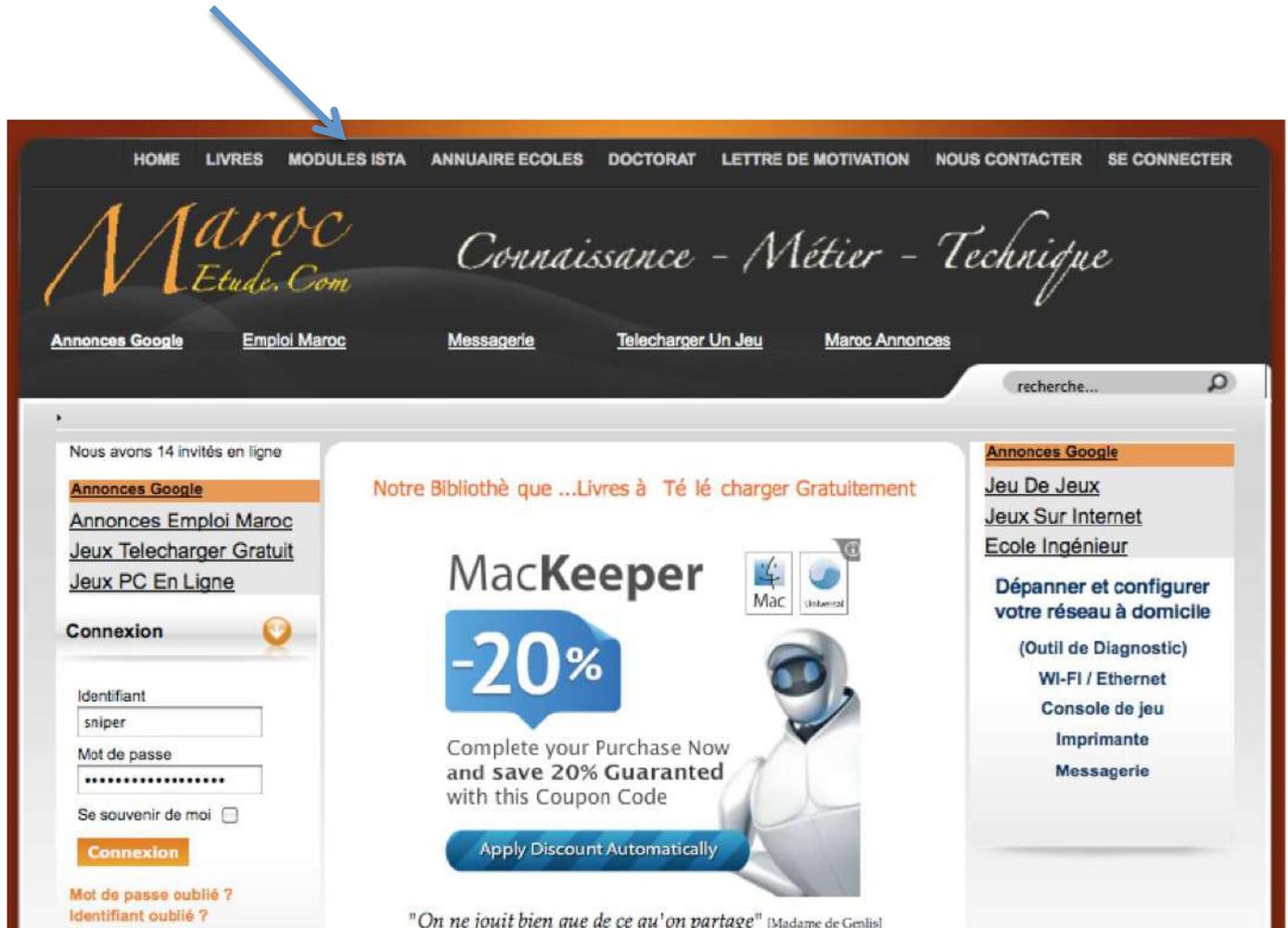


## PORTAIL DE LA FORMATION PROFESSIONNELLE AU MAROC

Télécharger tous les modules de toutes les filières de l'OFPPT sur le site dédié à la formation professionnelle au Maroc : [www.marocetude.com](http://www.marocetude.com)

