

# 1 TD2 - Services web SOAP

## 1.1 Listings

### 1.1.1 Listing 1 - getDate1.wsdl

```
<!--../source/getDate1.wsdl-->

?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://mondomaine.fr/monAppli"
  xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  targetNamespace="http://mondomaine.fr/monAppli"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema
      targetNamespace="http://mondomaine.fr/monAppli"
      elementFormDefault="qualified">
      <s:element name="getDate1">
        <s:complexType/>
      </s:element>
      <s:element name="getDate1Response">
        <s:complexType>
          <s:sequence>
            <s:element
              name="getDate1Result"
              type="s:string"
              minOccurs="0" maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="getDate1SoapIn">
    <wsdl:part name="parameters" element="tns:getDate1"/>
  </wsdl:message>
  <wsdl:message name="getDate1SoapOut">
    <wsdl:part name="parameters" element="tns:getDate1Response"/>
  </wsdl:message>
  <wsdl:portType name="ServiceDateSoap">
    <wsdl:operation name="getDate1">
      <wsdl:input message="tns:getDate1SoapIn"/>
      <wsdl:output message="tns:getDate1SoapOut"/>
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>
```

```

        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="ServiceDateSoap" type="tns:ServiceDateSoap">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="getDate1">
            <soap:operation
                soapAction="http://mondomaine.fr/monAppli/getDate1"
                style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:binding name="ServiceDateSoap12" type="tns:ServiceDateSoap">
        <soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="getDate1">
            <soap12:operation
                soapAction="http://mondomaine.fr/monAppli/getDate1"
                style="document"/>
            <wsdl:input>
                <soap12:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap12:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="ServiceDate">
        <wsdl:port name="ServiceDateSoap" binding="tns:ServiceDateSoap">
            <soap:address location="http://localhost/ws.asmx"/>
        </wsdl:port>
        <wsdl:port name="ServiceDateSoap12" binding="tns:ServiceDateSoap12">
            <soap12:address location="http://localhost/ws.asmx"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>

```

### 1.1.2 Listing 2 - SOAP of getDate1.wsdl

```
<!--../source/getDate1SOAP.xml-->
```

POST /ws.asmx HTTP/1.1

Host: localhost  
Content-Type: application/soap+xml; charset=utf-8  
Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getDate1 xmlns="http://mondomaine.fr/monAppli" />
  </soap12:Body>
</soap12:Envelope>
```

HTTP/1.1 200 OK  
Content-Type: application/soap+xml; charset=utf-8  
Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getDate1Response xmlns="http://mondomaine.fr/monAppli">
      <getDate1Result>string</getDate1Result>
    </getDate1Response>
  </soap12:Body>
</soap12:Envelope>
```

### 1.1.3 Listing 3 - getDate2.wsdl

```
<!--../source/getDate2.wsdl-->

?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns="http://mondomaine.fr/monAppli"
  xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  targetNamespace="http://mondomaine.fr/monAppli"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema
```

```

targetNamespace="http://mondomaine.fr/monAppli"
elementFormDefault="qualified" >
<s:element name="getDate2">
  <s:complexType>
    <s:sequence>
      <s:element
        name="f"
        type="tns:Format"
        minOccurs="1" maxOccurs="1" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:simpleType name="Format">
  <s:restriction base="s:string">
    <s:enumeration value="JMA"/>
    <s:enumeration value="HMS"/>
    <s:enumeration value="HMSJMA"/>
  </s:restriction>
</s:simpleType>
<s:element name="getDate2Response">
  <s:complexType>
    <s:sequence>
      <s:element
        name="getDate2Result"
        type="s:string"
        minOccurs="0" maxOccurs="1" />
    </s:sequence>
  </s:complexType>
</s:element>
</s:schema>
</wsdl:types>
<wsdl:message name="getDate2SoapIn">
  <wsdl:part name="parameters" element="tns:getDate2"/>
</wsdl:message>
<wsdl:message name="getDate2SoapOut">
  <wsdl:part name="parameters" element="tns:getDate2Response"/>
</wsdl:message>
<wsdl:portType name="ServiceDateSoap">
  <wsdl:operation name="getDate2">
    <wsdl:input message="tns:getDate2SoapIn"/>
    <wsdl:output message="tns:getDate2SoapOut"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ServiceDateSoap" type="tns:ServiceDateSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="getDate2">

