Mawell | more time to care

Nationell Listningstjänst

Bas - Utvecklingsstöd

skapad för:

SVR

skapad av:

Robert Georén

Version: 1.2
Datum: 2010-05-19

Förändringar		
Vem	**När**	**Vad**
Robert Georén	2009-05-11	Skapade dokumentet, version 1.0.
Robert Georén	2009-09-15	Ändrade i ’göra tjänsteval’ anropet, version 1.1.
Robert Georén	2009-12-28	Uppdaterat efter att ha tagit bort EN13606. Version 1.2
Daniel Berggren	2010-01-26	Lagt till exempelkod för getPersonQueueStatus
Daniel Berggren	2010-02-10	- Ändrat kod för ”Göra tjänsteval” samt ”Hämta köinformation för en person” - Ändrat ”Generera proxy för tjänsten”
Daniel Berggren	2010-02-18	Ändrat meddelande för ”hämta tjänsteval” adderat ”namn” för facility och resource

adress
Mawell
Solna Torg 3, 3 tr
171 45 Solna

telefon
+46 (0)8 527 400 00
fax
+46 (0)8 527 400 40

e-post
contact@mawell.com
hemsida
www.mawell.com

organisationsnr
556582-5634

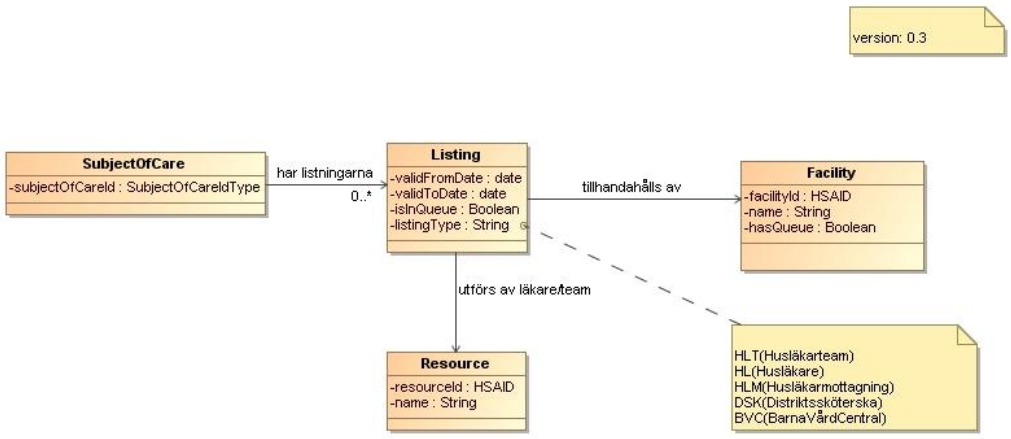
Innehåll

1. Inledning

Syftet med dokumentet är att underlätta för systemutvecklare som ska utveckla en Anslutningspunkt eller Konsument för den nationella listningstjänsten.

2. Arkitektur/Design

Domänmodell för Nationel listningstjänst.



** Domänmodell för Nationell Listningstjänst.*

2.1 Nationella anvisningar

Det finns nationella anvisningar för hur nationella tjänster bör implementeras och en av dem är RIV (se ref[1]). RIVs syfte är bl.a. att beskriva hur man realiserar utbyte av information mellan två parter. Följande avsnitt beskriver vad en Producent kan behöva veta för att realisera en anslutningspunkt.

2.1.1 WSDL och RIV

Det finns många regler för hur WSDL filerna ska utformas i enlighet med RIV Basic Profile 2.0. För att få en förståelse varför WSDL filerna ser ut som de gör, se referens [1]. Här följer en kortfattad summering av reglerna.

1. Regel: SOAP 1.1 ska användas.
2. Regel: SOAP meddelanden skall använda document/literal style.
3. Anpassning utökningsbarhet: <xsd:any> användas för att uppnå utökningsbarhet.

