

4.6 Filhantering



DEPRICATED

Denna sida tillhör en äldre version av tjänsten.

Aktuell version av detta dokument hittar ni på:

<https://confluence.cgiostersund.se/display/PU/Filhantering>

Innehåll

Innehåll

- 1. Inledning
 - 1.1. Om dokumentet
 - 1.2. Konnektivitet
 - 1.3. Behörighet
- 2. Användningsfall
 - 2.1. Söka personuppgifter via fil
 - 2.1.1.1. Parametrar
 - 2.1.1.2. Exempel
 - 2.2. Hämta resultat för personsök
 - 2.2.1. Exempel
 - 2.2.1.1. SearchPersonsForProfileByOrder
 - 2.2.1.2. GetFilesForOrderId
 - 2.2.1.3. REST
 - 2.3. Bilagor för reservidentitet
 - 2.3.1. Ladda upp bilaga
 - 2.3.1.1. URL
 - 2.3.1.2. Parametrar
 - 2.3.1.3. Exempel
 - 2.3.2. Koppla / Koppla ifrån bilaga
 - 2.3.3. Hämta bilaga
 - 2.3.3.1. URL
 - 2.3.3.2. Parametrar
 - 2.3.3.3. Exempel
 - 2.3.4. Ta bort bilaga
 - 2.3.4.1. URL
 - 2.3.4.2. Parametrar
 - 2.3.4.3. Exempel
- 3. Dokumentation

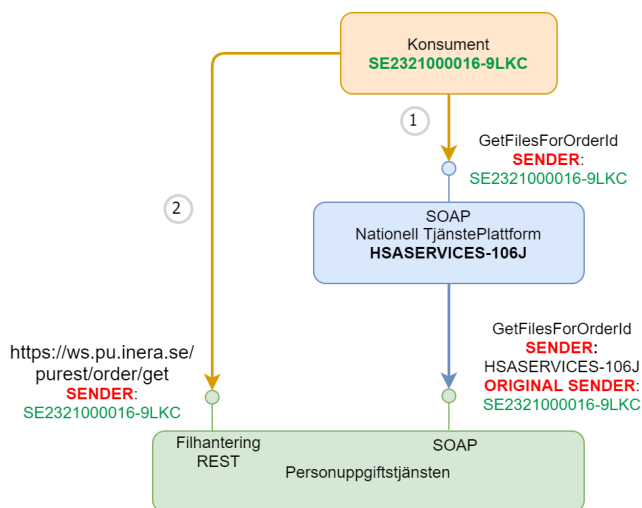
1. Inledning

1.1. Om dokumentet

Denna beskrivning är ett tillägg av tjänstekontrakten i tjänstedomänen **strategicresourcemanagement:persons:person**. Dokumentet beskriver hur tjänstens funktionalitet för att hämta och ladda upp filer är uppbyggd. Funktionaliteten baseras på dokument [ARK_0038](#) enligt RIV-TA standarden och där grundprincipen för att hämta filer bygger på en RIV-tjänst för att lista tillgängliga filer och en REST-tjänst för att hämta de filer som listas via RIV-tjänsten. Anrop mot tjänsterna sker med HTTPS (Mutual-TLS) och validering av certifikatet från tjänstekonsument utförs för att endast tillåta att tjänsten levererar data till tillåtna mottagare.

1.2. Konnektivitet

SOAP-tjänst för att lista tillgängliga filer är tillgängliga via Nationell Tjänsteplattform (NTJP). NTJP saknar dock stöd för REST-baserade tjänster vilket medför att REST-tjänst för att hämta filer (ladda upp/ta bort) är enbart tillgänglig genom direktaccess mot tjänsteproducent.



Figur: Anrop först till tjänst i NTJP med SOAP och sedan direkt till tjänsten med REST för att hämta resultatet från första anropet.

