

# Utiliser le Service web HYPERPLANNING (en version Réseau)


Pour utiliser le service web, vous devez utiliser la version Réseau et disposer d'une licence HYPERPLANNING.net.

## Présentation et démarrage

### Qu'est-ce qu'un service web ?

Un service web se présente comme une application web sauf qu'il est utilisé par un programme informatique et non par un individu. Il existe de nombreux exemples de services web, tous très spécialisés : des services de conversion monétaire, de météorologie, etc.

### A quoi sert HYPERPLANNING Service web ?

HYPERPLANNING Service web  permet d'échanger des données avec d'autres programmes informatiques indépendamment des plateformes et langages utilisés. Il permet notamment d'étendre et d'automatiser les fonctions d'import/export en abandonnant les contraintes liées à l'échange de données sous forme de fichiers.

### Compétences requises

Pour écrire un script ou un programme utilisant un service web, il faut connaître les bases de programmation comme :

- Qu'est ce qu'une variable, qu'est ce qu'un type de variable ?
- Qu'est ce qu'une routine / procédure / fonction ?

Vous pouvez écrire un script (VbScript, Javascript, Python, une macro VBA sous Microsoft Office, etc.) ou utiliser un environnement de programmation (Java, .net, C++, Pascal, etc.) selon vos besoins. Votre programme tourne sur la plateforme de votre choix.


### Normes employées


Nous avons choisi de nous appuyer sur des standards pour l'implémentation d'HYPERPLANNING service web : le service est décrit par un document WSDL et communique avec les applications en utilisant le protocole SOAP sur une connexion HTTP. Tous les langages de programmation récents intègrent ce qu'il faut pour interroger un serveur de ce type sans connaissances préalables de WSDL, SOAP ou HTTP, et il existe des extensions pour le faire également sur la quasi totalité des langages de programmation plus anciens.

### Configuration requise

MS-Windows NT4, 2000, XP ou 2003, en version serveur ou non. La version serveur est recommandée car sur celle-ci le nombre de connexions simultanées sur la machine n'est pas limité.

### Démarrer HYPERPLANNING Service web

Utilisez le raccourci  ou la commande **Démarrer > Programmes > HYPERPLANNING Réseau 2009 > Service web HYPERPLANNING 2009** (si l'application a été installée sur le poste).

Si les paramètres de connexions ont été correctement définis, le Service web se connecte automatiquement au Contrôleur ou au serveur HYPERPLANNING. Dès que la connexion est établie, le Service web affiche la **Liste des plannings**  mis en service sur le réseau.

### Créer un service Windows pour HYPERPLANNING Service web

Dès qu'un planning est chargé sur le serveur, ce dernier peut s'exécuter comme un service Windows, c'est-à-dire s'ouvrir au démarrage de la machine où il est installé. Pour cela, activez la commande **Fichier > Créer le service Windows associé** qui prendra effet à partir du prochain démarrage de l'ordinateur.

### Ne plus exécuter HYPERPLANNING Service web comme un service Windows

Pour cela, activez la commande **Fichier > Supprimer le service Windows associé** ou **Supprimer tous les services Windows HYPERPLANNING** si vous souhaitez agir sur toutes les applications HYPERPLANNING. Pour utiliser cette commande, un planning doit être chargé sur le Serveur.

## Connexion à la base

HYPERPLANNING service web accède à la base de données en se connectant au planning associé.

Par défaut, le service web est paramétré pour se connecter directement au contrôleur. Mais vous pouvez modifier tous les paramètres de connexion selon vos besoins. Pour effectuer des modifications de ces paramètres, aucun planning ne doit être ouvert sur le service web.

### Connexion au Contrôleur

La connexion au contrôleur permet de choisir, parmi les plannings mis en service, celui auquel vous souhaitez vous connecter. 2 protocoles peuvent être utilisés pour établir la connexion :

- UDP : s'utilise uniquement pour les connexions venant de l'intérieur du réseau.
- TCP/IP : s'utilise pour tous types de connexions, notamment celles provenant de l'extérieur du réseau.