```

```

        <soap:operation
            soapAction="http://mondomaine.fr/monAppli/getDate2"
            style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="ServiceDateSoap12" type="tns:ServiceDateSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="getDate2">
        <soap12:operation
            soapAction="http://mondomaine.fr/monAppli/getDate2"
            style="document"/>
        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="ServiceDate">
    <wsdl:port name="ServiceDateSoap" binding="tns:ServiceDateSoap">
        <soap:address location="http://localhost/ws.asmx"/>
    </wsdl:port>
    <wsdl:port name="ServiceDateSoap12" binding="tns:ServiceDateSoap12">
        <soap12:address location="http://localhost/ws.asmx"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

#### 1.1.4 Listing 4 - SOAP of getDate2.wsdl

```
<!--../source/getDate2SOAP.xml-->
```

```

POST /ws.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

```

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<soap12:Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getDate2 xmlns="http://mondomaine.fr/monAppli">
      <f>JMA or HMS or HMSJMA</f>
    </getDate2>
  </soap12:Body>
</soap12:Envelope>

```

-----

HTTP/1.1 200 OK  
Content-Type: application/soap+xml; charset=utf-8  
Content-Length: length

```

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getDate2Response xmlns="http://mondomaine.fr/monAppli">
      <getDate2Result>string</getDate2Result>
    </getDate2Response>
  </soap12:Body>
</soap12:Envelope>

```

### 1.1.5 Listing 5 - getDate3.wsdl

```

<!--../source/getDate3.wsdl-->

?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns="http://mondomaine.fr/monAppli"
  xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  targetNamespace="http://mondomaine.fr/monAppli"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">

```

```

<wsdl:types>
  <s:schema
    targetNamespace="http://mondomaine.fr/monAppli"
    elementFormDefault="qualified" >
    <s:element name="getDate3">
      <s:complexType/>
    </s:element>
    <s:element name="getDate3Response">
      <s:complexType>
        <s:sequence>
          <s:element name="getDate3Result"
            type="tns:Date"
            minOccurs="0" maxOccurs="1" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:complexType name="Date">
      <s:sequence>
        <s:element
          name="jour"
          type="s:int"
          minOccurs="1" maxOccurs="1" />
        <s:element
          name="mois"
          type="s:int"
          minOccurs="1" maxOccurs="1" />
        <s:element
          name="annee" type="s:int"
          minOccurs="1" maxOccurs="1" />
      </s:sequence>
    </s:complexType>
  </s:schema>
</wsdl:types>
<wsdl:message name="getDate3SoapIn">
  <wsdl:part name="parameters" element="tns:getDate3"/>
</wsdl:message>
<wsdl:message name="getDate3SoapOut">
  <wsdl:part name="parameters" element="tns:getDate3Response"/>
</wsdl:message>
<wsdl:portType name="ServiceDateSoap">
  <wsdl:operation name="getDate3">
    <wsdl:input message="tns:getDate3SoapIn"/>
    <wsdl:output message="tns:getDate3SoapOut"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ServiceDateSoap" type="tns:ServiceDateSoap">

```

```

<soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="getDate3">
  <soap:operation
    soapAction="http://mondomaine.fr/monAppli/getDate3"
    style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="ServiceDateSoap12" type="tns:ServiceDateSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="getDate3">
    <soap12:operation
      soapAction="http://mondomaine.fr/monAppli/getDate3"
      style="document"/>
    <wsdl:input>
      <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="ServiceDate">
  <wsdl:port name="ServiceDateSoap" binding="tns:ServiceDateSoap">
    <soap:address location="http://localhost/ws.asmx"/>
  </wsdl:port>
  <wsdl:port name="ServiceDateSoap12" binding="tns:ServiceDateSoap12">
    <soap12:address location="http://localhost/ws.asmx"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

#### 1.1.6 Listing 6 - SOAP of getDate3.wsdl

```
<!--../source/getDate3SOAP.xml-->
```

```

POST /ws.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length

```



SOAPAction: "http://mondomaine.fr/monAppli/getDate3"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getDate3 xmlns="http://mondomaine.fr/monAppli" />
  </soap:Body>
</soap:Envelope>
```

