

Corrigé-Type - EMD du module : Systèmes d'informations – Bases de données avancées
2ème Année Master GL – (S3)

Question1 : (02 Pts)

REST Permet l'envoi de messages sans enveloppe SOAP et dans un encodage libre (XML, JSON, binaire, simple texte).

	REST	SOAP
Standardisé	Non	Oui
Sécurité	Non (restreinte par l'emploi des méthodes HTTP)	Oui (WS-Security)
Complexité	Non	Oui (lourdeur)

Question2 : (03 Pts = 01 pt pour la définition de UDDI et 0.5 Pt pour chaque structure de l'UDDI avec un moindre détail)

- Informations métier (businessEntity).** La structure businessEntity regroupe des informations sur l'entreprise ayant publié le service, tel que le nom, la description, les contacts et les identificateurs de cette entreprise.
- Informations sur le service (businessService).** Informations décrivant un groupe de services Web. Cette structure contient des informations sur les familles de services techniques. Elle regroupe un ensemble de services Web associés à un processus métier ou à un groupe de services
- Informations de liaison (bindingTemplate).** Cette structure détient des informations techniques servant à déterminer le point d'entrée et les spécifications de construction pour l'appel d'un service Web. La structure bindingTemplate fournit les descriptions de service Web utiles aux développeurs d'applications souhaitant rechercher et appeler un service Web. La structure bindingTemplate pointe vers des descriptions d'implémentation d'un service, par le biais d'une adresse URL, par exemple.
- Informations décrivant les spécifications les services (tModel).** Cette structure représente les métadonnées afférentes aux différentes spécifications implémentées par un service Web donné. tModel fournit un système de références (index) facilitant la reconnaissance des services Web.

Exercice 01 :

Soit le document XML :

- `<?xml version="1.0" encoding="ISO-8859-1" standalone="standalone"?>` **yes ou no**
- `<gdt month=February year=2004 <!-- à partir d'ici, on peut utiliser XML --> >` **commentaire non autorisé entre < et >**
- Les attributs doivent être délimités par des "" ou des ''**
- `<expose day="5">`
- `<speaker>"Nicolas Baudru"</speaker>`
- `<title>Netcharts et HMSC <!-- à partir d'ici, on ne peut pas utiliser XML --></Title>` **Balise d'ouverture différente à la balise de fermeture**
- `<time>12h45</time>`
- `<salle>102</salle>`
- `</expose>`
- `<expose day="12">`

10. <comment>Exposé proposé par François Denis, équipe BDA </comment>
11. <speaker>Liva Ralaivola **manque balise de fermeture </speaker>**
12. <title>apprentissage statistique, noyaux et applications à la bio-informatique</title>
13. <time>14h</time>
14. <salle>104</salle>
15. </expose>
16. <expose day="19">
17. <comment>pas de groupe de travail</comment>
18. </expose>
19. <expose day="26">
20. <speaker>Pedro d'Argenio</speaker>
21. <title>Secure Information Flow by Self-Composition</title>
22. <time>12h45</time>
23. <salle>102</salle>
24. </expose>
25. </gdt>

Maintenant, on considère la grammaire xsd :

```
<?xml version="1.0" encoding="utf-16"?>
<xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="gdt">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="expose" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="comment" type="xs:string" minOccurs="0" />
              <xs:element name="speaker" type="xs:string" minOccurs="0" />
              <xs:element name="title" type="xs:string" minOccurs="0" />
              <xs:element name="time" type="xs:string" minOccurs="0" />
              <xs:element name="salle" type="xs:string" minOccurs="0" />
            </xs:sequence>
            <xs:attribute name="day" type="xs:string" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="month" type="xs:string" />
      <xs:attribute name="year" type="xs:string" />
    </xs:complexType>
  </xs:element>
  <xs:element name="NewDataSet">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="gdt" />
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Questions :

1. Le document n'est pas bien formé (les erreurs sont colorées en rouge) **(03 Pts)**
2. Est-ce qu'il est valide ? non **(03 Pts)**

Voilà la grammaire corrigée :

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
```

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="gdt">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="expose" minOccurs="0" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="comment" type="xsd:string" minOccurs="0" />
              <xsd:element name="speaker" type="xsd:string" minOccurs="0" />
              <xsd:element name="title" type="xsd:string" minOccurs="0" />
              <xsd:element name="time" type="xsd:string" minOccurs="0" />
              <xsd:element name="salle" type="xsd:string" minOccurs="0" />
            </xsd:sequence>
            <xsd:attribute name="day" type="xsd:string" />
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="month" type="xsd:string" />
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="NewDataSet">
    <xsd:complexType>
      <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element ref="gdt" />
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```

