



RIV TA Tjänsteschema 2.1

RIV Tekniska Anvisningar

Utgåva C
2012-01-03

*Utgåvehistorik*

<i>Utgåva</i>	<i>Revision Datum</i>	<i>Beskrivning</i>	<i>Ändringarna gjorda av</i>	<i>Definitiv revision fastställd av</i>
PA1	2011-04-27	Upprättat dokumentet baserat på version 2.0. Uppdaterat dokument för begärda ändringar enligt följande trackers på Osor: 15115.	Marcus.krantz@callistaenterprise.se	
A	2011-10-19	Revision fastställd		Arkitekturledningens tekniska expertgrupp, Center för eHälsa i samverkan
PC1	2011-12-16	Uppdateringar med anledning av flytt av projektplats från Osor till Google code. Endast länkar under rubriken Referenser uppdaterade	hans.thunberg@callistaenterprise.se	
C	2012-01-03	Revision fastställd		Arkitekturledningens tekniska expertgrupp, Center för eHälsa i samverkan

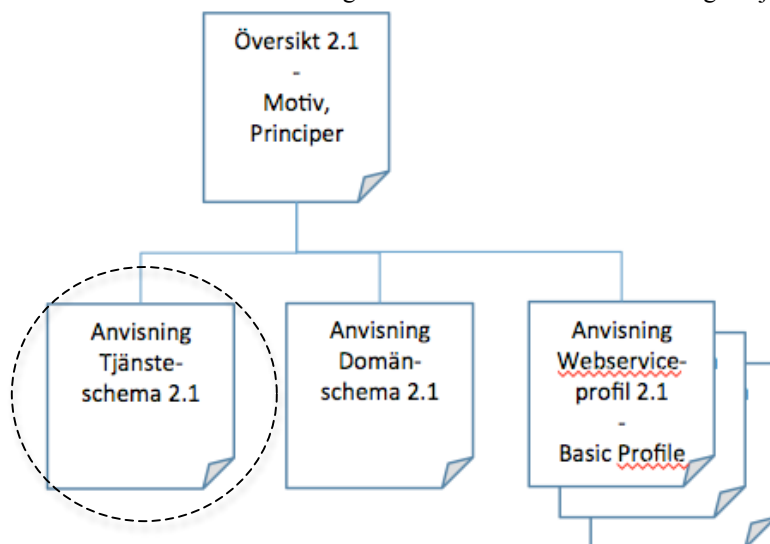
**INNEHÅLLSFÖRTECKNING**

1. INLEDNING	4
1.1. MÅLGRUPP	4
1.2. SYFTE	4
1.3. TILLGÄNGLIGHET	4
1.4. REFERENSER	4
2. BESKRIVNING AV NAMNREGLER	7
3. DETALJERADE REGLER	7
REGEL #1, DESIGNMÖNSTER FÖR TJÄNSTESCHEMAN	7
REGEL #2, NAMN PÅ XSD-FILEN	7
REGEL #3, NAMN PÅ TARGET NAMESPACE	7
REGEL #4, NAMN PÅ ELEMENT	7
REGEL #5, NAMN PÅ TYPER	8
REGEL #6, ANVÄNDNING AV SCHEMA-ATTRIBUTEN ELEMENTFORMDEFAULT OCH ATTRIBUTEFORMDEFAULT	8
REGEL #7, ANVÄNDNING AV SCHEMA-ATTRIBUTET VERSION	8
REGEL #8, ANVÄNDNING AV ANY-ELEMENT FÖR UTÖKNINGSBARHET	8
REGEL #9, BAKÅTKOMPATIBLA ELEMENT I UTÖKNINGSSCHEMA	8
REGEL #10 NATIONELLA TECKEN	10
REGEL #11, BEST-PRACTICE FÖR FELHANTERING	10

RIV TA Tjänsteschema 2.1

1. Inledning

Detta dokument beskriver regelverket RIV Tekniska Anvisningar Tjänsteschema 2.0.



1.1. Målgrupp

Denna anvisning riktar sig till dem som ska specificera XML-scheman för tjänstekontrakt i en nationell tjänsteinteraktion. Anvisningen innehåller endast regeluppsättningen. För bakgrund, motiv, krav samt de principer som ligger till grund för framtagning av reglerna hänvisas till Översikt RIV Tekniska Anvisningar 2.0 [R2].