2.1.2 “Contract-first Development”

WSDL filen utgör kontraktet mellan Konsument och Producent och RIV ger riktlinjen att använda det designmönstret. Detta designmönster anses även som best-practice eftersom det skyddar

Konsumenter ifrån kodförändringar i tjänsteimplementationen och skillnader som finns mellan verktygen som genererar WSDL filer ifrån kod.

2.1.3 Riktlinjer ifrån VIT Boken

Nationella Listningstjänsten definieras som en *fråga-svar* tjänsteinteraktion enligt VIT-boken. Ett icke-funktionellt krav på producenter är att loggning av anropen till tjänsten ska göras, detta främst för att kunna mäta svarstider på tjänsten.

3. Utvecklingsstöd för Anslutningspunkter - Java

Här följer några tips vid utveckling av en Anslutningspunkt.

- Generera server artifakter med ett verktyg så att de blir JSR 181 konformt (se ref[12]). wsgen ingår i JDK 1.6 men andra alternativ är t.ex. Apache CFX (ref [7]) eller Metro (ref [8]). I dagsläget (2010) så har CXF bäst verktyg för att generera proxy/bidningsklasser, vilket då rekommenderas.
- Använd anvisningar i Meddelandestruktur (ref [5]) för att ta reda på vad som skall returneras till Konsumenter.

4. Utvecklingsstöd för Konsumenter - Java

4.1 Generera proxy för tjänsten

Det rekommenderade sättet att använda en webservice av denna karaktär är att generera proxys för tjänsten och paketera proxy'na i en .jar fil. Använd ditt favorit verktyg för proxy generering (se ref [7] och [8]).

För att virtualiseringstjänsten ska veta vem som ska utföra en fråga så måste HSAID anges vid varje tjänsteinteraktion.

```
AttributedURIType logicalAddress = new AttributedURIType();

logicalAddress.setValue("01");
```