Pour cela visiter notre site [www.marocetude.com](http://www.marocetude.com) et choisissez la rubrique :

### MODULES ISTA



The image shows a screenshot of the website Maroc Etude.Com. At the top, there is a navigation menu with the following items: HOME, LIVRES, **MODULES ISTA**, ANNUAIRE ECOLES, DOCTORAT, LETTRE DE MOTIVATION, NOUS CONTACTER, and SE CONNECTER. Below the menu is the website logo 'Maroc Etude.Com' and the tagline 'Connaissance - Métier - Technique'. There are also several utility links: Annonces Google, Emploi Maroc, Messagerie, Telecharger Un Jeu, and Maroc Annonces. A search bar is located on the right side of the header. The main content area features a central advertisement for MacKeeper with a -20% discount. To the left is a login section with fields for 'Identifiant' (containing 'sniper') and 'Mot de passe', and a 'Connexion' button. To the right is a sidebar with a search bar and a list of links under the heading 'Annonces Google', including 'Jeu De Jeux', 'Jeux Sur Internet', 'Ecole Ingénieur', 'Dépanner et configurer votre réseau à domicile', '(Outil de Diagnostic) WI-FI / Ethernet', 'Console de jeu', 'Imprimante', and 'Messagerie'. At the bottom of the page, there is a quote: '"On ne jouit bien que de ce qu'on partage"' [Madame de Genlis].

# LES SERVICES WEB XML / INTERET ET UTILISATION

Un service Web XML est un composant pouvant exposer ses méthodes sur le Web. Il offre un ensemble de fonctionnalités prêtes à l'emploi qu'un développeur pourra appeler et exploiter dans ses propres applications sans avoir à connaître leur implémentation. De cette manière, on pourra relier plusieurs applications entre elles à travers Internet et tirer partie d'applications existantes sans avoir à les développer à nouveau.

Pour communiquer, les services Web utilisent des protocoles Web et des formats de données standards ce qui les rend indépendants des plates-formes et des langages utilisés pour leur développement.

Les services Web XML n'ont pas d'interface utilisateur. On leur associe des méthodes et des fonctions offrant chacune un service spécifique et exploitable par d'autres applications. A chaque service offert sera associée une page descriptive exposant les services fournis dite **Contrat**.

## CREER UN SERVICE WEB XML

---

### Créer un service Web

- Pour insérer un service Web XML dans une application ASP.Net :

- Accéder au menu Projet / Ajouter un service Web
- Attribuer un nom au service XML dans la zone Nom et cliquer sur Ouvrir.

- Il est possible de créer un projet de services Web sous ASP.Net :

- Accéder au menu Fichier / Nouveau Projet
- Choisir, dans la fenêtre qui apparaît, Projets Visual Basic / Service Web ASP.Net
- Attribuer un nom au service XML dans la zone Emplacement et valider par le bouton OK.

Chaque projet Service Web créé contient par défaut un seul service Web nommé Service1 mais il peut en contenir plusieurs. Pour ajouter d'autres services web, accéder à Projet / Ajouter un service Web.

Remarque : Chaque service web est représenté par une page .asmx (**A**ctive **M**ethod **P**age / **M**icrosoft **E**Xtension File)

### Programmer un service Web

Dans un service Web, on peut créer des méthodes qui vont offrir des services à l'utilisateur final. Ces méthodes peuvent être des fonctions ou des procédures :

#### Pour créer une fonction :

```
<web method()>public Function NomFonction(ByVal|ByRef param1 as Type, ByVal|ByRef  
                                     param2 as Type,...) as type  
    ...  
    NomFonction=Valeur_Retour  
End Function
```

#### Pour créer une procédure :

```
<web method()>public Sub NomProcédure(ByVal|ByRef param1 as Type, ByVal|ByRef  
                                     param2 as Type,...)  
    ...  
End sub
```