1.3. Behörighet

Vid användandet av REST-tjänster i Personuppgiftstjänsten måste man vara upplagd som webservice-klient direkt i Personuppgiftstjänsten. Därför måste samtliga system, förutom att bli anslutna till NTJP för t.ex SearchPersonsForProfileByOrder och/eller GetFilesForOrderId även kontakta Ineras kundtjänst på kundservice@inera.se med information i listan nedan. Detta är inte nödvändigtvis samma systemid som är anslutet till NTJP (som t.ex kan vara en regional tjänsteplattform) utan det HSA-id som återfinns i http headern *x-rivta-original-serviceconsumer-hsaid* i enlighet med [RIVTA Regel 23](#) samt "[Bevara ursprunglig avsändares identitet i http header](#)", original sender i bild ovan.

⚠ Mejllet ska innehålla följande värden:

- Miljö: Test/Prod
- REST-tjänst(er) man önskar ansluta till.
- Det anslutande systemets HSA-id, original sender i bild ovan: <SE...>
- Systemnamn: <...>
- Kontaktuppgifter: <endast funktionsbrevlåda vid produktionsanslutning>
- Tidigare ärendenummer för NTJP anslutningen ("Beställning från Beställningsstödet"): <...>

För att säkerställa att det är samma system som hämtar resultatfilen som gjorde sökningen så matchas HSA-id i certifikatet som användes vid sökningen (med det i filhämtningen (se bild i stycket ovan). Då själva sökningen skall göras via NTJP så används headern *x-rivta-originalserviceconsumer-hsaid* för att i tjänsten ta reda på vilken konsument som låg bakom anropet. Görs anropet dessutom genom en RTJP är det alltså viktigt att den regionala tjänsteplattformen populerar denna header (som ska förbli densamma genom NTJP ifall den redan existerar).

2. Användningsfall

Filhantering förekommer för två olika användningsfall inom PU-tjänsten.

1. Söka personuppgifter via fil
2. Hämta resultat för asynkron personsökning: SearchPersonsForProfileByOrder och Söka personuppgifter via fil.
3. Hämta, ladda upp samt ta bort bilagor för reservidentitet

Användningsfall 1 och 2 följer flödet enligt ARK_0038 (både SOAP och REST), medans användningsfall(en) för bilagor och filsök ej har behov av den listande SOAP-tjänsten. Flöden för samtliga användningsfall specificeras i följande kapitel.

2.1. Söka personuppgifter via fil

Ny funktion från version 4.6

Funktionen finns endast som *restricted*-variant. Returnerar alltså inte adress m.m på sekretessmarkerade personposter.

Flödet startar med att att en lista med personidentiteter behöver uppdateras. Listan upprättas som en radbruten CSV-fil där identiteterna anges med personid och personid-typ (OID) separerade med semikolon. Ex: 191212121212;1.2.752.129.2.1.3.1

Filen laddas upp via REST-metoden **SearchPersonsByFile** som vid lyckad uppladdning returnerar ett Order-Id. Detta Order-Id används som inparameter till tjänsten **GetFilesForOrderId** som returnerar ett svar om var den resulterande filen kan hämtas.

Om ordern inte är redo för nedladdning än så returneras inget svar, försök senare. Då ordern läggs på kö så kan väntetiden variera mycket. Kontakta supporten om din order inte är redo efter 24h.

OBS: Filer skapade via personsök är garanterat tillgängliga i ett dygn, därefter renas dom automatiskt bort från Personuppgiftstjänsten. Filer som har hämtats kan renas bort tidigare.

Den slutgiltiga filen som hämtas är en zippad fil innehållande en XML med det urval som efterfrågats. Formatet är densamma som i [GetPersonsForProfileResponse](#). Det innebär att varje efterfrågad personpost kommer med ett stycke för *RequestedPersonalIdentity* men endast ifall personposten hittas i databasen så levereras stycket *PersonRecord*. Sekretessmarkerade personuppgifter hanteras som i *GetPersonsForProfile* med endast ett fåtal uppgifter på returnerade sekretessflaggade personposter.