### Problème de connexion ?

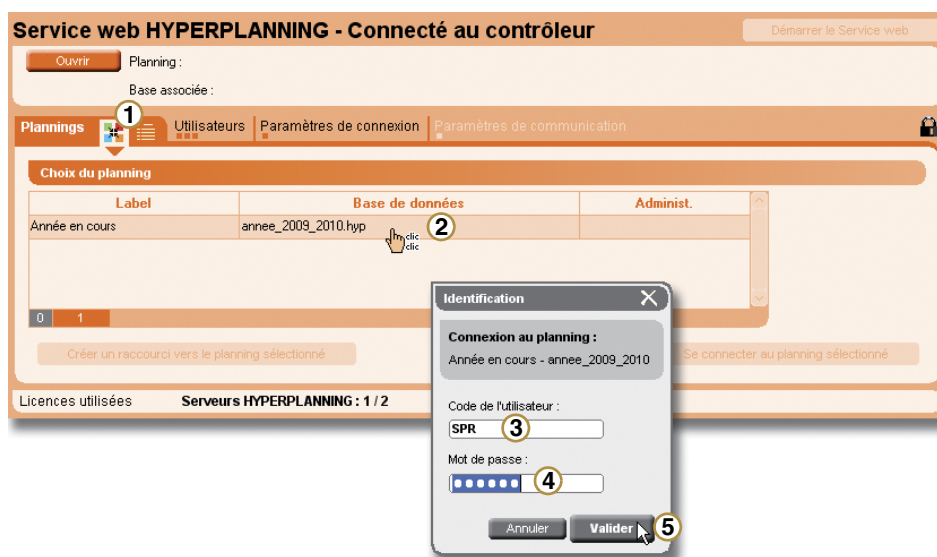
Si la mention Recherche du contrôleur s'affiche, cela signifie que la connexion au contrôleur n'arrive pas à se faire. Dans ce cas, vérifiez que :


- le Contrôleur est lancé ;
- le Contrôleur est activé ;
- les **Paramètres de connexion** au Contrôleur sont bons ("*Connexion au Contrôleur*", page 12).



### Choix du planning

Si le Service web se connecte à HYPERPLANNING Contrôleur , il vous revient de choisir votre planning :



1. Lancez HYPERPLANNING Service web, il s'ouvre sur l'affichage de **Choix du planning** .
2. Dans la liste des plannings, double-cliquez sur un planning pour vous connecter à la base qui lui est associée.
3. Saisissez votre code d'utilisateur.
4. Saisissez votre mot de passe. Il est indispensable d'avoir un mot de passe pour faire fonctionner le Client SOAP.
5. Cliquez sur le bouton **Valider**. Le Service web est démarré et le planning que vous avez sélectionné est immédiatement publié.

## Connexion directe au Serveur

Ce type de connexion permet de se connecter directement au Serveur qui abrite le planning, sans passer par le Contrôleur. Cela implique que vous connaissiez les coordonnées de connexion au planning voulu (adresse IP et port TCP) La connexion peut se faire depuis l'intérieur ou l'extérieur du réseau. Si vous utilisez ce type de connexion, le Service web se connecte automatiquement au Serveur lors de son lancement.

### Problème de connexion ?

Si le message ci-contre s'affiche, cela signifie que la connexion au Serveur n'arrive pas à se faire. Dans ce cas, vérifiez que :

- le Serveur a un planning **En service** ;
- les **Paramètres de connexion** au Serveur sont corrects (*"Connexion au Serveur"*, page 13).

## Identification de l'utilisateur

Seuls le SPR, les administrateurs et les utilisateurs dont l'option **Utiliser le service web** est cochée dans leur profil d'utilisation (*"Définir les profils d'autorisation"*, page 41) peuvent se connecter au planning, après identification. Les informations personnelles, notamment les informations d'extraction, sont conservées lors de l'utilisation du service web. Ce sont les mêmes identifiants qui permettront d'identifier les programmes communicant avec le Service web. Le Code et le Mot de passe de chaque utilisateur sont définis par les administrateurs du planning dans l'**Administration des utilisateurs** (*"Gestion des utilisateurs"*, page 40).

## Paramètres de communication

Les **Paramètres de communication** permettent de définir sur quel port et à quelle adresse le Service web va communiquer. Vous devrez fournir ces informations à vos utilisateurs pour qu'ils puissent communiquer avec le Service web.

Port d'écoute d'HYPERPLANNING Service web.

Adresse IP de la machine où est installé HYPERPLANNING Service web. C'est l'URL que les utilisateurs devront saisir dans leur navigateur pour accéder aux documents de référence.