4.2 Hämta tjänsteval – JAXWS-RI

Nedan följer Java kod för användningsfallet "Hämta tjänsteval".För att se XML datat som finns i SOAP Body, se ref [9].

```
package com.mawell.nlt.consumer;

import java.net.MalformedURLException;
import java.net.URL;

import javax.xml.namespace.QName;
import javax.xml.ws.Service;

import org.w3._2005._08.addressing.AttributedURIType;

import riv.crm.carelisting._1.Facility;
import riv.crm.carelisting._1.Resource;
import riv.crm.carelisting._1.SubjectOfCare;
import riv.crm.carelisting.getlisting._1.rivtabp20.GetListingResponderInterface;
import riv.crm.carelisting.getlisting._1.rivtabp20.PersonNotFound;
```

```
import riv.crm.carelisting.getlisting._1.rivtabp20.TechnicalException;
import riv.crm.carelisting.getlistingresponder._1.GetListingRequestType;
import riv.crm.carelisting.getlistingresponder._1.GetListingResponseType;


/**
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations
 * under the License.
 * Exempelkod för användningsfallet "Hämta tjänsteval".
 *
 * @author Robert Siwerz, www.mawell.com.
 */
public class UseCaseHamtaTjansteval
{

   private void useCaseHamtaVardval() throws MalformedURLException
   {
      // Hämtar referens till SEI (Service Endpoint Interface).
      QName serviceName = new QName("urn:riv:crm:carelisting:GetListing:1:rivtabp20",
"GetListingResponderService");
      Service service = Service.create(new
URL("http://127.0.0.1:8088/mockGetListingResponderBinding?WSDL"), serviceName);

      GetListingResponderInterface listingSEI = service.getPort(GetListingResponderInterface.class);

      // Hämtar listningsinformation för angiven person.
      AttributedURIType logicalAddress = new AttributedURIType();
      logicalAddress.setValue("01"); // Områdeskod

      GetListingRequestType request = new GetListingRequestType();
      request.setPersonId("195005055005");
      GetListingResponseType response = null;

      try
      {
         response = listingSEI.getListing(logicalAddress, request);
      } catch (PersonNotFound e)
      {
         // Använd affärsregel för att hantera detta.
      } catch (TechnicalException e)
      {
         // Gör ett nytt försök...
      }

      // Skriver ut debug information
      SubjectOfCare patientData = response.getSubjectOfCare();
      // 1. Personnummer
      System.out.println("1. Sökandes personnummer: " + patientData.getPersonId());

      // 2. Tjänsteutövarens (t.ex. Vårdenhet) HSAID
      String hsaID = patientData.getListing().get(0).getHealthcareFacility().getFacilityId();
      System.out.println("2. Tjänsteutövaren (HSAID): " + hsaID);

      // 3. Hämtar mer detaljer om tjänsteutövaren.
      Facility facility = patientData.getListing().get(0).getHealthcareFacility();
      Resource resource = patientData.getListing().get(0).getResource();

      if(facility != null)
      {
         // tjänsteutövaren är en vårdenhet och innehåller följande data.
```

```
          System.out.println(" Vårdenhet:");
          // 1. Namn.
          System.out.println(" 1. Namn: " + facility.getFacilityName());

         // 2. Har vårdenheten kö just nu.
         String queue = (facility.isHasQueue())? "Ja" : "Nej";
         System.out.println(" 2. Är det kö just nu: " + queue);
      }
      if (resource != null)
      {
         // Tjänsteutövaren är en specifik läkare och innehåller följande
         System.out.println(" Vårdgivare:");

         // *** Hämtar ut data ***
         // 1. Person id.
         System.out.println(" 1. Person id: " + resource.getResourceId());
         // 1. Namn.
         System.out.println(" 2. Namn: " + resource.getResourceName());

      }
   }

   /**
    * Entry point i Java applikationen.
    *
    * @param args kommando-prompt argument.
    */
   public static void main(String[] args)
   {
      try
      {
         UseCaseHamtaTjansteval exempelkod = new UseCaseHamtaTjansteval();
         exempelkod.useCaseHamtaVardval();
      } catch (Exception e)
      {
         e.printStackTrace();
      }
   }
}
```

**Hämtar tjänsteval/vårdval för angiven person.*

Följande är en exempelutskrift när ovan program körs när personen är listad på en vårdenhet

```
1. Sökandes personnummer: 121212-1212
2. Tjänsteutövaren (HSAID): SE90KC2-232A
 Vårdenhet:
 1. Namn: Björkhagens BVC
 2. Är det kö just nu: Ja
```

** Utskrift ifrån Java Console när programmet exekveras*

Följande är en exempelutskrift när ovan program körs när personen är listad på en läkare

```
1. Sökandes personnummer: 121212-1212
2. Tjänsteutövaren (HSAID): SE90KC2-232A
 Vårdenhet:
 1. Namn: Björkhagens BVC
 2. Är det kö just nu: Nej
 Vårdgivare:
 1. Person id: SE90KC2-2KU27
 2. Namn: Erik Tryggelin
```

