa-soap-call-GetCompany\"",
46  "service_factoryPid" : "org.apache.karaf.decanter.collector.jmx",
47  "service_pid" : "org.apache.karaf.decanter.collector.jmx.57b6f154-8dd6-4358-801b-baf960340df9",
48  "timestamp" : 1478865099148,
49  "type" : "jmx-local",
50  "url" : "local"
}

```

- 1 Pyyntöjen kokonaismäärä
- 2 Epäonnistuneiden pyyntöjen määrä
- 3 Viimeisimmän epäonnistuneen pyynnön aikaleima

1. [MML Paikkatiedot palveluväylässä -loppuseminaari, 18.5.2016](https://www.paikkatietoikkuna.fi/c/document_library/get_file?uuid=0bea7e7e-2e88-4c34-ad61-3428ab81b7da&groupId=108478)

(https://www.paikkatietoikkuna.fi/c/document_library/get_file?uuid=0bea7e7e-2e88-4c34-ad61-3428ab81b7da&groupId=108478)

2. [Kansallisen palveluväylän viitearkkitehtuuri](https://www.avoindata.fi/data/dataset/a6835067-f09f-492c-9a0d-05590f71085c/resource/1c4184ae-6f51-41a6-8464-98d9560c5faf/download/Palveluvaylan-viitearkkitehtuuri20.pdf)

(<https://www.avoindata.fi/data/dataset/a6835067-f09f-492c-9a0d-05590f71085c/resource/1c4184ae-6f51-41a6-8464-98d9560c5faf/download/Palveluvaylan-viitearkkitehtuuri20.pdf>)

3. Muunnettu <http://giallone.blogspot.fi/2014/05/continuous-integration-with-jboss-fuse.html>

Liite 1: Toteutusympäristön arkkitehtuuri