-----

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getDate3Response xmlns="http://mondomaine.fr/monAppli">
      <getDate3Result>
        <jour>int</jour>
        <mois>int</mois>
        <annee>int</annee>
      </getDate3Result>
    </getDate3Response>
  </soap:Body>
</soap:Envelope>
```

## 1.2 Web Service Description Language

### 1.2.1 Question 1

Étudiez ligne à ligne le fichier `getDate1.wsdl` afin de comprendre :

- dans un premier temps, sans se préoccuper des espaces de noms et autres problèmes xml, quel service est déclaré dans ce fichier
- ensuite les détails xml nécessaires
- enfin le fonctionnement de la partie concrète (avec les bindings)

### 1.2.1.1 Réponse

En général, le service déclaré dans un document WSDL est soit indiqué comme valeur de l'attribut `name` de l'élément `<definitions>` ou l'élément `<service>`. Dans `getDate1.wsdl`, le nom du service est **ServiceDate** (*indiqué dans `<service>`*).

L'élément `<definitions>` introduit les espaces de noms utilisés dans le document, incluant :

1. `targetNamespace="http://mondomaine.fr/monAppli"` : l'espace de noms de `getDate1.wsdl`.
2. `xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/"` : l'espace de noms de la spécification WSDL (*n.b., on aurait dû le déclarer en tant qu'espace de noms par défaut pour ne pas préfixer les éléments du document par `wSDL`*).
3. `xmlns:soap="http://schemas.xmlsoap.org/wSDL/soap/"` : l'espace de noms de la spécification SOAP 1.1.
4. `xmlns:soap12="http://schemas.xmlsoap.org/wSDL/soap12/"` : l'espace de noms de la spécification SOAP 1.2.
5. `xmlns:tns="http://mondomaine.fr/monAppli"` : l'espace de noms des éléments et types déclarés dans `getDate1.wsdl`.
6. `xmlns:s="http://www.w3.org/2001/XMLSchema"` : l'espace de noms de la spécification des schémas XML.

L'élément `<types>` introduit les éléments déclarés par `getDate1.wsdl`, incluant un élément nommé `getDate1` encapsulant un type complexe vide et un élément `getDate1Response` encapsulant un type complexe désignant un élément résultat optionnel `getDate1Result` de type `string`.

Le document déclare deux `<message>` nommés `getDate1SoapIn` et `getDate1SoapOut` référençant les éléments déclarés `getDate1` et `getDate1Response` respectivement, et constituant les paramètres d'entrée et de sortie (*i.e., `<input>` et `<output>`*) de l'`<operation>` nommée `getDate1`, regroupée au sein du `<portType>` nommé `ServiceDateSoap`.

Ensuite le document déclare deux `<binding>` nommés `ServiceDateSoap` et `ServiceDateSoap12` utilisant le protocole de transport HTTP avec SOAP 1.1 et 1.2 respectivement, en mode `document` et avec encodage selon le schéma XML défini dans `<type>` (*i.e., `usage literal`*).

Enfin le document déclare un `<service>` encapsulant deux **endpoints** (*i.e., `<port>`*) liant la même opération `getDate1` à l'adresse réseau `http://localhost/ws.asmx` pour les deux versions de SOAP.

En Java, l'opération du service web aura la signature suivante :