1.2. Syfte

Syftet med denna anvisning är att beskriva designregler och namngivningsregler för interoperabilitet samt riktlinjer för att bygga in stöd för versionshantering i tjänstescheman.

Ett uttalat syfte med tjänstescheman är att de ska kunna användas oberoende av kommunikationsstandard. Många tjänstescheman kommer dock att skapas i syfte att importeras i WSDL-filer. Reglerna för tjänstescheman är därför baserade på WS-Basic Profiles regelverk för s.k. "Document/literal" [R4]. Målet med anvisningen är att optimera för interoperabilitet vid användning av web-service-baserade RIV TA-profiler utan att göra tjänstescheman beroende av web-services för transport och kuvertering

Exempel på tjänsteschema som följer denna anvisning finns på RIV-förvaltningens hemsida [R5] tillsammans med exempelapplikationer [R6].

1.3. Tillgänglighet

Detta dokument är publicerade under licensen Creative Commons CC-BY-SA (<http://creativecommons.org/licenses/by-sa/2.5/se/>). Det betyder att du fritt får kopiera, distribuera och skapa bearbetningar av anvisningarna, under förutsättning att upphovsmannen (Sveriges Kommuner och Landsting) anges (men inte på ett sätt som antyder att de godkännt eller rekommenderar din användning av verket).

Denna profil är verifieras genom exempelapplikationer. Källkoden [R9] för dessa distribueras under öppen-källkodslicensen Apache License, Version 2.0 (<http://www.apache.org/licenses/LICENSE-2.0>)

1.4. Referenser

Ref	Dokument	Beskrivning och ev. webbadress	Ansvarig
-----	----------	--------------------------------	----------



Ref	Dokument	Beskrivning och ev. webbadress	Ansvarig
[R1]	T-Boken	VIT-bokens tekniska arkitektur. Principer för uppbyggnad av den nationella arkitekturen i form av en teknisk referensarkitektur samt användningsfall med ett tekniskt perspektiv på realisering. Webblänk till PDF för REV A: http://www.cehis.se/images/uploads/dokumentarkiv/Referensarkitektur_vard_omsorg_VIT-bokens_tekniska_arkitektur_Rapport_110314_REV_B.pdf	Arkitekturledningens tekniska expertgrupp, SKL
[R2]	Översikt RIV Tekniska Anvisningar 2.1	Bakgrund, motiv, krav samt de principer som ligger till grund för utvecklingen av denna anvisning. Webblänk till PDF för översikten: http://code.google.com/p/rivta/source/browse/wiki/specs/RIV_Tekniska_Anvisningar_Oversikt_REV_B.pdf	Arkitekturledningens tekniska expertgrupp, SKL
[R3]	RIV Teknisk Anvisning Basic Profile 2.1	Exempel på anvisning för profil som pekar ut användningen av denna anvisning för specifikation av meddelandehåll (teknisk del). Webblänk till PDF för anvisningen: http://code.google.com/p/rivta/source/browse/wiki/specs/RIV_Tekniska_Anvisningar_Basic_profile_2.1.pdf	Arkitekturledningens tekniska expertgrupp, SKL
[R4]	WS-I Basic Profile	" Defines the WS-I Basic Profile 1.1, consisting of a set of non-proprietary Web services specifications, along with clarifications, refinements, interpretations and amplifications of those specifications which promote interoperability " Webblänk till WS-I Basic Profile: http://www.ws-i.org/Profiles/BasicProfile-1.1.html	The Web Services Interoperability Organization och ISO
[R5]	Exempel - Tjänsteinteraktion	All fragment av WSDL och XML-scheman som finns i detta dokument härrör ur den tjänsteinteraktion som ligger till grund för exempelapplikationerna för Java och .Net. http://rivta.googlecode.com/svn/RefApp/rivta-bp-21/java/cxf/trunk/rivta-bp21-refapp-schemas/src/main/resources/schemas/interactions/MakeBookingInteraction/	Arkitekturledningens tekniska expertgrupp, SKL
[R6]	Exempel – konsument och producent i Java och .Net	Referensapplikationerna syftar till att vara ett generellt underlag för den utvecklare som ska utveckla en tjänstekonsument eller en tjänsteproducent för en tjänsteinteraktion som följer denna profil. Det är en målsättning att detta ska avlasta nationella projekt från att ta fram projektspecifika kodexempel för varje nationell tjänsteinteraktion som specificeras enligt denna profil. Webblänk till hemsida för exempelapplikationer: http://code.google.com/p/rivta/wiki/FrontPage	Arkitekturledningens tekniska expertgrupp, SKL
[R7]	Beskrivning av "Venetian Blind"	Dokumentet beskriver det designmönster som tillämpas för XML Schema design i denna anvisning. Webblänk till hemsidan:	Okänd.