Exercice 02 :

Un serveur tourne et vous propose un service de traduction pour lequel on vous fournit le fichier WSDL

1. La version du document 1.1, la présence de PortType à la place de « Interface » **(1 Pt)**
2. Où se trouve ce service ? <http://iic10:8080/axis/Traducteur.jws> **(1 Pt)**
3. Donnez la signature des opérations décrites dans le document **(0.5 Pt) pour chaque opération**

Opération	Input	Type de données	Output	Type de données
version	motATraduire	string	versionReturn	string
theme	motATraduire	string	themeReturn	string
nbMotDico	/	/	nbMotDicoReturn	int

4. Donnez un exemple d'URL pour déclencher une opération depuis un navigateur web **(1 Pt)**
<http://iic10:8080/axis/Traducteur.jws/method=version?motATraduire=Go>

5. Protocole supporté ? **SOAP (0.5Pt)**

```

<wsdl:definitions xmlns:apachesoap="http://xml.apache.org/xml-
soap" xmlns:impl="http://iic10:8080/axis/Traducteur.jws" xmlns:intf="http://iic10:8080/axis/Traducteur.jws" xmlns:s
oapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:wsdl
soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespac
e="http://iic10:8080/axis/Traducteur.jws">
  <wsdl:message name="nbMotDicoRequest"></wsdl:message>
  <wsdl:message name="versionRequest">
    <wsdl:part name="motATraduire" type="xsd:string"/>
  </wsdl:message>

```

```

</wsdl:message>
<wsdl:message name="versionResponse">
<wsdl:part name="versionReturn" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="nbMotDicoResponse">
<wsdl:part name="nbMotDicoReturn" type="xsd:int"/>
</wsdl:message>
<wsdl:message name="themeResponse">
<wsdl:part name="themeReturn" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="themeRequest">
<wsdl:part name="motATraduire" type="xsd:string"/>
</wsdl:message>
<wsdl:portType name="Traducteur">
<wsdl:operation name="version" parameterOrder="motATraduire">
<wsdl:input message="impl:versionRequest" name="versionRequest"/>
<wsdl:output message="impl:versionResponse" name="versionResponse"/>
</wsdl:operation>
<wsdl:operation name="theme" parameterOrder="motATraduire">
<wsdl:input message="impl:themeRequest" name="themeRequest"/>
<wsdl:output message="impl:themeResponse" name="themeResponse"/>
</wsdl:operation>
<wsdl:operation name="nbMotDico">
<wsdl:input message="impl:nbMotDicoRequest" name="nbMotDicoRequest"/>
<wsdl:output message="impl:nbMotDicoResponse" name="nbMotDicoResponse"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="TraducteurSoapBinding" type="impl:Traducteur">
<wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="version">
<wsdlsoap:operation soapAction=""/>
<wsdl:input name="versionRequest">
<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://DefaultNamespa
ce" use="encoded"/>
</wsdl:input>
<wsdl:output name="versionResponse">
<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://iic10:8080/axis/T
raducteur.jws" use="encoded"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="theme">
<wsdlsoap:operation soapAction=""/>
<wsdl:input name="themeRequest">
<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://DefaultNamespa
ce" use="encoded"/>
</wsdl:input>
<wsdl:output name="themeResponse">
<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://iic10:8080/axis/T
raducteur.jws" use="encoded"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="nbMotDico">
<wsdlsoap:operation soapAction=""/>
<wsdl:input name="nbMotDicoRequest">
<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://DefaultNamespa
ce" use="encoded"/>

```

```

</wsdl:input>
<wsdl:output name="nbMotDicoResponse">
<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://iic10:8080/axis/Traducteur.jws" use="encoded"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="TraducteurService">
<wsdl:port binding="impl:TraducteurSoapBinding" name="Traducteur">
<wsdlsoap:address location="http://iic10:8080/axis/Traducteur.jws"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Exercice 03 : Décrivez le code XML suivant : (03 Pts)

```

<SOAP-ENV:Envelope SOAP-ENV:encodingStyle=
"http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<SOAP-ENV:Body>
<ns1:getTime xmlns:ns1="urn:MySoapServices">
< fuseauHoraire xsi:type="xsd:string">GMT</fuseauHoraire>
</ns1:getTime>
</SOAP-ENV:Body> </SOAP-ENV:Envelope>

```

Ce code XML décrit une requête SOAP, invocation de la méthode getTime avec un paramètre fuseauHoraire de type chaîne de caractères dont la valeur est GMT.