Kommande förändring

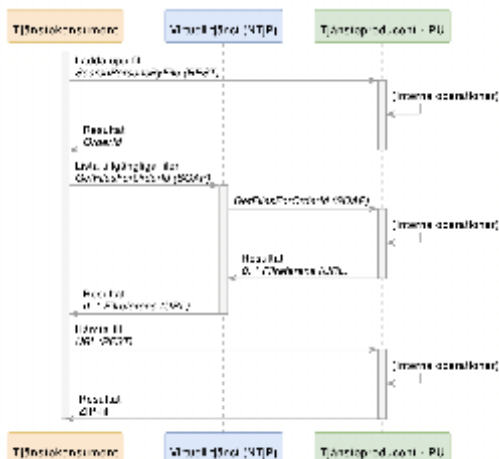
02 Dec 2021 Efter release av denna funktionalitet så upptäcktes att vid användande av parametern *fromDate* så returneras ändå stycket *RequestedPersonalIdentity* för de poster som inte har förändrats. Det är förstås inte önskvärt och kommer att förändras i nästa release.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:GetPersonsForProfileResponse xmlns:ns6="urn:riv:itintegration:r
      <ns2:requestedPersonRecord>
        <requestedPersonalIdentity>
          <root>1.2.752.129.2.1.3.1</root>
          <extension>197908190237</extension>
        </requestedPersonalIdentity>
      </ns2:requestedPersonRecord>