** Utskrift ifrån Java Console när programmet exekveras*

4.3 Hämta tillgängliga tjänsteutövare – JAXWS-RI

För att se XML datat som finns i SOAP Body, se ref [11].

```
package com.mawell.nlt.consumer;

import java.net.MalformedURLException;
import java.net.URL;
import java.util.Iterator;
import java.util.List;

import javax.xml.namespace.QName;
import javax.xml.ws.Service;

import org.w3._2005._08.addressing.AttributedURIType;

import riv.crm.carelisting._1.Facility;
import riv.crm.carelisting.getavailablefacilities._1.rivtabp20.GetAvailableFacilitiesResponderInterface;
import riv.crm.carelisting.getavailablefacilities._1.rivtabp20.TechnicalException;
import riv.crm.carelisting.getavailablefacilitiesresponder._1.GetAvailableFacilitiesRequestType;
import riv.crm.carelisting.getavailablefacilitiesresponder._1.GetAvailableFacilitiesResponseType;


/**
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations
 * under the License.
 * Exempelkod för användningsfallet "Hämta tillgänglia tjänsteutövare".
 * Tjänsteutövare kan vara en vårdenhet.
 * @author Robert Siwerz, www.mawell.com.
 */
public class UseCaseHamtaTillgangligaTjansteutovare
{
        /**
         * Exempelkod för Use Case "Hämtar tillgängliga tjänsteutövare".
         *
         * @author Robert Siwerz,www.mawell.com.
         * @throws Fel vid kommunikation med tjänsten.
         */
        public void useCaseHamtaTillgangligaVardenheter() throws MalformedURLException
        {
                // Hämtar referens till SEI (Service Endpoint Interface).
                QName serviceName = new QName("urn:riv:crm:carelisting:GetAvailableFacilities:1:rivtabp20",
"GetAvailableFacilitiesResponderService");
                Service service = Service.create(new
URL("http://127.0.0.1:8088/mockGetAvailableFacilitiesResponderBinding?WSDL"),      serviceName);

                GetAvailableFacilitiesResponderInterface listingSEI =
service.getPort(GetAvailableFacilitiesResponderInterface.class);

                // Hämta tillgängliga tjänsteutövare.
                AttributedURIType logicalAddress = new AttributedURIType();
                logicalAddress.setValue("SE239482390-23SAD"); // HSAID till huvudmannen som skall svara på
frågan.
                GetAvailableFacilitiesRequestType request = new GetAvailableFacilitiesRequestType();
                GetAvailableFacilitiesResponseType response = null;

                try
                {
```

```
                    response = listingSEI.getAvailableFacilities(logicalAddress, request);
                } catch (TechnicalException e)
                {
                    // Gör ett nytt försök...
                }

                // Skriver att tillgängliga tjänsteutövare.
                List<Facility> facilities = response.getHealthcareFacilities();
                System.out.println("Tillgängliga vårdenheter:");
                Iterator<Facility> hsaIterator = facilities.iterator();

                while (hsaIterator.hasNext())
                {
                    Facility facility = hsaIterator.next();
                        System.out.println("HSAID: " + facility.getFacilityId() + ", namn: " +
facility.getName());
                }
        }

        /**
         * Entry point i Java applikationen.
         *
         * @param args kommando-prompt argument.
         */
        public static void main(String[] args)
        {
                try
                {
                        UseCaseHamtaTillgangligaTjansteutovare exempelkod = new
UseCaseHamtaTillgangligaTjansteutovare();
                        exempelkod.useCaseHamtaTillgangligaVardenheter();
                } catch (Exception e)
                {
                        e.printStackTrace();
                }
        }
}
```

**Hämtar tillgänglia vårdenheter.*

Exempel på programexekvering

```
Tillgängliga vårdenheter:
HSAID: SE234232-9A32S, namn: Vårdecentralen Kronan
HSAID: SE234232-2886A, namn: Skytteholms Vårdcentral
```