Laissez l'option **Mode «RPC:Encoded»** sauf si vous utilisez un langage qui nécessite un Encodage en **«RPC:Literal»**. Attention, le document Wsdl sera modifié en conséquence de votre choix pour l'une ou l'autre option.

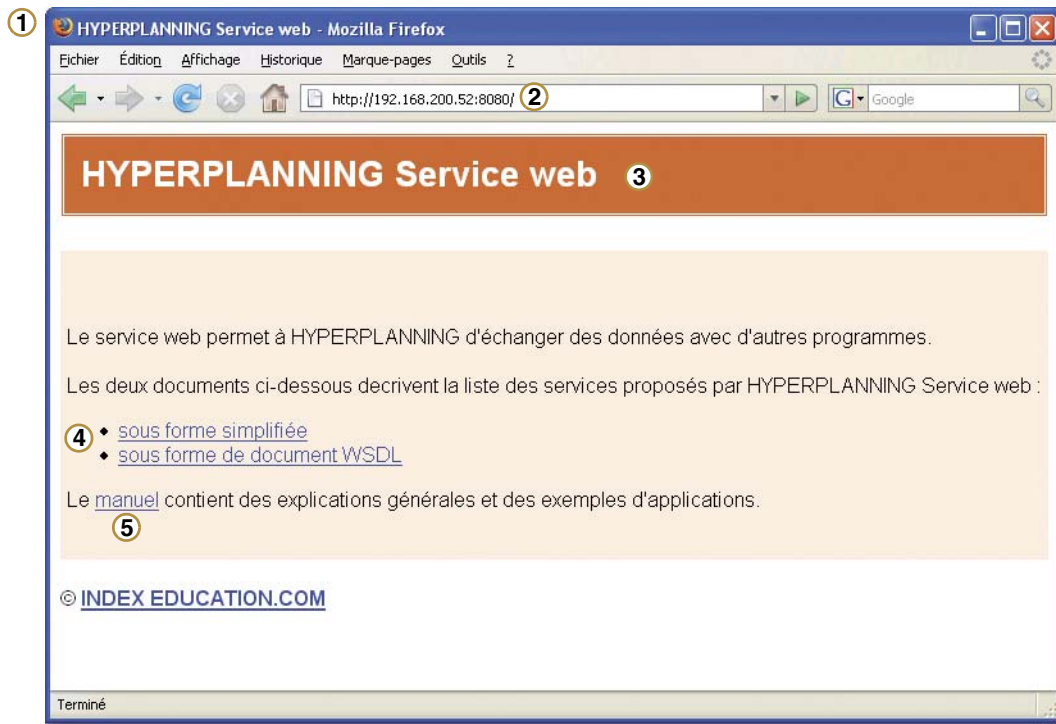
Pour éviter des temps de réponse trop longs ou des piratages.

Permet d'inclure une authentification dans les messages sans que les utilisateurs ne soient trop rapidement bloqués.

Cochez cette option si vous souhaitez que le service web ne réponde pas avec les types définis dans le document WSDL mais avec les types de base définis dans XML Schema, connu nativement de la plate-forme Microsoft.net.

**Remarque :** il est convenu que le Service web et le programme qui s'y connecte communiquent par « messages », qu'il s'agisse d'une requête provenant du programme ou d'une réponse émanant du Service web.

## Accéder au document WSDL



1. Ouvrez un navigateur internet.
2. Saisissez l'URL de votre HYPERPLANNING Service web, suivi du numéro de port.
3. La page de présentation du Service web apparaît.
4. Pour écrire vos scripts et vos programmes, utilisez la liste des services proposés par HYPERPLANNING Service web, disponible sous deux formes :
  - sous forme simplifiée
  - sous forme de document WSDL : sous cette forme, vous disposez aussi de la liste des types utilisés. Si votre navigateur est récent, le document WSDL s'affiche sous forme arborescente, ce qui facilite la lecture du document. Les appels sont classés par service et par interface.
5. En cas de besoin, n'hésitez pas à consulter le manuel en ligne du Service web, disponible d'un clic.

## Exemples d'utilisation

Les programmes accédant au service web doivent s'identifier auprès du service web par authentification HTTP « Basic ». L'identifiant et le mot de passe utilisés doivent être les mêmes que ceux de l'utilisateur connecté.