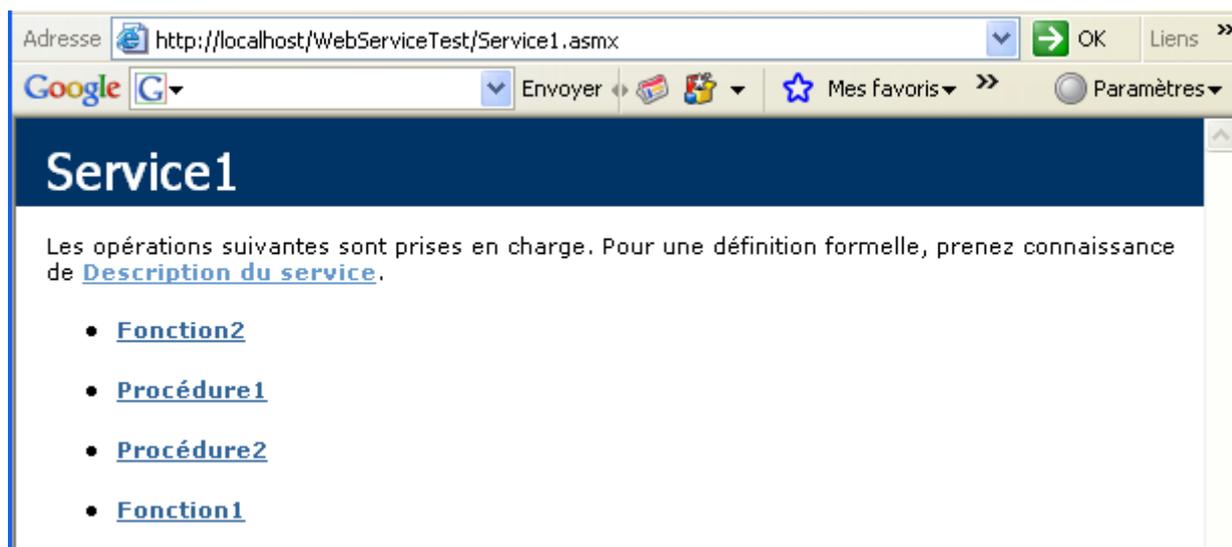
### Tester un service Web

Pour tester le service web, il faut l'exécuter dans le navigateur. Une page web correspondant à ce service s'affiche.

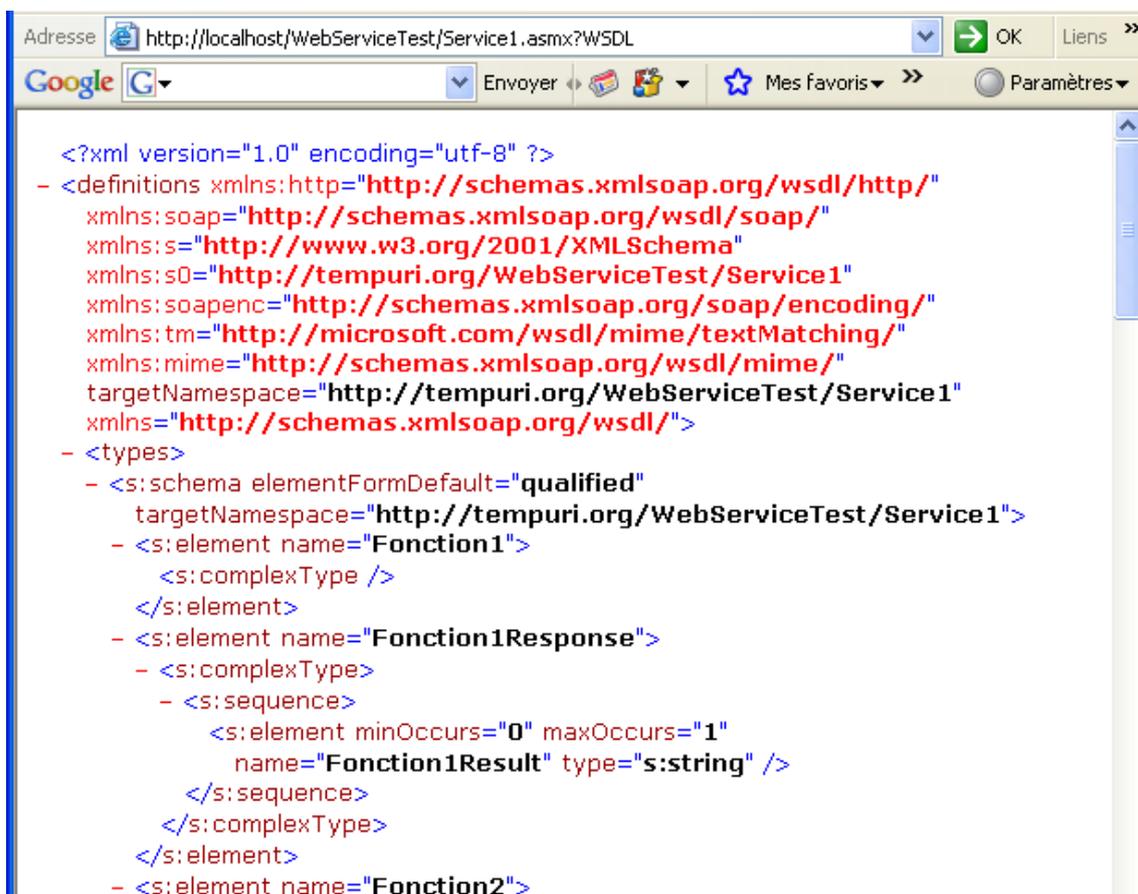
Toute page d'exécution d'un service web expose la liste des méthodes disponibles pour ce service. Elle inclut également un lien hypertexte "Description du Service" qui affiche le fichier WSDL correspondant à ce service.