** Utskrift ifrån Java Console när programmet exekveras*

4.4 Hämta tillgängliga tjänsteutövare – Axis 1.4.1

För att se XML datat som finns i SOAP Body, se ref [11].

```
package com.mawell.nlt.consumer;

import java.net.MalformedURLException;

import org.apache.axis2.databinding.types.URI;


import rivtabp20._1.getavailablefacilities.carelisting.crm.riv.TechnicalException;
import rivtabp20._1.getavailablefacilities.carelisting.crm.riv.GetAvailableFacilitiesResponderServiceStub;
import rivtabp20._1.getavailablefacilities.carelisting.crm.riv.GetAvailableFacilitiesResponderServiceStub.AttributedUR
import rivtabp20._1.getavailablefacilities.carelisting.crm.riv.GetAvailableFacilitiesResponderServiceStub.CountyCode;
import rivtabp20._1.getavailablefacilities.carelisting.crm.riv.GetAvailableFacilitiesResponderServiceStub.Facility;
import rivtabp20._1.getavailablefacilities.carelisting.crm.riv.GetAvailableFacilitiesResponderServiceStub.GetAvailable
import
```

```
rivtabp20._1.getavailablefacilities.carelisting.crm.riv.GetAvailableFacilitiesResponderServiceStub.GetAvailableFacilit
import
rivtabp20._1.getavailablefacilities.carelisting.crm.riv.GetAvailableFacilitiesResponderServiceStub.GetAvailableFacilit
import rivtabp20._1.getavailablefacilities.carelisting.crm.riv.GetAvailableFacilitiesResponderServiceStub.To;

/**
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations
 * under the License.
 * Exempelkod för användningsfallet "Hämta tillgänglia tjänsteutövare".
 * Tjänsteutövare kan vara en vårdenhet.
 * @author Robert Siwerz, www.mawell.com.
 */
public class UseCaseHamtaTillgangligaTjansteutovareAxis
{
   /**
    * Exempelkod för Use Case "Hämtar tillgängliga tjänsteutövare".
    *
    * @author Robert Siwerz,www.mawell.com.
    * @throws Fel vid kommunikation med tjänsten.
    */
   public void useCaseHamtaTillgangligaVardenheter() throws MalformedURLException
   {
      GetAvailableFacilitiesResponse response = null;

      try
      {
         GetAvailableFacilitiesResponderServiceStub stub = new
GetAvailableFacilitiesResponderServiceStub("http://127.0.0.1:8088/mockGetAvailableFacilitiesResponderBinding?WSDL");

         GetAvailableFacilities getAvailableServiceProviders0 = new GetAvailableFacilities();
         GetAvailableFacilitiesRequestType data = new GetAvailableFacilitiesRequestType();
         CountyCode countyCode = new CountyCode();
         countyCode.setCountyCode("01");
         data.setCountyCode(countyCode);
         getAvailableServiceProviders0.setGetAvailableFacilities(data);

         To to1 = new To();
         AttributedURIType logicalAddress = new AttributedURIType();
         URI uri = new URI();
         uri.setPath("SE239482390-23SAD");
         logicalAddress.setAnyURI(uri); // HSAID till huvudmannen som skall svara på frågan.
         to1.setTo(logicalAddress);

         // Hämta tillgängliga tjänsteutövare.
         response = stub.getAvailableFacilities(getAvailableServiceProviders0, to1);
      } catch (TechnicalException e)
      {
         // Gör ett nytt försök...
      } catch (Exception e)
      {
         e.printStackTrace();
         return;
      }

      // Skriver att tillgängliga tjänsteutövare.
      Facility[] hsaIDs = response.getGetAvailableFacilitiesResponse().getHealthcare_facilities();
      System.out.println("Tillgängliga vårdenheter:");

      for (int i =0; i < hsaIDs.length; i++)
      {
```

```
            Facility serviceProvider = hsaIDs[i];
            System.out.println("HSAID: " + serviceProvider.getFacilityId() + ", " + serviceProvider.getName());
         }
      }

      /**
       * Entry point i Java applikationen.
       *
       * @param args kommando-prompt argument.
       */
      public static void main(String[] args)
      {
         try
         {
            UseCaseHamtaTillgangligaTjansteutovareAxis exempelkod = new UseCaseHamtaTillgangligaTjansteutovareAxis();
            exempelkod.useCaseHamtaTillgangligaVardenheter();
         } catch (Exception e)
         {
            e.printStackTrace();
         }
      }
}
```

**Hämtar tillgänglia vårdenheter.*

Exempel på programexekvering

```
Tillgängliga vårdenheter:
HSAID: SE234232-9A32S, namn: Vårdecentralen Kronan
HSAID: SE234232-2886A, namn: Skytteholms Vårdcentral
```