Tous les exemples suivants sont disponibles dans le manuel à l'adresse <http://adresseduseur:8080/>. Ainsi, vous pouvez effectuer des copier-coller qui vous feront gagner du temps.

### Récupérer le nombre de matières depuis VbScript

Saisissez le programme suivant dans le bloc-notes, modifiez les constantes **C\_WSDL**, **C\_Identifiant** et **C\_MotDePasse** et enregistrez le fichier sous le nom **ExempleVbScript.vbs**.

```
Option Explicit

' URL du document WSDL de HYPERPLANNING Service web
Const C_WSDL = "http://localhost:8080/wsd1"
' L'identifiant et le mot de passe doivent être en majuscules
Const C_Identifiant = "TT"
Const C_MotDePasse = "TTTTTTTT"

On Error Resume Next
Dim IAdmin
Set IAdmin = CreateObject("MSSOAP.SOAPClient")
With IAdmin
    Call .MSSoapInit(C_WSDL, "HpSvcWAdmin", "PortAdmin")
    .ConnectorProperty("AuthUser") = C_Identifiant
    .ConnectorProperty("AuthPassword") = C_MotDePasse
    WScript.Echo "Connecté à " & .Version () & " avec succès."
    If Err Then WScript.Echo .FaultString
End With
Set IAdmin = Nothing

Dim IMatieres
Set IMatieres = CreateObject("MSSOAP.SOAPClient")
With IMatieres
    Call .MSSoapInit(C_WSDL, "HpSvcWDonnees", "PortMatieres")
    .ConnectorProperty("AuthUser") = C_Identifiant
    .ConnectorProperty("AuthPassword") = C_MotDePasse
    WScript.Echo "Il y a " & .NombreMatiere () & " matières dans la base."
    If Err Then WScript.Echo .FaultString
End With
Set IMatieres = Nothing
```

Vous pouvez exécuter ce script en ligne de commande en tapant **cscript ExempleVbScript.vbs**, cela renvoie le nombre de matières dans la base HYPERPLANNING publiée.