Ref	Dokument	Beskrivning och ev. webbadress	Ansvarig
		http://www.xfront.com/GlobalVersusLocal.html#ThirdDesign	
[R8]	RIV Metodanvisningar	<p>Beskriver den nationellt fastställda metoden för att utarbeta meddelanden och databasmodeller. Metoden tar sin utgångspunkt i referensmodeller för informationsstruktur (VTIM). Genom RIV-metoden utarbetas de meddelandestrukturer som beskriver de informationsobjekt som är parametrar och resultat för de operationer som definieras i tjänsteinteraktioner. De XML-Scheman som skapas ur RIV-metoden importeras vanligen i tjänstescheman i syfte att vara "type" för de request- och response-element som bygger upp tjänstekontrakten i tjänsteinteraktionen.</p> <p>Webblänk till dokumentet: http://www.cehis.se/images/uploads/dokumentarkiv/RIV_21_Anvisning_Regelverk_110120.pdf</p>	Arkitekturledningen, SKL
[R9]	W3C-rapport om utökningsbara XML- scheman	<p>Beskriver problemställningar och strategier för design av meddelanden som ger bra stöd för versionshantering. Versioneringsstrategin som beskrivs i denna översikt och som tillämpas i RIV Teknisk Anvisning Tjänsteschema är baserad på strategi nr 2.5 i denna rapport.</p> <p>Webblänk till rapportens hemsida: http://www.w3.org/2001/tag/doc/versioning-xml</p>	W3C

2. Beskrivning av namnregler

Namngivningsregler i detta dokument är formulerade enligt följande uppställning:

1. Tjänstedomänens namn: **$\{\text{tjänsteDomän}\}$** , t ex crm:scheduling
2. Tjänsteinteraktionens namn: **$\{\text{tjänsteInteraktion}\}$** , t ex MakeBooking
3. Tjänsteinteraktionsroll: **$\{\text{roll}\}$** = Initiator eller Responder, motsvarande tjänsteinteraktionsroller initiativtagare och utförare
4. Tjänsteinteraktionens version:
m.n = förkortning av **$\{\text{majorVersion}\}$** . **$\{\text{minorVersion}\}$**
m = förkortning av **$\{\text{majorVersion}\}$**
5. Operationens namn: **$\{\text{operation}\}$** , t ex MakeBooking

3. Detaljerade regler

Regel #1, designmönster för tjänstescheman

"Venetian Blind" [R7] *skall* användas som designmönster för tjänstescheman. Designmönstret Venetian Blind innebär följande:

- Den interna strukturen i ett meddelande byggs upp med hjälp av globalt deklarerade typer. Med "globalt deklarerade" menas deklarationer som görs direkt under schema-elementet.
- Endast rotelementet är deklarerat som ett globalt element. I web-service-fallet innebär det att request- och response-elementen är globalt deklarerade element medan resten är typer.

Anm. I vissa fall kan även andra element än request och response behöva vara globala, t ex används element-referenser till globala element i importerade scheman för att stödja versioneringsstrategin, se nedan.

Motiv: Interoperabilitet, WS-I Basic Profile

Exempel: Tjänstekontrakten för exempelapplikationerna [R5]

Regel #2, namn på xsd-filen

Schema-filen för ett tjänstekontrat *bör* namnges enligt följande regel: **$\{\text{tjänsteInteraktion}\}\{\text{roll}\}_{\text{m.n}}.xsd$**

Motiv: Att ha med versionsnummer i namnet på källkodsfiler är generellt sett något man försöker undvika då det försvårar användning av versionshanteringsverktyg (t ex Subversion, Microsoft Visual Studio). I fallet med tjänsteschema behöver man dock kunna hantera flera olika versioner samtidigt (i byggsystem mm) och för att underlätta den hanteringen ingår versionsnumret i filnamnet på tjänstekontrakt.