** Utskrift ifrån Java Console när programmet exekveras*

4.5 Göra tjänsteval – JAXWS-RI

För att se XML datat som finns i SOAP Body, se ref [13].

```
package com.mawell.nlt.consumer;

import java.net.MalformedURLException;
import java.net.URL;

import javax.xml.namespace.QName;
import javax.xml.ws.Service;

import org.w3._2005._08.addressing.AttributedURIType;

import riv.crm.carelisting.createlisting._1.rivtabp20.CreateListingResponderInterface;
import riv.crm.carelisting.createlisting._1.rivtabp20.InvalidFacilityException;
import riv.crm.carelisting.createlisting._1.rivtabp20.PersonNotFoundException;
import riv.crm.carelisting.createlisting._1.rivtabp20.TechnicalException;
import riv.crm.carelisting.createlistingresponder._1.CreateListingRequestType;
import riv.crm.carelisting.createlistingresponder._1.CreateListingResponseType;

/**
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
```

```
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations
 * under the License.
 * Exempelkod för användningsfallet "Göra Tjänsteval".
 *
 * @author Robert Siwerz,www.mawell.com.
 */
public class UseCasePerformAListing
{

   private void useCasePerformASpecificListing() throws MalformedURLException
   {
      // Hämtar referens till SEI (Service Endpoint Interface).
      QName serviceName = new QName("urn:riv:crm:carelisting:CreateListing:1:rivtabp20",
            "CreateListingResponderService");
      Service service = Service.create(new URL(
"http://127.0.0.1:8088/mockCreateListingResponderBinding?WSDL"),
            serviceName);

      CreateListingResponderInterface listingSEI =
service.getPort(CreateListingResponderInterface.class);

      try
      {
         AttributedURIType logicalAddress = new AttributedURIType();
         logicalAddress.setValue("01"); // Områdeskod

         // Skapar ett fråge objekt.
         CreateListingRequestType request = new CreateListingRequestType();
         request.setPersonId("195005055005");
         request.setHealthcareFacility("SE345345-ASD323");

         // Utför tjänstevalet.
         CreateListingResponseType response = listingSEI.createListing(logicalAddress, request);

         String status = (response.isSuccess() == true) ? "OK" : "FEL";
         System.out.println("Listnings status: " + status);
         System.out.println("Kommleterande information om listningsstatus: " + response.getComment());
         System.out.println("Kod från listningstjänsten: " + response.getSystemCode());
      } catch (PersonNotFoundException e)
      {
         System.out.println("Felkod:" + e.getFaultInfo().getCode());
      } catch (InvalidFacilityException e)
      {
         System.out.println("Felkod:" + e.getFaultInfo().getCode());
      } catch (TechnicalException e)
      {
         // Gör ett nytt försök...
      }
   }

   /**
    * Entry point i Java applikationen.
    *
    * @param args
    *           kommando-prompt argument.
    */
   public static void main(String[] args)
   {
      try
      {
         UseCasePerformAListing exempelkod = new UseCasePerformAListing();
         exempelkod.useCasePerformASpecificListing();
      } catch (Exception e)
      {
         e.printStackTrace();
      }
   }
```

```
}
```

** Göra ett vårdval för en specifik person och vårdenhet.*

Exempel på programexekvering

```
Listnings status: FEL
Kommleterande information om listningsstatus: Listningen
genomfördes inte pga. kö på vårdenheten
Kod från listningstjänsten: 1
```