Un fichier **WSDL(Web Service Description Language)** est un fichier au format XML qui définit la grammaire permettant de communiquer avec le service web. Les applications souhaitant utiliser le service Web vont communiquer avec ce fichier.

Exemple : La page d'exécution d'un service nommé Service1 contenant les méthodes Fonction1, Fonction2, Procédure1 et Procédure2, aura l'aspect suivant :



Ci-après une partie du fichier WSDL obtenu en cliquant sur le lien "Description du service" dans la fenêtre précédente :



Remarque : Il est possible d'accéder directement au contenu du fichier WSDL en rajoutant ?WSDL à l'adresse du service web dans la barre d'adresse du navigateur.

### Découvrir un service web

La découverte de services web est une technologie propre à Microsoft utilisée pour localiser les services Web XML dans un répertoire particulier sur un serveur particulier. **Les fichiers de découverte**

contiennent des liens (sous forme d'URL) vers des ressources nécessaires à l'exploitation d'un service web à travers Internet.

La découverte peut être statique (fichiers **.disco**) ou dynamique (fichiers **.vsdisco**) :

### Découverte statique :

Les fichiers de découverte sont générés lorsque l'utilisateur accède en ajoutant le mot clé ?Disco à l'URL d'exécution du service Web dans la barre d'adresse du navigateur.

Exemple : Ci-après une partie du fichier DISCO correspondant au service Web décrit précédemment :



```
<?xml version="1.0" encoding="utf-8" ?>
- <discovery xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://schemas.xmlsoap.org/disco/">
  <contractRef ref="http://localhost/WebServiceTest/Service1.asmx?
    wsdl" docRef="http://localhost/WebServiceTest/Service1.asmx"
    xmlns="http://schemas.xmlsoap.org/disco/scl/" />
  <soap address="http://localhost/WebServiceTest/Service1.asmx"
    xmlns:q1="http://tempuri.org/WebServiceTest/Service1"
    binding="q1:Service1Soap"
    xmlns="http://schemas.xmlsoap.org/disco/soap/" />
</discovery>
```

### Découverte dynamique

Les fichiers de découverte sont générés automatiquement par Microsoft .NET Framework. Il suffit que le système soit configuré dans ce sens (fichier de configuration Machine.config)

### **Tester une méthode d'un service web**

Pour tester une méthode appartenant à un service web, il suffit de cliquer sur le lien correspondant à son nom dans la page d'exécution du service Web. Une page de test demandant la saisie des paramètres correspondant à cette méthode apparaît. En saisissant des valeurs pour le test et en cliquant sur le bouton Appeler, le système affiche le résultat de la méthode encapsulé dans un format XML.

