

Cours n°6 : Web services

SERVICES RÉSEAUX

Hélène CHASSAGNE
helene.chassagne@orange.fr


Frédéric CHASSAGNE
frederic.chassagne@atosorigin.com



Plan du cours

- ♦ Web services
 - Présentation
 - Architecture
 - Web service en .net


2



Web services

- ♦ Web service = Application avec laquelle on communique par l'échange de message
- ♦ Architecture
 - Service Oriented Architecture
 - Frontières explicites
 - Coût des transactions
 - Entité autonome
 - Déploiement et évolution d'un service indépendant des clients
 - Structure d'utilisation définie sans ambiguïté
 - Sémantique d'utilisation définie sans ambiguïté

3



Web services

- ♦ Service web
 - Composant développé dans n'importe quel langage
 - Composant développé sur n'importe quelle plateforme
 - Composant enveloppé dans une couche XML
- ♦ Obligation
 - Pouvoir être invoqué par n'importe quel autre service

4

Web services

- ♦ Description des Web services
 - Web service = 3 protocoles
 - SOAP (Simple Object Access Protocol) = protocole permettant l'échange de données quelque soit la plateforme (flux de données XML sur HTTP).
 - WSDL (Web Service Description Language) = description au format XML des fonctionnalités du service web
 - UDDI (Universal Description Discovery and Integration) = normalisation d'une solution d'annuaire des services web

5

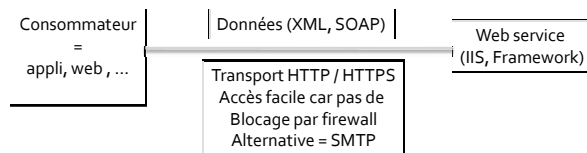
Web services

- ♦ Base XML
 - SOAP
 - WSDL
- ♦ Indépendance de la plateforme d'exécution du client et du service
- ♦ Indépendance du protocole de transport des messages
 - ⇒ Haut potentiel d'interopérabilité

6

Web services

- ♦ Architecture



7

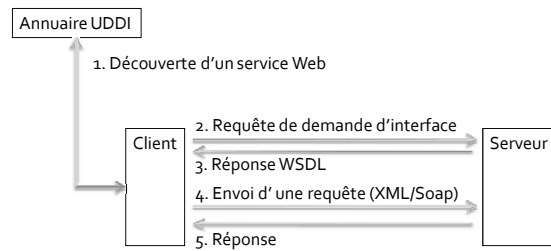
Web services

- ♦ Deux grandes façon d'aborder les Web services
 - Modèle de message requête / réponse
 - Mode de transport associé = HTTP
 - Modèle de message asynchrones, traitement par file d'attente
 - Mode de transport associé = SMTP
- ♦ Dans les 2 cas, peu de soucis de Firewall !

8

Web services

♦ Cycle de vie



9

Web services

♦ SOAP

- Langage XML utilisé pour la rédaction des messages échangés entre services
- Fournit un mode d'empaquetage standard des messages
- Standard de l'industrie
- Popularisé par des grands : Microsoft, IBM, Sun ...

10

Web services

♦ Avantages de SOAP

- Non étroitement lié à un langage
- Non couplé à un protocole de transport particulier
- Lié à aucune infrastructure d'objets distribués
- Utilise les standards existants de l'industrie
- Permet l'interopérabilité entre plusieurs environnements

11

Web services

♦ Anatomie d'un message SOAP

- Message SOAP = 1 enveloppe
- Définissant type d'encodage et namespaces
- Contenant
 - Un entête utilisé pour décrire le message
 - Un corps contenant le message
- Exemple

```

<soap:Envelope xmlns:soap=http://www.w3.org/2001/12/soap-envelope
  soap:encodingStyle=http://www.w3.org/2001/12/soap-encoding>
  <soap:Header>
    ...
  </soap:Header>
  <soap:Body>
    ...
  <soap:Fault>
    ...
  </soap:Fault>
  </soap:Body>
</soap:Envelope>
  
```

The diagram shows the structure of a SOAP message. It is an XML document with a root element <soap:Envelope>. The envelope has two attributes: xmlns:soap and soap:encodingStyle. Inside the envelope, there is a <soap:Header> element, followed by a <soap:Body> element, and optionally a <soap:Fault> element. The example shows the closing tags for the Header, Body, and Fault elements.