** Utskrift ifrån Java Console när programmet exekveras*

4.6 Hämta listningstyper – JAXWS-RI

För att se XML datat som finns i SOAP Body, se ref [10].

```
package com.mawell.nlt.consumer;

import java.net.MalformedURLException;
import java.net.URL;
import java.util.List;

import javax.xml.namespace.QName;
import javax.xml.ws.Service;

import org.w3._2005._08.addressing.AttributedURIType;

import riv.crm.carelisting.getlistingtypes._1.rivtabp20.GetListingTypesResponderInterface;
import riv.crm.carelisting.getlistingtypes._1.rivtabp20.TechnicalException;
import riv.crm.carelisting.getlistingtypesresponder._1.GetListingTypesRequestType;
import riv.crm.carelisting.getlistingtypesresponder._1.GetListingTypesResponseType;

/**
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations
 * under the License.
 * Exempelkod för användningsfallet "Hämta tillgänglia listningstyper, t.ex {BVC, HLM}".
 * @author Robert Siwerz, www.mawell.com.
 */
public class UseCaseGetListingTypes
{

   public void useCaseGetListingTypes() throws MalformedURLException
   {
      // Hämtar referens till SEI (Service Endpoint Interface).
      QName serviceName = new QName("urn:riv:crm:carelisting:GetListingTypes:1:rivtabp20",
"GetListingTypesResponderService");
      Service service = Service.create(new
URL("http://127.0.0.1:8088/mockGetListingTypesResponderBinding?WSDL"),   serviceName);

      GetListingTypesResponderInterface listingSEI =
```

```
service.getPort(GetListingTypesResponderInterface.class);

      // Hämta tillgängliga tjänsteutövare.
      AttributedURIType logicalAddress = new AttributedURIType();
      logicalAddress.setValue("SE239482390-23SAD"); // HSAID till huvudmannen som skall svara på
frågan.
      GetListingTypesRequestType request = new GetListingTypesRequestType();
      GetListingTypesResponseType response = null;

      try
      {
         response = listingSEI.getListingTypes(logicalAddress, request);
      } catch (TechnicalException e)
      {
         // Gör ett nytt försök...
      }

      // Itererar över listan med listningstyper.
      System.out.println("Möjliga listningsval för personen");
      List<String> listingTypes = response.getListingType();

      for(String type : listingTypes)
      {
         System.out.println("Listningstyp: " + type);
      }
   }

   /**
    * Entry point i Java applikationen.
    *
    * @param args kommando-prompt argument.
    */
   public static void main(String[] args)
   {
      try
      {
         UseCaseGetListingTypes exempelkod = new UseCaseGetListingTypes();
         exempelkod.useCaseGetListingTypes();
      } catch (Exception e)
      {
         e.printStackTrace();
      }
   }
}
```

Exempel på programexekvering

```
Möjliga listningsval för personen
Listningstyp: BVC
Listningstyp: HLM
```