Anm. Detta gäller principiellt sett också de XML Schema som importeras/inkluderas av att tjänsteschema och som beskriver RIV Meddelanden men denna anvisning täcker inte in utformning av dessa XML Scheman, se [R8].

Exempel: MakeBookingResponder_1.0.xsd

Regel #3, namn på target namespace

Attributet targetNamespace på schema-elementet *skall* ha ett värde som definieras av följande regel:

$\text{urn:riv}:\{\text{tjänsteDomän}\}:\{\text{tjänsteInteraktion}\}\{\text{roll}\}:\{\text{m}\}$

Motiv: Användningen av major-version i namnrymden är en av att följa fastslagen versioneringsstrategi [R9]. Att ha en unik namnrymd per tjänstekontrakt (tjänsteinteraktion + roll) är en förutsättning för att följa WS-I Basic Profiles [R4] regel om "operation signature". Det också generellt goda förutsättningar för att implementera generella bryggor och tjänsteväxlar

Exempel: urn:riv:crm:scheduling:MakeBookingResponder:1

Regel #4, namn på element

Attributet "name" på element som deklarerar request-element i tjänsteschemat *skall* ha ett värde som följer följande regel:

$\{\text{operation}\}$, t ex: MakeBooking

Attributet "name" på element som deklarerar response-element i tjänsteschemat *skall* ha ett värde som följer följande regel:

$\{\text{operation}\}$ Response, t ex: MakeBookingResponse



Motiv: För konsistent namngivning skall element för in-parametrar ha samma namn som operationen.

Exempel: "MakeBooking" respektive "MakeBookingResponse"

Regel #5, namn på typer

Attributet "name" på element som deklarerar request-typer i tjänsteschemat *bör* ha ett värde som följer följande regel:

`{operation}Type`

Attributet "name" på element som deklarerar response-typer i tjänsteschemat *skall* ha ett värde som följer följande regel:

`{operation}ResponseType`

Motiv: Enhetlighet.

Exempel: "MakeBookingType" respektive "MakeBookingResponseType"

Regel #6, användning av schema-attributen `elementFormDefault` och `attributeFormDefault`

Schema-attributen `elementFormDefault` och `attributeFormDefault` *skall* sättas till "qualified" respektive "unqualified".

Motiv: För att versioneringsstrategin skall fungera är det viktigt att alla element i instans-dokument är namespace-qualified. Detta uppnås genom att sätta schema-attributet `elementFormDefault` till "qualified".

Exempel: Tjänstekontrakten för exempelapplikationerna [R5]

Regel #7, användning av schema-attributet `version`

Schema-attributet `version` *bör* sättas till "m.n"

Motiv: Då namnrymden inte innehåller minor-version, ger detta en dokumentation som följer intentionen med attributet.

Exempel: `<schema ... version="1.0">`

Regel #8, användning av `any-element` för utökningsbarhet

För att uppnå framåtkompatibilitet skall ett `xsd:any` element läggas in sist i alla komplexa typer som ska kunna utökas, exempel:

Motiv: För att uppnå framåtkompatibilitet måste man "förbereda" sina XML scheman för framtida utökningsbarhet. Detta är en del av den tillämpade strategin för versionering [R9].

Exempel:

```
<xsd:complexType name="SomeType">
  <xsd:sequence>
    <xsd:element name="someElement" type="xsd:string" />
    <xsd:element name="someOtherElement" type="xsd:int" />
    <xsd:any processContents="lax" minOccurs="0" maxOccurs="unbounded" namespace="##other"/>
  </xsd:sequence>
</xsd:complexType>
```

Regel #9, bakåtkompatibla element i utökningsschema

För att skapa en ny minor-version av ett tjänsteschema, *skall* följande regler följas:

- De nya icke-obligatoriska elementen läggs till i befintligt schema närmast före `any-element` i den komplexa typ som ska utökas. Dessa nya element har ingen typ, utan refererar (`xsd:ref="..."`) element som är rotelement i en ny schema-fil (*utöknings-schema*)
- Definieras de nya icke-obligatoriska elementen i en ny schema-fil (utökningsschema) med ett namn som följer följande regel: `{tjänsteInteraktion}_{roll}_{m.n}_ext.xsd`
- Utökningsschemat ska ha en `targetNamespace` enligt följande regel: `urn:riv:{tjänsteDomän}:{tjänsteInteraktion}_{roll}:{m.n}`
- Tjänsteschemat importerar (`xsd:import`) utöknings-schemat som ges namnrymdsalias enligt följande regel: `m${n}`
- Tjänsteschemats versionsattribut ändras till den nya minor-versionen.
- I nästa major-version av tjänsteschemat flyttas element-deklarationerna in från alla utökningsscheman (det finns ett för varje minor-version som tillkommit sedan förra major-versionen skapades).

Motiv: Detta förfarande är en konsekvens av vald strategi för versionering [R9]. Se [R2] för ytterligare bakgrund.

Exempel: Följande exempel är baserat på en delmängd av tjänsteschemat för exempelapplikationerna. Det visar en ny minor-version, samt dess utökningsschema.

Tjänsteschema, ny minorversion: **GetAvailableTimeslotsResponder_1.1.xsd**

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:core="urn:riv:crm:scheduling:1"
  xmlns:tns="urn:riv:crm:scheduling:GetAvailableTimeslotsResponder:1"
  xmlns:m1="urn:riv:crm:scheduling:GetAvailableTimeslotsResponder:1.1"
  targetNamespace="urn:riv:crm:scheduling:GetAvailableTimeslotsResponder:1" elementFormDefault="qualified"
  attributeFormDefault="unqualified" version="1.1">
```

```
<xs:import namespace="urn:riv:crm:scheduling:GetAvailableTimeslotsResponder:1.1"
  schemaLocation="GetAvailableTimeslotsResponder_1.1_ext.xsd" />
```

```
  <xs:element name="GetAvailableTimeslots" type="tns:GetAvailableTimeslotsType" />

  <xs:complexType name="GetAvailableTimeslotsType">
    <xs:sequence>
      <xs:element name="healthcare_facility" type="core:HsaIdType" minOccurs="1" maxOccurs="1" />
      <xs:element name="bookingId" type="core:BookingIdType" minOccurs="0" maxOccurs="1" />
      <xs:element name="startDateInclusive" type="core:DT" minOccurs="1" maxOccurs="1" />
      <xs:element name="endDateInclusive" type="core:DT" minOccurs="1" maxOccurs="1" />
      <xs:element name="performer" type="core:HsaIdType" minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="timeTypeName" type="xs:string" maxOccurs="1" minOccurs="0" />
      <xs:element name="timeTypeID" type="core:TimeTypeIDType" minOccurs="0" maxOccurs="1" />
      <xs:element name="careTypeName" type="xs:string" maxOccurs="1" minOccurs="0" />
      <xs:element name="careTypeID" type="core:CareTypeIDType" minOccurs="0" maxOccurs="1" />
      <xs:element ref="m1:subject_of_care" minOccurs="1"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Utökningsschema med element som tillkommit i 1.1: **GetAvailableTimeslotsResponder_1.1_ext.xsd**

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:core="urn:riv:crm:scheduling:1"
  xmlns:tns="urn:riv:crm:scheduling:GetAvailableTimeslotsResponder:1.1"
  targetNamespace="urn:riv:crm:scheduling:GetAvailableTimeslotsResponder:1.1"
  elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.1">