12

Web services - SOAP

- ♦ Élément Header
 - Facultatif
 - Destiné à passer des données qui ne sont pas destinées à être encodées dans le corps du message
 - Exemple : Si corps du message = compressé, header contient le type d'algo de compression utilisé
 - Peut servir pour
 - Authentification, relevé d'informations sur la sécurité, informations sur le routage, transactions, informations sur le paiement, ...

13

Web services - SOAP

- ♦ Élément Body
 - Obligatoire
 - Contient la charge utile du message
 - Pas de restriction sur le type d'encodage
 - Chaîne de caractères, tableaux d'octets, code XML
 - 1 seule contrainte
 - Caractères invalidant un document XML interdits

14

Web services - SOAP

- ♦ Élément Body
 - 2 orientation possibles
 - Procédure
 - Communication bidirectionnelle (RPC)
 - Définit action requise et ensemble paramètres d'entrée/sortie
 - Document
 - Communication unidirectionnelle
 - Ex : Bon de commande

15

Web services - SOAP

- ♦ Élément Fault
 - Définit un standard de communication de message d'erreur
 - Contient 4 éléments
 - Faultcode (obligatoire) = Code d'erreur SOAP
 - Faultstring (obligatoire) = Explication lisible pour un humain
 - Faultactor (optionnel) = Source exacte de l'erreur
 - Détail = Détail de l'erreur non lié au Body du message

16

Web services - SOAP

- ♦ Élément Fault
 - Principaux codes erreur

Code d'erreur	Description
VersionMismatch	Namespace invalide spécifié dans l'enveloppe
MustUnderstand	Attribut du header non compris
Client	Contenu du message = cause 1 ^{re} de l'erreur
Server	Contenu du message ≠ cause 1 ^{re} de l'erreur

17

Web services - SOAP

- ♦ Élément Fault
 - Exemple

```

...
<soap:Body>
  <soap:Fault>
    <faultcode>soap:Client</faultcode>
    <faultstring>The ISBN value contains invalid characters </faultstring>
    <faultactor>http://www.xyzcorp.com</faultactor>
    <detail> <mh:InvalidIsbnFaultDetail>
      <offending-value>19318224-D</offending-value>
      <conformance-rules>
        The first nine characters must be digits.
      </conformance-rules>
    </mh:InvalidIsbnFaultDetail>
    </detail>
  </soap:Fault>
</soap:Body>

```

18

Web services

- ♦ WSDL
 - Langage XML de description des services web
 - Description normalisée qui permet une utilisation inter-plateforme
 - Masquage du code du web service
 - Regroupe :
 - Les méthodes du service web
 - Les paramètres et les valeurs de retour
 - Le protocole de transfert (SOAP ou autre)
 - La localisation (url du service)

19

Web services - WSDL

- ♦ WSDL
 - Une description WSDL = balise définition qui contient 5 types d'éléments enfants :
 - Types = Définition de schéma des messages qui peuvent être envoyé ou reçu.
 - Message = Référence croisée associant message à sa définition dans le schéma
 - PortType = Jeu d'interfaces proposée par le service
 - Binding = Associe portType à un protocole particulier
 - Service = Ensemble des points d'entrée du service

20

Web services - WSDL

- ♦ WSDL
 - Utilisation d'un web service = Description WSDL de ensemble services mis à disposition
- ♦ Problème = nécessaire de connaître l'url WSDL
- ♦ Besoin = automatisation de la découverte
- ♦ Solution
 - UDDI
 - DISCO

21

Web services

- ♦ UDDI (Universal Description, Discovery and Integration)
 - standard (microsoft, IBM, sun, Oracle, HP...)
 - Annuaire d'informations administratives et techniques sur les services
 - Enregistrement via un opérateur (microsoft ou IBM)
 - Plusieurs types de recherche (identification d'entreprises, catégories,...) et recherche de services