```
public String getDate1();
```

## 1.2.2 Question 2

Quel service est publié dans le fichier `getDate2.wsdl` ?

### 1.2.2.1 Réponse

Dans `getDate2.wsdl`, le nom du service est **ServiceDate** (*indiqué dans `<service>`*).

L'élément `<definitions>` introduit les mêmes espaces de noms utilisés dans `getDate1.wsdl`, et introduit en plus :

- `xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"` : l'espace de noms de la spécification sur les règles d'encodage de SOAP1.1.

L'élément `<types>` introduit les éléments déclarés par `getDate2.wsdl`. En premier, un élément nommé `getDate2` est déclaré, qui encapsule un type complexe composé d'un élément nommé `f` de type `tns:Format`. En deuxième, le type `tns:Format` est défini comme type simple ayant une valeur parmi **JMA (Date)**, **HMS (Temps)**, et **HMSJMA (Date et Temps)**. Enfin, le document définit un élément `getDate2Response` encapsulant un type complexe désignant un élément résultat optionnel `getDate2Result` de type `string`.

Le document déclare deux `<message>` nommés `getDate2SoapIn` et `getDate2SoapOut` référant les éléments déclarés `getDate2` et `getDate2Response` respectivement, et constituant les paramètres d'entrée et de sortie de l'`<operation>` nommée `getDate2`, regroupée au sein du `<portType>` nommé `ServiceDateSoap`.

Ensuite le document déclare deux `<binding>` nommés `ServiceDateSoap` et `ServiceDateSoap12` utilisant le protocole de transport HTTP avec SOAP 1.1 et 1.2 respectivement, en mode `document` et avec encodage selon le schéma XML défini dans `<type>` (*i.e., `usage literal`*).

Enfin le document déclare un `<service>` encapsulant deux **endpoints** (*i.e., `<port>`*) liant la même opération `getDate2` à l'adresse réseau `http://localhost/ws.asmx` pour les deux versions de SOAP.

En C#, l'opération du service web aura la signature suivante :

```
public String getDate2(Format f);
```

Alors que `Format` aura la définition suivante :

```
public enum Format
{
    JMA,
    HMS,
    HMSJMA
}
```

### 1.2.3 Question 3

Quel service est publié dans le fichier `getDate3.wsdl` ?

#### 1.2.3.1 Réponse

Dans `getDate3.wsdl`, le nom du service est **ServiceDate** (*indiqué dans `<service>`*).

L'élément `<definitions>` introduit les mêmes espaces de noms utilisés dans `getDate2.wsdl`, et introduit en plus :

1. `xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"` : l'espace de noms pour l'utilisation de HTTP avec WSDL.
2. `xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"` : l'espace de noms de la spécification du protocole **MIME (Multipurpose Internet Mail Extensions)**.
3. `xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"` : l'espace de noms pour l'utilisation des éléments liés au text matching avec les types MIME.

L'élément `<types>` introduit les éléments déclarés par `getDate3.wsdl`. En premier, un élément nommé `getDate3` est déclaré, qui encapsule un type complexe vide. En deuxième, le type `getDate3Response` est défini comme type complexe désignant un élément résultat optionnel `getDate3Result` de type `tns:Date`. Enfin, le type `tns:Date` est défini comme type complexe composé de trois sous-éléments de type `int`, nommés `jour`, `mois`, et `annee` respectivement.

Le document déclare deux `<message>` nommés `getDate3SoapIn` et `getDate3SoapOut` référant les éléments déclarés `getDate3` et `getDate3Response` respectivement, et constituant les paramètres d'entrée et de sortie de l'`<operation>` nommée `getDate3`, regroupée au sein du `<portType>` nommé `ServiceDateSoap`.

Ensuite le document déclare deux `<binding>` nommés `ServiceDateSoap` et `ServiceDateSoap12` utilisant le protocole de transport HTTP avec SOAP 1.1 et 1.2 respectivement, en mode `document` et avec encodage selon le schéma XML défini dans `<type>` (*i.e., usage literal*).

Enfin le document déclare un `<service>` encapsulant deux **endpoints** (*i.e., `<port>`*) liant la même opération `getDate3` à l'adresse réseau `http://localhost/ws.asmx` pour les deux versions de SOAP.

En C#, l'opération du service web aura la signature suivante :

```
public Date getDate3();
```

Alors que `Date` aura la définition suivante :

```

public class Date
{
    int jour;
    int mois;
    int annee;

    public Date()
    {

    }

    public Date(int jour, int mois, int annee)
    {
        this.jour = jour;
        this.mois = mois;
        this.annee = annee;
    }
}

```

#### 1.2.4 Question 4

Avec la connaissance du fichier `.cs` qui a permis la génération des wsdl, réfléchissez à la bonne façon d'écrire vos méthodes de service de manière à ce que les services soient facilement utilisables.

##### 1.2.4.1 Réponse

```

// ../source/ServiceDate.cs

using System;
using System.Collections.Generic;
using System.Text;
using System.Web.Services;

namespace TDWS
{
    [WebService(Namespace = "http://mondomaine.fr/monAppli")]
    class ServiceDate
    {
        [WebMethod]
        public String getDate1()
        {
            return getDate2(Format.HMSJMA);
        }
    }
}

```