** Utskrift ifrån Java Console när programmet exekveras*

4.7 Hämta köinformation för en person– JAXWS-RI

```
package com.mawell.nlt.consumer;

import java.net.MalformedURLException;
import java.net.URL;

import javax.xml.namespace.QName;
import javax.xml.ws.Service;

import org.w3._2005._08.addressing.AttributedURIType;

import riv.crm.carelisting._1.PersonQueueStatus;
import
```

```
riv.crm.carelisting.getpersonqueuestatus._1.rivtabp20.GetPersonQueueStatusResponderInterface;
import riv.crm.carelisting.getpersonqueuestatus._1.rivtabp20.PersonNotFoundException;
import riv.crm.carelisting.getpersonqueuestatus._1.rivtabp20.TechnicalException;
import riv.crm.carelisting.getpersonqueuestatusresponder._1.GetPersonQueueStatusRequestType;
import riv.crm.carelisting.getpersonqueuestatusresponder._1.GetPersonQueueStatusResponseType;

/**
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations
 * under the License.
 * Exempelkod för användningsfallet "Hämta köinformation för en person".
 * @author Daniel Berggren, www.mawell.com.
 */
public class UseCaseGetPersonQueueStatus
{

   public void useCaseGetPersonQueueStatus() throws MalformedURLException
   {
      // Hämtar referens till SEI (Service Endpoint Interface).
      QName serviceName = new
QName("urn:riv:crm:carelisting:GetPersonQueueStatus:1:rivtabp20",
"GetPersonQueueStatusResponderService");
      Service service = Service.create(new
URL("http://localhost:8088/mockGetPersonQueueStatusResponderBinding?WSDL"),   serviceName);

      GetPersonQueueStatusResponderInterface listingSEI =
service.getPort(GetPersonQueueStatusResponderInterface.class);

      // Hämta köstsatus för person
      AttributedURIType logicalAddress = new AttributedURIType();
      logicalAddress.setValue("01"); // Områdeskod

      GetPersonQueueStatusRequestType request = new GetPersonQueueStatusRequestType();
      request.setPersonId("1212121212-1212");

      GetPersonQueueStatusResponseType response = null;

      try
      {
         response = listingSEI.getPersonQueueStatus(logicalAddress, request);
      }
      catch(PersonNotFoundException e)
      {

      }
      catch(TechnicalException e)
      {

      }

      // Itererar över listan med listningstyper.
      System.out.println("Köstatus för personen ( " + request.getPersonId() + ") är");

      if(response.getQueueStatus()==PersonQueueStatus.IN_QUEUE)
      {
                System.out.println(" står i kö");
                System.out.println("  ---------------");
                System.out.println("  På vårdenheten:");
                System.out.println("    HsaId: " +
response.getHealthcareFacility().getFacilityId());
                System.out.println("    köstatus för vårdenheten: " +
```

```
response.getHealthcareFacility().isHasQueue());
                System.out.println("    lisningstyper:");
                for (String
listingType:response.getHealthcareFacility().getSupportedListingTypes())
                {
                        System.out.println("    " + listingType);
                }
        }
        else
            System.out.println(" står inte i kö");
    }

    /**
     * Entry point i Java applikationen.
     *
     * @param args kommando-prompt argument.
     */
    public static void main(String[] args)
    {
        try
        {
            UseCaseGetPersonQueueStatus exempelkod = new UseCaseGetPersonQueueStatus();
            exempelkod.useCaseGetPersonQueueStatus();
        } catch (Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

Exempel på programexekvering

```
Köstatus för personen ( 1212121212-1212) är
 står i kö
  ---------------
  På vårdenheten:
    HsaId: SE2342322-886A
    köstatus för vårdenheten: true
    lisningstyper:
    - BVC
    - HLM
```

** Utskrift ifrån Java Console när programmet exekveras*

5. Testning

För att säkerställa att Anslutningspunkten/Konsument fungerar som den skall så behöver tester göras som simulerar användningsfallen. Testerna kan skrivas för hand eller alternativt kan testverktyg används som simulerar användningsfallen. Ett sådant testverktyg är soapUI (se ref[4]).

6. Referenser

[1] - RIV Tekniska anvisningar (Regelverk för Interoperabilitet inom Vård och omsorg): http://rivta.forge.osor.eu/

[2] EN13606-1: http://www.chime.ucl.ac.uk/resources/CEN/EN13606-1/

[3] – wsimport: http://java.sun.com/javase/6/docs/technotes/tools/share/wsimport.html

[4] – soapUI: http://www.soapui.org/

[5] VIT Boken: http://arkitekturledningen.se/undermappar/links/VITstart.htm

[6] Meddelandestruktur, Meddelandestruktur_Visa_vårdval_Nationell_Listningstjänst_EN13606_version_0.1.xlsx

[7] Apache CFX: http://cxf.apache.org/

[8] Metro: https://metro.dev.java.net/

[9] Hämta vårdval: SOAP exempel_Visa_tjänsteval.xml

[10] Göra vårdval: SOAP exempel_Göra_tjänsteval.xml

[11] Visa tillgängliga vårdenheter: SOAP exempel_Hämta_möjliga_tjänsteutövare.xml

[12] JSR 181 Web Services Metadata: http://jcp.org/en/jsr/detail?id=181

[13] Visa listningstyper: SOAP exempel_Hämta_listningstyper.xml