### Créer et lister des matières depuis VBA

VBA, c'est Visual Basic for Applications, le basic intégré dans tous les produits de la suite Microsoft Office. Dans Excel, cliquez sur **Outils > Macro > Visual Basic Editor**, puis tapez le code suivant dans **Insertion > Module** :

```
Option Explicit

'*****
' Code générique pour accéder à HYPERPLANNING Service web
'*****

Dim G_WSDL As String
Dim G_Identifiant As String
Dim G_MotDePasse As String

Function GetParametres() As Boolean
    ' Demande l'url du document WSDL, l'identifiant et le mot de passe si nécessaire
    Dim LAnnulation As Boolean
    LAnnulation = False

    If G_WSDL = "" Then
        G_WSDL = InputBox("Document WSDL décrivant HYPERPLANNING Service web", _
            "Information sur le serveur", _
            "http://<adresse du serveur>:<port d'écoute>/wsdl")
        LAnnulation = (G_WSDL = "")
    End If
    If (G_Identifiant = "") And (Not LAnnulation) Then
        ' Toujours en majuscules
        G_Identifiant = UCase(InputBox("Identifiant de l'utilisateur du service web", _
            "Identification"))
        LAnnulation = (G_Identifiant = "")
    End If
    If (G_MotDePasse = "") And (Not LAnnulation) Then
        ' Toujours en majuscules
```

```
G_MotDePasse = UCase(InputBox("Mot de passe de l'utilisateur du service web", _
    "Identification"))
LAnnulation = (G_MotDePasse = "")
End If
If LAnnulation Then
    G_WSDL = ""
    G_Identifiant = ""
    G_MotDePasse = ""
End If
GetParametres = Not LAnnulation
End Function

Function InterfaceDAccesHyperPlanning(ANomDuService As String, _
    ANomDuPort As String) As Object
    ' Construit un objet d'accès à HYPERPLANNING Service web
    ' Pour connaître le nom du port à utiliser, consultez les sections
    ' <service> dans le document WSDL
    Set InterfaceDAccesHyperPlanning = CreateObject("MSSOAP.SOAPClient")
    With InterfaceDAccesHyperPlanning
        Call .MSSoapInit(G_WSDL, ANomDuService, ANomDuPort)
        .ConnectorProperty("AuthUser") = G_Identifiant
        .ConnectorProperty("AuthPassword") = G_MotDePasse
    End With
End Function

Function InterfaceDAccesAuxDonneesHyperPlanning(ANomDuPort As String) As Object
    ' Utilisation type :
    ' Dim IMatieres As Object
    ' Set IMatieres = InterfaceDAccesAuxDonneesHyperPlanning("PortMatieres")
    ' ... utilisation de IMatieres ...
    ' Set IMatieres = Nothing
    Set InterfaceDAccesAuxDonneesHyperPlanning = _
        InterfaceDAccesHyperPlanning("HpSvcWDonnees", ANomDuPort)
End Function

Function TypeHpEnChaine(oXMLDOMNodeList As Object) As String
    ' Les types définis dans le document WSDL (section <types>) ne sont pas
    ' accessibles depuis VBA. Les valeurs de retour des appels au service web sont
    ' alors de type IXMLDOMNodeList
    ' Cette fonction en récupère la valeur chaîne
    TypeHpEnChaine = oXMLDOMNodeList.Item(0).Text
End Function

'*****
' Exemples de code utilisateur
'*****

Sub CreerMatiereHyperPlanning()
    If GetParametres() Then
        ' On affiche le numéro de version de HYPERPLANNING Service web
        Dim IAdmin As Object
        Set IAdmin = InterfaceDAccesHyperPlanning("HpSvcWAdmin", "PortAdmin")
        MsgBox "Connecté à " & IAdmin.Version()
        Set IAdmin = Nothing

        ' Crée puis supprime une nouvelle matière en donnant le nombre de matières
        ' avant et après chaque étape
        Dim IMatieres As Object
        Set IMatieres = InterfaceDAccesAuxDonneesHyperPlanning("PortMatieres")
        With IMatieres
            MsgBox "Il y a " & .NombreMatieres() & " matières dans la base."
            Dim oMatiere As Object
            Set oMatiere = .CreerMatiere("Toto", "TT")
            MsgBox "Création réussie : il y a " & _
                .NombreMatieres() & " matières dans la base."
            Call .SupprimerMatiere(oMatiere)
            MsgBox "Suppression réussie : il y a à nouveau " & _
                .NombreMatieres() & " matières dans la base."
        End With
        Set IMatieres = Nothing
    End If
End Sub

Sub ListerLesMatiereHyperPlanning()
    ' Interroge HYPERPLANNING pour récupérer la liste des matières
    If GetParametres() Then
        Dim IMatieres As Object
        Set IMatieres = InterfaceDAccesAuxDonneesHyperPlanning("PortMatieres")

        ' Efface le contenu de la feuille de calcul
        Cells.ClearContents
        ' Titres
        With Rows(1)
            ' C'est du texte
            .NumberFormat = "@"
            .HorizontalAlignment = xlCenter
            .Font.Bold = True
        End With
        Cells(1, 1).Value = "Clé matière"
        Cells(1, 2).Value = "Code"
        Cells(1, 3).Value = "Libellé"
```

```

Cells(1, 4).Value = "Libellé long"

Dim LCles() As Object
Dim LCodes() As String
Dim LLibelles() As String
Dim LLibellesLongs() As String
With IMatieres
    ' Deux premières requêtes pour obtenir toutes les clés matières
    ' puis les trier
    LCles = .TrierTableauDeMatiereParLibelleEtCode(.ToutesLesMatiere())
    ' Trois autres requêtes au service web, une par colonne
    LCodes = .CodesTableauDeMatiere(LCles)
    LLibelles = .LibellesTableauDeMatiere(LCles)
    LLibellesLongs = .LibellesLongsTableauDeMatiere(LCles)
End With

' Tout le reste c'est pour l'affichage
Dim LIndice As Integer
' Delta entre indices dans LCles et numéros de lignes dans la feuille de calcul
Dim LDelta As Integer
LDelta = LBound(LCles) - 2
For LIndice = LBound(LCles) To UBound(LCles)
    ' Clé matière
    With Cells(LIndice - LDelta, 1)
        .NumberFormat = "0"
        .HorizontalAlignment = xlRight
        .IndentLevel = 1
        .Value = TypeHpEnChaine(LCles(LIndice))
    End With
    ' Code
    With Cells(LIndice - LDelta, 2)
        .NumberFormat = "@"
        .HorizontalAlignment = xlLeft
        .IndentLevel = 1
        .Value = LCodes(LIndice)
    End With
    ' Libellé
    With Cells(LIndice - LDelta, 3)
        .NumberFormat = "@"
        .HorizontalAlignment = xlLeft
        .IndentLevel = 1
        .Value = LLibelles(LIndice)
    End With
    ' Libellé long
    With Cells(LIndice - LDelta, 4)
        .NumberFormat = "@"
        .HorizontalAlignment = xlLeft
        .IndentLevel = 1
        .Value = LLibellesLongs(LIndice)
    End With
Next LIndice

Set IMatieres = Nothing
End If
End Sub