```

[WebMethod]
public String getDate2(Format f)
{
    DateTime now = DateTime.Now;
    String result = "";
    switch(f)
    {
        case JMA:
            result = now.ToString("dd/MM/yyyy");
            break;

        case HMS:
            result = now.ToString("hh:mm:ss");
            break;

        case HMSJMA:
            result = now.ToString("hh:mm:ss dd/MM/yyyy");
            break;
    }

    return result;
}

[WebMethod]
public Date getDate3()
{
    DateTime now = DateTime.Now;
    Date today = new Date(now.Day, now.Month, now.Year);
    return today;
}

}

public class Date
{
    public int jour;
    public int mois;
    public int annee;
    public Date()
    {
    }

    public Date(int j,int m,int a)
    {
        jour=j;
        mois=m;
    }
}

```

```

        annee=a;
    }
}

public enum Format
{
    JMA, // jour mois annee
    HMS, // Heure, minutes, secondes
    HMSJMA
}
}

```

## 1.3 SOAP

### 1.3.1 Question 1

Étudiez les requêtes SOAP correspondant à un appel au service web `getDate1` défini dans la section précédente de manière à comprendre l'encodage SOAP en fonction du fichier WSDL. Cet encodage dépend de ce qui est spécifié dans les bindings. Vous trouverez des détails à ce sujet à cette adresse : <https://developer.ibm.com/articles/ws-whichwsdl/>

#### 1.3.1.1 Réponse

Quand le client invoque le service `public String getDate1()` associé à SOAP 1.2, le message SOAP envoyé est transporté par HTTP et utilise le mode **Document** avec un encodage **literal**. Le paramètre d'entrée vide et la valeur de retour sont encodés tels qu'ils sont définis au sein de leur schéma de définition XML. De plus, la valeur de retour `<getDate1Result>` est encapsulée par un élément **wrapper** `<getDate1Response>` constituant le seul élément enfant de `<SOAP-ENV:Body>`. Enfin, on remarque que le résultat retourné désigne bien un string, mais sa valeur ne désigne pas une date.

### 1.3.2 Question 2

Mêmes questions pour `getDate2` et `getDate3`.

#### 1.3.2.1 Réponse

Les services `public String getDate1()`, `public String getDate2(Format f)`, et `public Date getDate3()` partagent tous le même protocole de transport (HTTP), le même protocole d'échange de messages (SOAP), et le même mode de représentation et d'encodage des données, i.e., **Document** et **literal**

respectivement. Toutefois, `getDate1` et `getDate2` utilisent SOAP 1.2 alors que `getDate3` utilise SOAP 1.1.

#### 1.3.2.1.1 `getDate2`

Le paramètre d'entrée et la valeur de retour sont encodés tels qu'ils sont définis au sein de leur schéma de définition XML. Le paramètre d'entrée `f` est un élément de type `Format` correspondant à `public enum Format {JMA, HMS, HMSJMA}` et la valeur de retour `<getDate2Result>` est encapsulée par un élément **wrapper** `<getDate2Response>` constituant le seul élément enfant de `<SOAP-ENV:Body>`. De plus, on remarque que la valeur renseignée pour le paramètre d'entrée désigne bien un string mais ne désigne pas une valeur valide parmi JMA, HMS, et HMSJMA, et qu'il s'agit simplement d'un **string placeholder**. Enfin, on remarque que le résultat retourné désigne bien un string, mais sa valeur ne désigne pas une date.

#### 1.3.2.1.2 `getDate3`

Le paramètre d'entrée et la valeur de retour sont encodés tels qu'ils sont définis au sein de leur schéma de définition XML. Le paramètre d'entrée est vide, alors que la valeur de retour `<getDate3Result>` est un élément de type `Date` correspondant à `public class Date { int jour; int mois; int annee; /* constructors */ }`. De plus, la valeur de retour est encapsulée par un élément **wrapper** `<getDate3Response>` constituant le seul élément enfant de `<SOAP-ENV:Body>`. Enfin, on remarque que la valeur de retour ne désigne pas une date et que les valeurs renseignées pour les éléments enfants de `<getDate2Result>` désignent des strings, et non pas des entiers. Il s'agit ainsi simplement d'un **string placeholder**.