      <ns2:requestedPersonRecord>
        <requestedPersonalIdentity>
          <root>1.2.752.129.2.1.3.1</root>
          <extension>196709272691</extension>
        </requestedPersonalIdentity>
        <personRecord>
          <personalIdentity>
            <root>1.2.752.129.2.1.3.1</root>
            <extension>196709272691</extension>
          </personalIdentity>
          <gender>1</gender>
          <protectedPersonIndicator>false</protectedPersonIndicator>
        </personRecord>
      </ns2:requestedPersonRecord>
    </ns2:GetPersonsForProfileResponse>
  </soap:Body>
</soap:Envelope>
```

Ingen träff på 197908190237

Träff på 196709272691



2.1.1.1. Parametrar

Namn	Beskrivning
profile	Obligatorisk parameter för vilken profil som önskas i svaret. Se TKB för beskrivning av profilerna P1-P5. Observera att profilen skall anges med versalt P. Datatyp: String. Exempel: P1
fromDate	Valfri parameter för det datum man är intresserad av förändringar från och med baserat på fältet version i personposten. Datatyp: LocalDateTime. Exempel: 2021-10-12T00:00:00
maxResultsPer File	Valfri parameter för max antal personposter per fil. Ej angivet så returneras alltid resultatet i endast en fil. Detta värde får ej vara lägre än 500. Datatyp: Integer. Exempel: 1000
primaryidentity	Valfri parameter som om satt kommer endast personposter som är huvudentitet inkluderas i svaret. Datatyp: Boolean. Exempel: true

2.1.1.2. Exempel

Via cURL

Request	Response
<pre>curl -X POST \ https://ws.pu.ineratest.org:443/purest/searchPersonsByFile \ -H 'content-type: multipart/form-data;' \ -F profile=P1 \ -F fromDate=2021-10-12T00:00:00 \ -F maxResultsPerFile=1000 \ -F primaryidentity=true \ -F file=@bilagan.csv</pre>	<p>Statuskod: 201 CREATED Content-Type: application/json; charset=ISO-8859-1 Transfer-Encoding: chunked Content:</p> <pre>{ "orderId": "1028-TO63-15133551" }</pre> <p>Statuskod: 400 BAD_REQUEST * Filen är inte av typ CSV * Filen är tom * De första raderna innehåller ogiltigt formaterad data * Den angivna profilen är inte giltig * Det angivna värdet för max antal personposter per fil är för lågt</p> <p>Statuskod: 401 UNAUTHORIZED * Konsumenten saknar behörighet till tjänsten</p> <p>Statuskod: 500 INTERNAL SERVER ERROR * Internt oförutsett fel * Filens storlek i MB överskrider tillåten gräns * Felaktig datatyp angiven för parameter</p>

2.2. Hämta resultat för personsök

Flödet startar med att ett urval för ett stort antal patienter behöver göras. Detta urval sker genom anrop till någon av de asynkron "ByOrder-tjänsterna" eller SearchPersonsByFile som alla returnerar ett Order-Id.

Fråga	Svar
-------	------