```

Quittez l'éditeur Visual Basic et enregistrez le document.

Depuis le menu **Outils > Macro > Macros ...** vous pouvez maintenant exécuter les macros **CreerMatiereHyperPlanning** et **ListerLesMatiereHyperPlanning**.

## Créer et lister les matières depuis Borland Delphi

Pour qu'un programme écrit avec Borland Delphi communique sans erreur avec HYPERPLANNING Service web, ce dernier doit être en mode RPC / Encoded et non en mode RPC / Literal ("*Paramètres de communication*", page 3).

Borland Delphi sait générer du code d'accès à un service web depuis son document WSDL. Dans le menu **Fichier > Nouveau > Autre**, choisissez **Projets Delphi > Services Web > Importateur WSDL**. Cela génère l'unité **Delphi wsdl.pas**. Dans les options vous pouvez décocher **Mapper String en WideString**, HYPERPLANNING ne gérant pas les caractères unicode.

L'unité Delphi wsdl.pas peut aussi être générée en ligne de commande comme suit :

**WSDLIMP -P -Ow- http://serveur:8080/wsdl**

Créez maintenant ce programme en ligne de commande (ExempleDelphi.dpr) :

```

program ExempleDelphi;

{$APPTYPE CONSOLE}

uses
    ActiveX, // Pour CoInitialize / CoUnInitialize
    wsdl,    // Généré par Delphi depuis le document WSDL
    SysUtils;
(*****)
(* Crée puis supprime une nouvelle matière en donnant le nombre de matières *)
(* avant et après chaque étape *)
(*****)

```

```
procédure CreerMatiereHyperPlanning ();

var
  IAdmin      : IHpSvcWAdmin;
  IMatieres   : IHpSvcWMatieres;
  LMatiere    : THpSvcWCleMatiere;

begin
  // On affiche le numéro de version de HYPERPLANNING Service web
  IAdmin := GetIHpSvcWAdmin ();
  Writeln ('Connecté à ' + IAdmin.Version ());
  IAdmin := nil;

  IMatieres := GetIHpSvcWMatieres ();
  with IMatieres do
  begin
    Writeln (Format ('Il y a %d matières dans la base.',
                    [NombreMatieres ()]));
    LMatiere := CreerMatiere ('Toto', 'TT');
    Writeln (Format ('Création réussie : il y a %d matières dans la base.',
                    [NombreMatieres ()]));
    SupprimerMatiere (LMatiere);
    Writeln (Format ('Suppression réussie : il y a à nouveau %d matières dans la base.',
                    [NombreMatieres ()]));
  end;
  IMatieres := nil;
end;

(*****
(* Interroge HYPERPLANNING pour récupérer la liste des matières          *)
*****)

procédure ListerLesMatieresHyperPlanning ();

const
  C_FormatColonnes = '%0:3d %1:-8s %2:-20s %3:s';

var
  IMatieres      : IHpSvcWMatieres;
  LCles           : ArrayOfTHpSvcWCleMatiere;
  LCodes,
  LLibelles,
  LLibellesLongs : ArrayOfstring;
  LIndice         : cardinal;

begin
  IMatieres := GetIHpSvcWMatieres ();
  with IMatieres do
  begin
    LCles       := TrierTableauDeMatieresParLibelleEtCode (ToutesLesMatieres ());
    LCodes      := CodesTableauDeMatieres (LCles);
    LLibelles   := LibellesTableauDeMatieres (LCles);
    LLibellesLongs := LibellesLongsTableauDeMatieres (LCles);
  end;
  for LIndice := low (LCles) to high (LCles) do
    Writeln (Format (C_FormatColonnes,
                    [LCles [LIndice],
                     LCodes [LIndice],
                     LLibelles [LIndice],
                     LLibellesLongs [LIndice]]));
  end;
  IMatieres := nil;
end;

(*****

begin
  // CoInitialize / CoUnInitialize nécessaires pour msxml.dom.CreateDOMDocument
  CoInitialize (nil);
  try
    CreerMatiereHyperPlanning ();
    ListerLesMatieresHyperPlanning ();
  finally
    CoUnInitialize ();
  end;
end.
```