  <xs:element name="subject_of_care" type="core:SubjectOfCareIdType"/>

</xs:schema>
```

Vid nästa major-version (i detta exempel med ett nytt obligatoriskt element) integreras elementen från mellanliggande utökningsscheman i huvudschemat: **GetAvailableTimeslotsResponder_2.0.xsd**

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:core="urn:riv:crm:scheduling:1"
  xmlns:tns="urn:riv:crm:scheduling:GetAvailableTimeslotsResponder:2"
  targetNamespace="urn:riv:crm:scheduling:GetAvailableTimeslotsResponder:2"
  elementFormDefault="qualified" attributeFormDefault="unqualified" version="2.0">

  <xs:element name="GetAvailableTimeslots" type="tns:GetAvailableTimeslotsType" />

  <xs:complexType name="GetAvailableTimeslotsType">
    <xs:sequence>
      <xs:element name="healthcare_facility" type="core:HsaIdType" minOccurs="1" maxOccurs="1" />
      <xs:element name="bookingId" type="core:BookingIdType" minOccurs="0" maxOccurs="1" />
      <xs:element name="startDateInclusive" type="core:DT" minOccurs="1" maxOccurs="1" />
```



```
<xs:element name="endDateInclusive" type="core:DT" minOccurs="1" maxOccurs="1" />
<xs:element name="performer" type="core:HsaIdType" minOccurs="0" maxOccurs="unbounded" />
<xs:element name="timeTypeName" type="xs:string" maxOccurs="1" minOccurs="0" />
<xs:element name="timeTypeID" type="core:TimeTypeIDType" minOccurs="0" maxOccurs="1" />
<xs:element name="careTypeName" type="xs:string" maxOccurs="1" minOccurs="0" />
<xs:element name="careTypeID" type="core:CareTypeIDType" minOccurs="0" maxOccurs="1" />
<xs:element name="subject_of_care" type="subject_of_care" minOccurs="1"/>
<xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:schema>
```

Regel #10 Nationella tecken

Tjänstescheman *ska* undvika att använda nationella tecken i såväl elementnamn, attributnamn som vid listning av värdeomängder för uppräknings typer. Följande exempel bör därför undvikas:

```
<xs:simpleType name=" " >
  <xs:annotation>
    <xs:documentation>...</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="Återställas helt" />
    <xs:enumeration value="Återställas delvis" />
    <xs:enumeration value="Det går inte att bedöma" />
  </xs:restriction>
</xs:simpleType>
```

Motiv: För att undvika interoperabilitetsproblem bör man ej använda sig av nationella tecken när man definierar typer som kommer att användas i ett tjänstekontrakt. Ofta uppstår annars fel vid kodgenerering från schemat.

Regel #11, Best-practice för felhantering

Ett tjänstekontrakt *ska* inte definiera några egna fel (SOAP-exceptions). Istället bör följande struktur för felhantering tillämpas i svarsmeddelandet (för fråga-svar):

```
<xs:complexType name="MakeBookingResponseType">
  <xs:sequence>
    <xs:element name="bookingId" type="core:BookingIdType" minOccurs="0" maxOccurs="1" />
    <xs:element name="resultCode" type="tns:ResultCodeEnum" minOccurs="1" maxOccurs="1" />
    <xs:element name="resultText" type="xs:string" minOccurs="0" maxOccurs="1" />
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="ResultCodeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="OK"/>
    <xs:enumeration value="ERROR"/>
    <xs:enumeration value="INFO"/>
  </xs:restriction>
</xs:simpleType>
```

Vid ett tekniskt fel levereras ett generellt undantag (SOAP-Exception). Exempel på felsituationer som rapporteras som tekniskt fel kan vara deadlock i databasen eller följd effekter av programmeringsfel. Denna information bör loggas av tjänstekonsumenten. Informationen är inte riktad till användaren. Användaren kommer enbart att se "tekniskt fel – inte detaljinformation. Den riktar sig till systemförvaltaren. Vid ett logiskt fel i de uppdaterande tjänsterna levereras resultCode, resultText. Syftet med resultText är att tjänstekonsumenten av tjänsten ska kunna visa upp informationen för invånaren. resultCode kan vara:

OK

Transaktionen har utförts enligt uppdraget i frågemeddelandet.

*INFO*

Transaktionen har utförts enligt uppdraget i frågemeddelandet, men det finns ett meddelande som tjänstekonsumenten måste visa upp för invånaren. Exempel på detta kan vara ”kom fastande”.

ERROR

Transaktionen har INTE kunnat utföras enligt uppdrag i frågemeddelandet p.g.a. logiskt fel. Det finns ett meddelande som konsumenten måste visa upp (om tillämpligt, annars t.ex. skrivs i batch-log). Exempel på detta kan vara ”tiden har blivit upptagen av annan patient” (från tjänstedomän Invånarens tidbokning).

Motiv: Erfarenheter har visat att felhantering genom egen-definierade fel skapar interoperabilitetsproblem och försvårar hantering i intermediärer.

Exempel: Se tjänsteschemat i referensapplikationen.