<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelop <soapenv:Header> <urn:LogicalAddress>SE165565594230-1000</urn:LogicalAddress> </soapenv:Header> <soapenv:Body> <urnl:SearchPersonsForProfileByOrder> <urnl:query>FROM PersonRecord.name WHERE givenName = 'Johan'</u <urnl:queryLanguage>SimpleQL</urnl:queryLanguage> <urnl:profile>Pl</urnl:profile> </urnl:SearchPersonsForProfileByOrder> </soapenv:Body> </soapenv:Envelope> </pre>	<pre> <soap:Envelope xmlns:soap="1 <soap:Body> <SearchPersonsForProf: <orderId>1201-T068< </SearchPersonsForPro: </soap:Body> </soap:Envelope> </pre>
---	---

Detta Order-Id används sedan som inparameter till tjänsten GetFilesForOrderId som returnerar ett svar med en eller flera länkar till resultatfilerna.

Fråga	Svar
<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/enve: <soapenv:Header> <urn:LogicalAddress>SE165565594230-1000</urn:LogicalAddress> </soapenv:Header> <soapenv:Body> <urnl:GetFilesForOrderId> <urnl:orderId>1201-T068-09391884</urnl:orderId> </urnl:GetFilesForOrderId> </soapenv:Body> </soapenv:Envelope> </pre>	<pre> <soap:Envelope xmlns:soap="http://schem <soap:Body> <GetFilesForOrderIdResponse xmlns <multimedia> <ns2:id>61a734b7de773d21d00< <ns2:mediaType>application/ <ns2:reference>https://ws.p </multimedia> </GetFilesForOrderIdResponse> </soap:Body> </soap:Envelope> </pre>

Då ordnarna som inkommer via dessa asynkrona tjänster läggs på kö så är det inte säkert att resultatet är färdigt ifall man frågar direkt efter att ordern har lagts. Tiden för ordena att bli klar varierar beroende på last på systemet och storleken på resultatet. Ett frågande system bör dock kunna förvänta sig ett svar inom fyra timmar.

Under tiden ordern inte är klar returnerar GetFilesForOrderId ett svar utan länkar/<multimedia>-stycke.

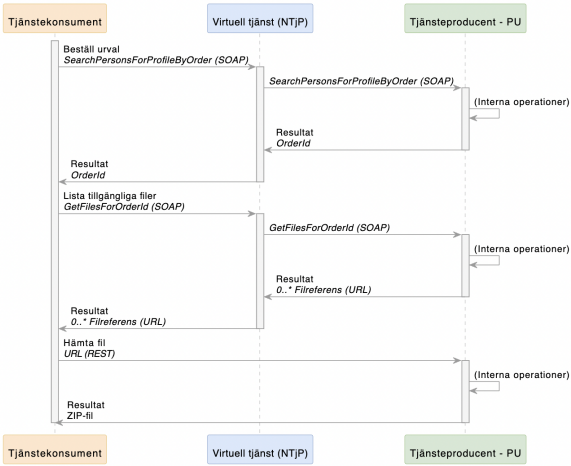
```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetFilesForOrderIdResponse xmlns:ns6="urn:riv:itintegration:registry:1" :
  </soap:Body>
</soap:Envelope>

```

OBS: Resultatfilerna är garanterat tillgängliga i ett dygn. Därefter rensas dom automatiskt bort. Filen kan bara hämtas en gång då de efter hämtning rensas bort.

Resultatfilerna är ZIP:ade och innehåller en XML-fil med det urval som efterfrågats. Formatet som specificeras i TKB och XSD för strategicresourcemanage ment:persons:person.



2.2.1. Exempel

2.2.1.1. SearchPersonsForProfileByOrder

Request	Response
<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap /envelope/" xmlns:urn="urn:riv:itintegration:registry:1" xmlns:urn1="urn:riv:strategicresourcemanagement: persons:person: SearchPersonsForProfileByOrderResponder:3"> <soapenv:Header> <urn:LogicalAddress>SE165565594230-1000</urn: LogicalAddress> </soapenv:Header> <soapenv:Body> <urn1:SearchPersonsForProfileByOrder> <urn1:query>FROM personrecord WHERE name. givenname LIKE 'Daniel%';</urn1:query> <urn1:queryLanguage>SimpleQL</urn1: queryLanguage> <urn1:profile>P5</urn1:profile> </urn1:SearchPersonsForProfileByOrder> </soapenv:Body> </soapenv:Envelope></pre>	<pre><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"> <S:Body> <ns2:SearchPersonsForProfileByOrderResponse xmlns:ns2="urn:riv:strategicresourcemanagement: persons:person: SearchPersonsForProfileByOrderResponder:3" xmlns:ns3="urn:riv:strategicresourcemanagement: persons:person:3" xmlns:ns4="urn:riv:itintegration:registry:1"> <ns2:orderId>0622-T017-09215997</ns2:orderId> </ns2:SearchPersonsForProfileByOrderResponse> </S:Body> </S:Envelope></pre>

2.2.1.2. GetFilesForOrderId

Request	Response
---------	----------

<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org /soap/envelope/" xmlns:urn="urn:riv:itintegration:registry:1" xmlns:urn1="urn:riv: strategicresourcemanagement:persons:person: GetFilesForOrderIdResponder:3"> <soapenv:Header> <urn:LogicalAddress>SE165565594230-1000< /urn:LogicalAddress> </soapenv:Header> <soapenv:Body> <urn1:GetFilesForOrderId> <urn1:orderId>0622-TO17-09215997</urn1: orderId> </urn1:GetFilesForOrderId> </soapenv:Body> </soapenv:Envelope></pre>	<p><u>Resultat ej klart:</u></p> <pre><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"> <S:Body> <ns2:GetFilesForOrderIdResponse xmlns:ns2="urn:riv:strategicresourcemanagement:persons: person:GetFilesForOrderIdResponder:3" xmlns:ns3="urn:riv:strategicresourcemanagement:persons: person:3" xmlns:ns4="urn:riv:itintegration:registry:1"> </ns2:GetFilesForOrderIdResponse> </S:Body> </S:Envelope></pre> <p><u>Resultat klart:</u></p> <pre><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"> <S:Body> <ns2:GetFilesForOrderIdResponse xmlns:ns2="urn:riv:strategicresourcemanagement:persons: person:GetFilesForOrderIdResponder:3" xmlns:ns3="urn:riv:strategicresourcemanagement:persons: person:3" xmlns:ns4="urn:riv:itintegration:registry:1"> <ns2:multipedia> <ns3:mediaType>application/zip</ns3:mediaType> <ns3:reference>https://ws.pu.ineratest.org:443 /purest/order/get/fd6a78ee-da71-4ec6-a099-cbccc21fe468</ns3: reference> </ns2:multipedia> </ns2:GetFilesForOrderIdResponse> </S:Body> </S:Envelope></pre>
---	---

2.2.1.3. REST

Request	Response
<p>https://ws.pu.ineratest.org:443/purest/order/get/fd6a78ee-da71-4ec6-a099-cbccc21fe468</p>	<p>Statuskod: 200 OK Content-Type: application/zip Content-Disposition: attachment; filename="<filnamn>" Transfer-Encoding: chunked Content: fil (zippat xmldata enligt strategicresourcemanagement:persons: person)</p> <p>Struktur för filnamn (både ZIP och XML): <orderId>_<datum>_<löpnummer>.zip (samt .xml) Exempel: 0622-TO17-09215997_20170622_1.zip</p> <p>Statuskod: 400 BAD_REQUEST * Ogiltig context path * HSA-id för konsument saknas * Certifikat för OrderId matchar inte certifikat som används vid hämtning av fil * Filen har redan hämtats</p> <p>Statuskod: 401 UNAUTHORIZED * Konsumenten saknar behörighet till tjänsten</p> <p>Statuskod: 404 NOT FOUND * Begärd referens (GUID) existerar inte</p> <p>Statuskod: 500 INTERNAL SERVER ERROR * Internt oförutsätt fel</p>

