

2013-02-08

# **SHS Version 2.0 SOAP-based Protocol Riktlinjer för Tjänsteschema**

**Försäkringskassan - Swedish Social Insurance Agency**

Utgåva PA3  
2013-02-08

Copyright © 2012, 2013 Swedish Social Insurance Agency. All Rights Reserved.



2013-02-08

## Utgåvehistorik

<i>Utgåva</i>	<i>Revision Datum</i>	<i>Beskrivning</i>	<i>Ändringarna gjorda av</i>	<i>Definitiv revision fastställd av</i>
PA2	2012-05-02	Dokumentet skapat baserat på RIV TA Domänschema 2.1	<a href="mailto:Anders.asplund@callistaenterprise.se">Anders.asplund@callistaenterprise.se</a>	
	2012-10-12	SHS Version 2.0, efter 1:a remiss		
PA3	2013-02-08	Slutgiltig version SHS 2.0		

2013-02-08

**INNEHÅLLSFÖRTECKNING**

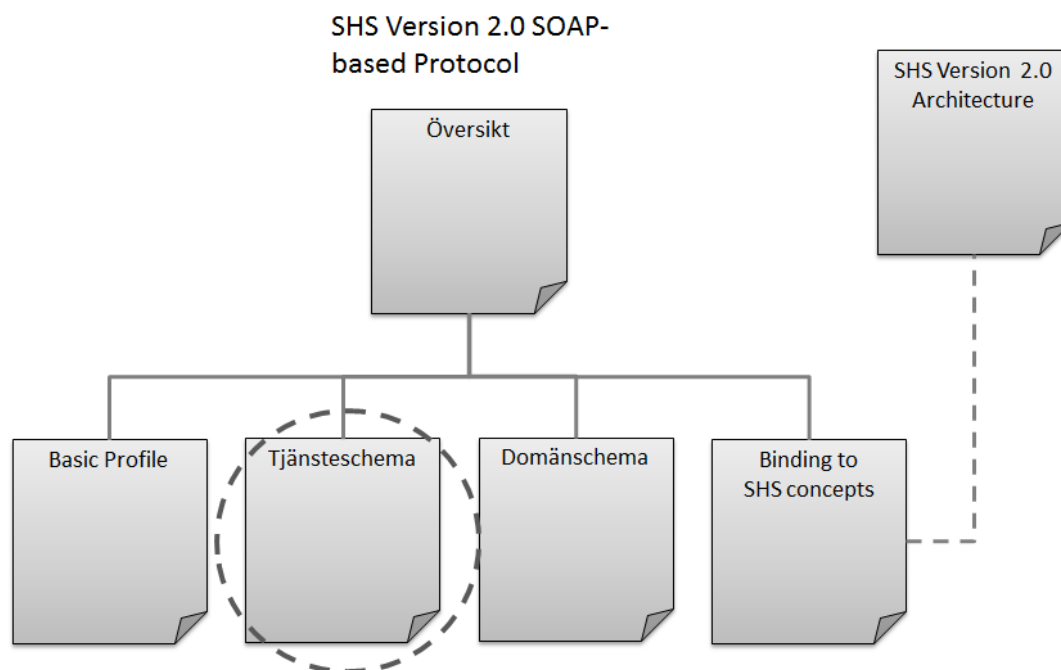
<b>1. INLEDNING</b>	<b>4</b>
1.1. MÅLGRUPP	4
1.2. SYFTE	4
1.3. TILLGÄNGLIGHET	4
1.4. UPPHOVSMAN	4
1.5. REFERENSER	5
<b>2. BESKRIVNING AV NAMNREGLER</b>	<b>6</b>
<b>3. DETALJERADE REGLER</b>	<b>6</b>
REGEL #1, DESIGNMÖNSTER FÖR TJÄNSTESCHEMAN	6
REGEL #2, NAMN PÅ XSD-FILEN	6
REGEL #3, NAMN PÅ TARGET NAMESPACE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

2013-02-08

# SHS Tjänsteschema

## 1. Inledning

Detta dokument beskriver regelverket SHS Version 2.0 SOAP-based Protocol Tjänsteschema.



### 1.1. Målgrupp

Denna anvisning riktar sig till dem som ska specificera XML-scheman för tjänstekontrakt för kommunikation med en SHS-nod eller mellan SHS-noder. Anvisningen innehåller endast regeluppsättningen. För bakgrund, motiv, krav samt de principer som ligger till grund för framtagning av reglerna hänvisas till SHS Version 2.0 SOAP-based Protocol Översikt [R1].