Exemple : Soit le service web Service1 décrit dans un exemple précédent. La méthode Fonction1 a la structure suivante :

```
<WebMethod()> Public Function Fonction1(ByVal largeur As Double, ByVal longueur As
  Double) As Double
  Fonction1 = largeur * longueur
End Function
```

Cette méthode reçoit la largeur et la longueur d'un rectangle et retourne la surface. En cliquant sur le lien correspondant à Fonction1 dans la page d'exécution du service, la page ci-après apparaît :

## Service1

Cliquez [ici](#) pour une liste complète des opérations.

### Fonction2

#### Test

Pour tester l'opération en utilisant le protocole HTTP POST, cliquez sur le bouton 'Appeler'.

Paramètre	Valeur
largeur:	<input type="text"/>
longueur:	<input type="text"/>

#### SOAP

Le texte suivant est un exemple de demande et de réponse SOAP. Les **espaces réservés** affichés doivent être remplacés par des valeurs réelles.

```
POST /WebServiceTest/Service1.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/WebServiceTest/Service1/Fonction2"

<?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>
    <Fonction2 xmlns="http://tempuri.org/WebServiceTest/Service1">
      <largeur>double</largeur>
      <longueur>double</longueur>
    </Fonction2>
  </soap:Body>
</soap:Envelope>
```

L'utilisateur saisit une largeur et une longueur et appuie sur le bouton Appeler. Le système affiche le résultat dans un fichier XML :



## PUBLIER UN SERVICE WEB XML

Une fois créé, un service Web XML, pour être reconnu sur Internet, peut être inscrit sur un site Web UDDI (Universal Description, Discovery and Integration). Il s'agit d'un **registre** commercial mondial conçu pour héberger des informations sur les entreprises et leurs services Web.

Ce site répond aux spécifications du consortium **UDDI.org** (ces spécifications définissent un mode de publication et de détection des informations standard). Il est hébergé gratuitement par plusieurs serveurs appartenant à des opérateurs tel Microsoft et IBM.

Les fournisseurs de services Web peuvent inscrire leurs services Web sur tel ou tel serveur et des systèmes de réplication sont en place pour synchroniser régulièrement les serveurs entre eux via des canaux sécurisés.