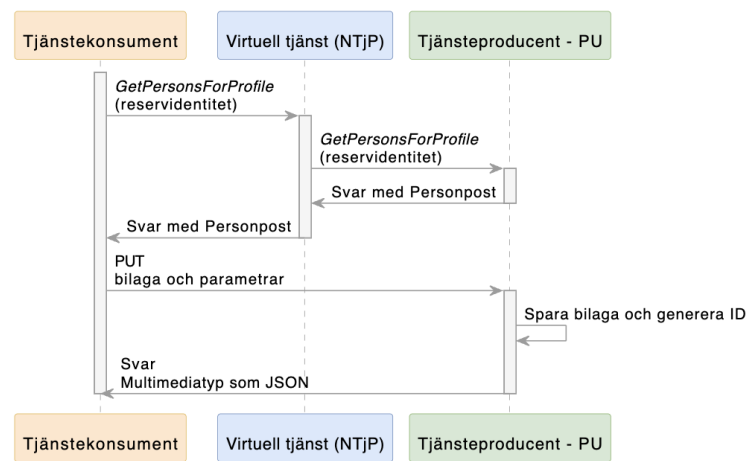
2.3. Bilagor för reservidentitet

En reservidentitet kan i PU-tjänsten ha 0..n bilagor. Dessa används för att styrka identiteten och kan vara exempelvis en kopia på ett pass. Bilagan kan laddas upp till reservidentiteten och senare kopplas till en specifik uppgift om styrkande av identitet. All operationer gällande bilagor görs med fördel i det av PU-tjänsten tillhandahålla GUI't, men kan vid behov hanteras av integrerande tjänster. För hantering av bilagor via GUI hänvisas till Användarhanboken för PU. Nedan beskrivs de flöden som appliceras vid integration av konsument.

Enbart exempel för REST-tjänster redovisas i följande kapitel. För information om användandet av RIV-tjänsterna hänvisas till TKB.

2.3.1. Ladda upp bilaga

Uppladdning av bilaga sker genom PUT eller POST och föregås av en hämtning av personpost för aktuell reservidentitet. Detta för att säkerställa att identiteten finns.



2.3.1.1. URL

Beror på miljö, men exempelvis: <https://ws.pu.ineratest.org:443/purest/attachment/put>

2.3.1.2. Parametrar

Namn	Beskrivning
actorRoot	root (OID) för aktör
actorExtension	värde för aktör
actorProfessionalRoot	root (OID) för aktörens organisation/vårdgivare
actorProfessionalExtension	värde för aktörens organisation/vårdgivare
patientRoot	root (OID) för patientidentitet
patientExtension	värde patientidentitet
file	bilagan

2.3.1.3. Exempel

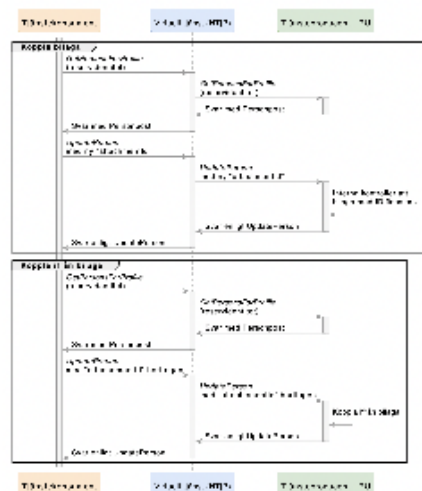
Via cURL

Request	Response
---------	----------

<pre>curl -X PUT \ https://ws.pu.ineratest.org:443/purest/attachment /put \ -H 'content-type: multipart/form-data; \ -F actorRoot=HSAID \ -F actorExtension=SE123456789 \ -F actorProfessionalRoot=HSAID \ -F actorProfessionalExtension=SE987654321 \ -F patientRoot=1.2.752.74.9.1 \ -F patientExtension=00002040AA11 \ -F file=@bilagan.jpg</pre>	<p>Statuskod: 200 OK Content-Type: application/json; charset=ISO-8859-1 Transfer-Encoding: chunked Content:</p> <pre>{ "id": "36a7068d-172b-4fee-bed5-13ecbdfa7ff2", "mediaType": "image/jpeg", "value": null, "reference": "https://ws.pu.ineratest.org:443/purest/attachment/get/36a7068d-172b-4fee-bed5-13ecbdfa7ff2" }</pre> <p>Statuskod: 400 BAD_REQUEST * Ogiltig context path * HSA-id för konsument saknas * Ogiltig patientidentitet / parametrar</p> <p>Statuskod: 401 UNAUTHORIZED * Konsumenten saknar behörighet till tjänsten</p> <p>Statuskod: 500 INTERNAL SERVER ERROR * Internt oförutsätt fel</p>
--	---

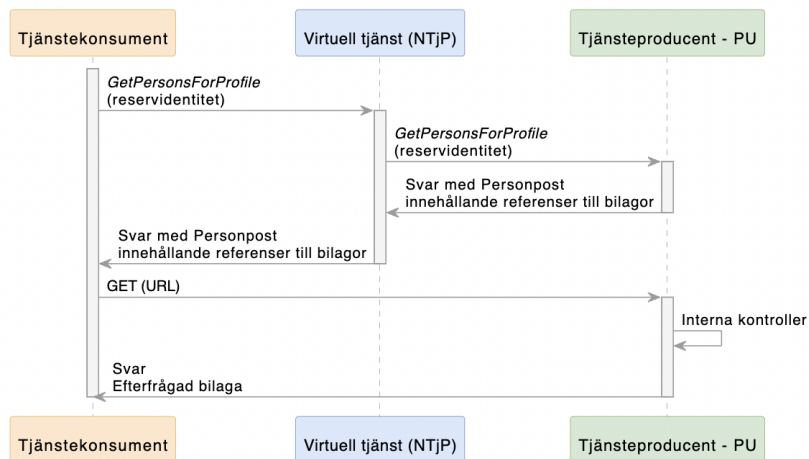
2.3.2. Koppla / Koppla ifrån bilaga

Dessa operationer är rena RIV-tjänster och beskrivs i TKB.



2.3.3. Hämta bilaga