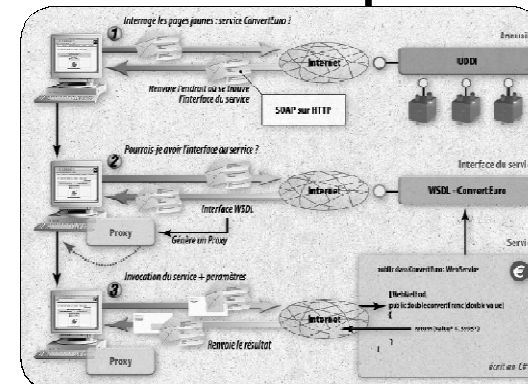
22

Web services

- ♦ DISCO
 - Création de Microsoft
 - Mécanisme décentralisé
 - Permet l'interrogation des services web proposés par un serveur
 - Implémente le paradigme de navigation

23

Web services -Récapitulatif



24

Web services

- ♦ Les raisons du succès
 - Utilisation d'HTTP
 - Protocole Internet
 - Serveur web présent dans la majorité des entreprises
 - Protocole généralement autorisé au niveau de parefeu
 - Protocole disponible sur toutes les plateformes
 - Utilisation d'XML
 - Massivement utilisé et reconnu
 - Permet de structurer l'information facilement

25

Web services

- ♦ Implémentations existantes
 - Microsoft .Net
 - Sun JavaONE : J2EE + Web services (WSDP = JAXP, JAXRPC, JAXM...)
 - Apache SOAP / Axis (JAXRPC)
 - IBM WSTK
 - Oracle, Bea, Iona, Enhydra ...

26

Web services en .NET

- ♦ Présentation
 - Web services = Pages ASP.net
 - Extension .asmx
- ♦ 3 format de liaisons des données
 - HTTP Get
 - HTTP Post
 - Format Soap
 - Avantage = Passage d'objet complexe type DataSet

27

Web services

- ♦ Gestion des protocoles de liaisons
 - Web.config

```
<configuration>
  <webServices>
    <protocols>
      <remove name ="HttpPost" />
      <remove name ="HttpGet" />
    </protocols>
  </webServices>
</configuration>
```

28

Web services

- ♦ Un décorateur de méthode pour tout faire
 - [WebMethod]
- ♦ Plusieurs attributs au décorateur
 - EnableSession = true
 - BufferResponse=false
 - CacheDuration=nombre de secondes
 - Description="une description"
 - MessageName="alias nom de méthode"
 - TransactionOption=TransactionOption.[Disabled ou Required ou Supported ou NotSupported ou RequiresNew]

29

Web services - WebMethod

- ♦ EnableSession
 - Valeur par défaut = false
 - Activer le stockage de données en session
- ♦ Avantage
 - Persistance
- ♦ Inconvénients
 - Impact sur les performances
 - Gestion de cookies persistant sur le client

30

Web services - WebMethod

- ♦ BufferResponse
 - Valeur par défaut = true
 - Activation du cache
 - Création du flux dans son ensemble puis envoi
 - A modifier avec prudence
 - Impact sur les performances
 - Impossible de cumuler avec l'utilisation d'entête SOAP

31

Web services - WebMethod

- ♦ CacheDuration
 - Valeur par défaut = 0
 - Gestion d'une durée de validité pour une réponse
- ♦ Utilité
 - Méthode de service très sollicité
 - Ex : météo

32

Web services - WebMethod

- ♦ Description
 - Spécifie la valeur de l'élément description dans le WSDL généré par ASP.net
- ♦ MessageName
 - Valeur utilisée pour la communication entre consommateur et service

33

Web services - WebMethod

- ♦ TransactionOption
 - Spécification du support transactionnel mis en place pour l'implémentation de la méthode
- ♦ Options
 - Disabled
 - Required
 - Supported
 - NotSupported
 - RequiresNew

34

Web services

- ♦ Décorateur de classe
 - Modification de l'ensemble des propriétés du WS
- ♦ WebService
 - Description
 - Name
 - Namespace

35

Web services

- ♦ Comportement par défaut
 - Appels synchrones
 - Possibilité d'asynchroniser l'ensemble
 - Erreur déclenchée en cas de suppression du service
 - Penser à la gestion d'exception
 - Obligation d'accéder à un serveur Web
 - Aucune sécurité des échanges

36

Web services

- ♦ Création d'un web service simple
- ♦ Objectif = Validation d'un numéro de carte bancaire
- ♦ Algo
 1. Multiplier chaque chiffre impair du numéro par 2.
 2. Exprimer le résultat du calcul modulo 9.
 3. Additionner tous les chiffres obtenus.
 4. Si la somme obtenue est un multiple de 10, le numéro de carte est valide.