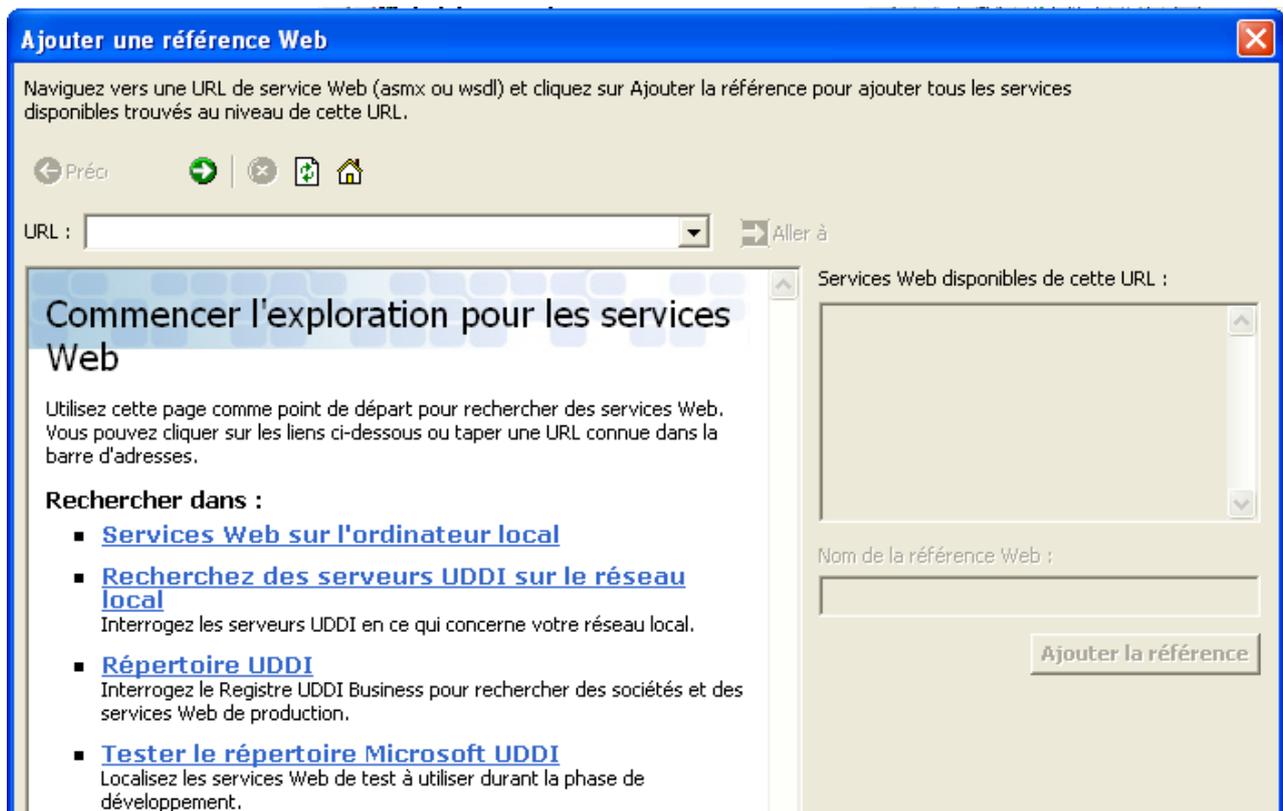
Les sites UDDI ne stockent pas les services web, ils se contentent de présenter les ressources et de diriger les utilisateurs intéressés vers ces ressources. En fait, il présente les URL des documents DISCO relatifs à ces services web. L'utilisateur en sélectionnant un service web accèdera donc à un document disco qui permet de localiser les fichiers WSDL associés à ce service.

## EXPLOITER UN SERVICE WEB XML

Pour utiliser un service Web dans une application, il faut d'abord ajouter une référence web pour ce service et ensuite instancier un objet pour le référencer.

### Pour ajouter une référence Web

Dans le projet ASP.Net où on souhaite utiliser le service web, on clique dans l'explorateur de solutions avec le bouton droit sur Références et on choisit "Ajouter une référence Web". La fenêtre qui apparaît offre la possibilité de rechercher des services Web à différents emplacements :



En choisissant un emplacement pour la recherche, la liste des services Web disponibles s'affiche. Il faut choisir un service et affecter un nom à la référence Web et cliquer sur le bouton "Ajouter la référence".

En ajoutant une référence à un service web dans une application, un objet **proxy** est créé (fichier `reference.vb` ou `reference.cs`). Cet objet permet à l'application d'accéder à un service Web comme s'il était stocké sur l'ordinateur local. Pourtant le proxy ne contient pas le code du service Web mais seulement la logique de transport correspondant à ce service.

L'accès d'une application et un service Web se fait pratiquement entre le proxy et le fichier WSDL qui, pour communiquer entre eux, utilisent le protocole **SOAP (Simple Object Access Protocol)** qui repose sur XML.

### **Pour instancier l'objet référençant le service**

Il faut d'abord appeler l'espace de nom correspondant à la référence Web concernant le service à utiliser et ensuite instancier une variable pour référencer ce service.

```
Imports NomApplication.NomRéférenceWeb
```

```
...
```

```
Dim Réf_Service as new Nom_Service
```

Exemple : Dans une application `TestServiceWeb`, en utilisant une référence Web nommée `MonPremierService` qui référence le service Web décrit précédemment (nommé `Service1`), on peut écrire le code suivant pour utiliser la méthode `Fonction1` :

```
Imports TestServiceWeb. MonPremierService
```

```
...
```

```
Dim S as new Service1
```

```
Dim large as double=10
```

```
Dim longue as double=20
```

```
Dim Surface as double=S.Fonction1(large, longue)
```