### 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" [R3]. Målet med anvisningen är att optimera för interoperabilitet vid användning av web-service-baserade SHS-profiler utan att göra tjänstescheman beroende av web-services för transport och kübertering

### 1.3. Tillgänglighet

Specifikationerna för SHS Version 2.0 publiceras på Försäkringskassans hemsida [www.fk.se](http://www.fk.se).

### 1.4. Upphovsman

SHS Version 2.0 SOAP-based Protocol är framtagen av Försäkringskassan och baserad på RIV TA version 2.1 med upphovsman SKL(Sveriges Kommuner och Landsting).



2013-02-08

## 1.5. Referenser

Ref	Dokument	Beskrivning och ev. webbadress	Ansvarig
[R1]	SHS Version 2.0 SOAP-based Protocol Översikt	Bakgrund, motiv, krav samt de principer som ligger till grund för utvecklingen av denna anvisning.	Försäkringskassan
[R2]	SHS Version 2.0 SOAP-based Protocol Basic Profile	Exempel på anvisning för profil som pekar ut användningen av denna anvisning för specifikation av meddelandehåll (teknisk del).	Försäkringskassan
[R3]	WS-I Basic Profile	<i>" 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 "</i> Webblänk till WS-I Basic Profile: <a href="http://www.ws-i.org/Profiles/BasicProfile-1.1.html">http://www.ws-i.org/Profiles/BasicProfile-1.1.html</a>	The Web Services Interoperability Organization och ISO
[R4]	Beskrivning av "Venetian Blind"	Dokumentet beskriver det designmönster som tillämpas för XML Schema design i denna anvisning. Webblänk till hemsidan: <a href="http://www.xfront.com/GlobalVersusLocal.html#ThirdDesign">http://www.xfront.com/GlobalVersusLocal.html#ThirdDesign</a>	Okänd.
[R5]	W3C-rapport om utökningsbara XML-scheman	Beskriver problemställningar och strategier för design av meddelanden som ger bra stöd för versionshantering. Versioneringsstrategin som beskrivs i denna översikt och som tillämpas i SHS Version 2.0 SOAP-based Protocol Tjänsteschema är baserad på strategi nr 2.5 i denna rapport. Webblänk till rapportens hemsida: <a href="http://www.w3.org/2001/tag/doc/versioning-xml">http://www.w3.org/2001/tag/doc/versioning-xml</a>	W3C



2013-02-08

## 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:  
 $\text{m.n}$  = förkortning av  $\{\text{majorVersion}\}.\{\text{minorVersion}\}$   
 $\text{m}$  = förkortning av  $\{\text{majorVersion}\}$
5. Operationens namn:  $\{\text{operation}\}$ , t ex MakeBooking

## 3. Detaljerade regler

### Regel #1, designmönster för tjänstescheman

"Venetian Blind" [R4] *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 deklarerationer 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 schan för att stödja versioneringsstrategin, se nedan.*

**Motiv:** Interoperabilitet, WS-I Basic Profile

**Exempel:**

### 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 SHS Meddelanden men denna anvisning täcker inte in utformning av dessa XML Scheman.*

**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 [R5]. Att ha en unik namnrymd per tjänstekontrakt (tjänsteinteraktion + roll) är en förutsättning för att följa WS-I Basic Profiles [R3] 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



2013-02-08

**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:**

## 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 [R5].

**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`-elementet 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 (*utökningsschema*)
- 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}:${m.n}`
- Tjänsteschemat importerar (`xsd:import`) utökningsschemat som ges namnrymdsalias enligt följande regel: `m${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 versionshantering [R5]. Se [R1] för ytterligare bakgrund.

**Exempel:**



2013-02-08

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

<xs:import namespace="urn:shs: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:shs:crm:scheduling:1"
xmlns:tns="urn:shs:crm:scheduling:GetAvailableTimeslotsResponder:1.1"
targetNamespace="urn:shs: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:shs:crm:scheduling:1"
xmlns:tns="urn:shs:crm:scheduling:GetAvailableTimeslotsResponder:2"
targetNamespace="urn:shs: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>
```





2013-02-08

## Regel #10 Nationella tecken

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

```
<xs:simpleType name="Å">
  <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.