37

Web services

- ♦ L'implémentation
 - Projet Web Service

```
public bool Validate(string cardNumber, DateTime
expDate)
{
    if (expDate >= DateTime.Today)
    {
        int total = 0; int temp = 0;
        char[] ccDigits = cardNumber.ToCharArray();

        for (int i = 0; i < cardNumber.Length; i++)
        {
            if (((i + 1) % 2) == 0) total +=
            int.Parse(ccDigits[i].ToString());
```

```
else
{
    temp =
    int.Parse(ccDigits[i].ToString()) * 2;
    if (temp > 9)
    {
        temp = (int)temp - 9;
    }
    total += temp;
}
}
if ((total % 10) == 0) return true;
else return false;
}
else
{
    return false;
}
}
```

38

Web services

- ♦ Transformation en Web service

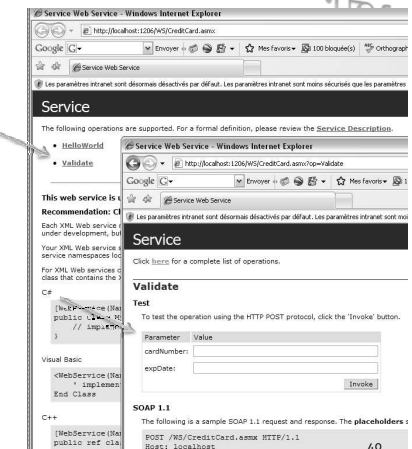
```
[WebMethod]
public bool Validate(string cardNumber, DateTime expDate)
{
    ...
}
```

Compiler et tester

39

Web services

Test d'un web service
depuis Visual studio 2005



40

Web services

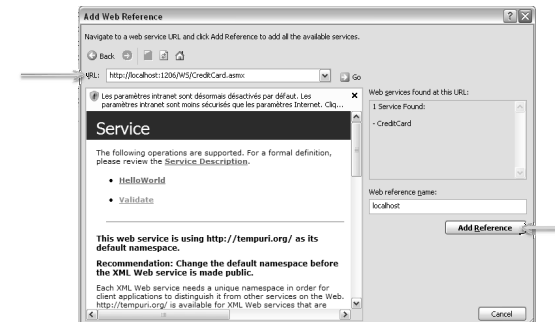
♦ Résultat

```
<?xml version="1.0" encoding="utf-8" ?>
<boolean
xmlns="http://tempuri.org/">false</boolean>
```

41

Web services

♦ Ajouter une référence depuis VS 2005



42

Web services

♦ Gestion de l'intelligence

```
protected void Button1_Click(object sender, EventArgs e)
{
    localhost.Service ws = new localhost.Service();
    if (ws.Validate(textBox1.Text, DateTime.Now.AddDays(1)))
    {
        SoapVersion
        Timeout
        Ticking
        UnsafeAuthenticatedConnectionSharing
        User
        UseDefaultCredentials
        UserAgent
        Validate
        ValidateAsync
        ValidateCompleted
        ma.s
        24JPN82
    }
    resultSetDataGrid.DataSource = ma.SendQueryToAmazon(textBox1.Text);
    resultSetDataGrid.VirtualItemCount = ma.TotalItems;
    resultSetDataGrid.DataBind();
}
```

43

Web services

♦ Génération d'une classe proxy

- WSDL.exe
- Utilitaire en ligne de commande
 - Syntaxe : WSDL.exe [options] urlDuWSDL
- Paramètres
 - /u:user = nom de l'utilisateur accédant au WS
 - /p:pass = passe de l'utilisateur accédant au WS

44