Bilagor erhålls som en referens-URL i svaret från PU-tjänstens. Dessa används sedan som GET anrop för att hämta filerna.



2.3.3.1. URL

Beror på miljö, men exempelvis: <https://ws.pu.ineratest.org:443/purest/attachment/get/e870cddb-5842-4c72-a804-7d39fa4c5478>

2.3.3.2. Parametrar

Bilagans referensID som en del av URL

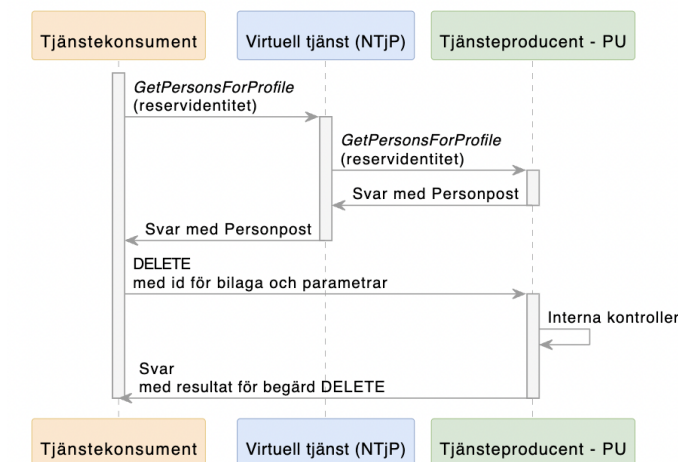
2.3.3.3. Exempel

Via cURL

Request	Response
curl -X GET \ https://ws.pu.ineratest.org:443/purest/attachment/get/e870cddb-5842-4c72-a804-7d39fa4c5478	<p>Statuskod: 200 OK Content-Type: <aktuell MIME-type> Content-Disposition: attachment; filename="<filnamn>" Transfer-Encoding: chunked Content: filen</p> <p>Statuskod: 400 BAD_REQUEST * Ogiltig context path * HSA-id för konsument saknas</p> <p>Statuskod: 401 UNAUTHORIZED * Konsumenten saknar behörighet till tjänsten</p> <p>Statuskod: 404 NOT FOUND * Begärd referens (GUID) existerar inte</p> <p>Statuskod: 500 INTERNAL SERVER ERROR * Internt oförutsätt fel</p>

2.3.4. Ta bort bilaga

Flödet förutsätter att bilagan inte har kvar några kopplingar inom reservidentiteten. Dvs flöde för "koppla ifrån bilaga" måste genomföras då det är applicerbart.



2.3.4.1. URL

Beror på miljö, men exempelvis: <https://ws.pu.ineratest.org:443/purest/attachment/delete>

2.3.4.2. Parametrar

Namn	Beskrivning
actorRoot	root (OID) för aktör
actorExtension	värde för aktör
actorProfessionalRoot	root (OID) för aktörens organisation/vårsgivare
actorProfessionalExtension	värde för aktörens organisation/vårsgivare
patientRoot	root (OID) för patientidentitet
patientExtension	värde patientidentitet
referenceId	bilagans id

2.3.4.3. Exempel

Via cURL.

Request	Response
<pre>curl -X DELETE \ https://ws.pu.ineratest.org:443/purest/attachment/delete \ -H 'content-type: multipart/form-data; \ -F actorRoot=HSAID \ -F actorExtension=SE123456789 \ -F actorProfessionalRoot=HSAID \ -F actorProfessionalExtension=SE987654321 \ -F patientRoot=1.2.752.74.9.1 \ -F patientExtension=00002040AA11 \ -F referenceId=e870cddb-5842-4c72-a804-7d39fa4c5478</pre>	<p>Statuskod: 200 OK Content-Type: text/plain; charset=iso-8859-1 Transfer-Encoding: chunked Content: File with id: e870cddb-5842-4c72-a804-7d39fa4c5478 was successfully deleted.</p> <p>Statuskod: 400 BAD_REQUEST * Ogiltig context path * HSA-id för konsument saknas * Ogiltig patientidentitet / parametrar</p> <p>Statuskod: 401 UNAUTHORIZED * Konsumenten saknar behörighet till tjänsten</p> <p>Statuskod: 404 NOT FOUND * Begärd referens (GUID) existerar inte</p> <p>Statuskod: 500 INTERNAL SERVER ERROR * Internt oförutsett fel</p>

3. Dokumentation

För mer detaljer kring tjänstekontrakten se tjänstekontraktsbeskrivningen för domänen strategicresourcemanagement:persons:person

Se: http://rivta.se/domains/strategicresourcemanagement_persons_person.html