Vous pouvez maintenant compiler le projet et l'exécuter en ligne de commande.

**D'autres langages peuvent être utilisés comme C# et Java. Des exemples sont disponibles sur notre site :**  
[www.index-education.fr](http://www.index-education.fr)

## Conseils aux programmeurs

### Débogage

Il existe des outils commerciaux et des outils gratuits de bonne qualité pour déboguer un programme consommant un service web, toutefois ils demandent une bonne connaissance du protocole SOAP pour être utilisés.

Certains de ces outils agissent comme un proxy et permettent d'analyser le trafic entre votre programme et le service web. Attention, ces outils ne vous montrent pas toujours tout. Par exemple ils peuvent masquer la demande de document WSDL au serveur, voire renvoyer un document WSDL modifié, ou masquer les entêtes HTTP alors que ce sont eux qui contiennent les informations d'authentification.

### Erreurs fréquentes

Erreur HTTP 401 : l'identifiant et le mot de passe n'ont pas été transmis au service web.

Erreur HTTP 403 : l'identifiant et le mot de passe ont été transmis au service web mais ne sont pas ceux attendus. Attention, l'identifiant et le mot de passe sont toujours en majuscules dans les bases HYPERPLANNING.

### A propos de l'authentification HTTP «Basic»

Que faire si votre langage de programmation ne gère pas l'authentification HTTP "basic" ?

Il est important que l'accès au service web reste protégé par un identifiant et un mot de passe. Si votre langage de programmation ne gère pas l'authentification HTTP "basic" vous pouvez spécifier l'identifiant et le mot de passe dans un document WSDL local. Voici la marche à suivre :

1. Accéder au document WSDL du serveur avec un navigateur internet `http://adresseduseveur:8080/wsdl` et enregistrez ce document localement.
2. Ouvrez ensuite le document sauvé dans un éditeur de texte et ajoutez l'authentification pour tous les éléments `soap:address`, par exemple `<soap:address location="http://TT:TTTTTTT@serveur:8080/Matieres" />` Attention, les identifiants et mots de passe doivent être en majuscules.
3. Votre programme peut maintenant utiliser ce document WSDL au lieu de celui du serveur et n'a plus besoin de s'authentifier.

N'oubliez pas à chaque mise à jour de HYPERPLANNING Service web de répéter ces opérations pour mettre à jour votre document WSDL local.

**Remarque pour les utilisateurs Windows :** depuis la mise à jour MS04-004 Cumulative Security Update for Internet Explorer (832894), la syntaxe `http://TT:TTTTTTT@serveur:8080/Matieres` est rejetée par défaut. Vous pouvez être amené à autoriser cette syntaxe pour un programme particulier en insérant une clef dans le registre Windows de votre ordinateur.

### À propos de MSSOAP (Microsoft Soap Toolkit)

Les exemples VbScript et VBA ci dessus font appel à la partie cliente du Microsoft SOAP Toolkit 2.0 qui est native sous Windows XP Professionnel (Cf. *Overview of SOAP Client in Windows XP* – en anglais). C'est bien commode car les exemples peuvent fonctionner sans aucune installation préalable.

Plusieurs problèmes peuvent toutefois survenir à l'exécution :

- L'appel `ConnectorProperty("AuthPassword")` échoue lorsque le mot de passe est vide.
- Parfois la connexion au service web échoue avec une erreur 5300 (CONNECTION\_ERROR) sans qu'aucune demande de connexion ne soit tentée.

Il semble que basculer la clé de registre

`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSOAP\30\SOAPISAP\NoNagling` de 0 à 1 améliore les performances tout en éliminant ces erreurs de connexion.

La dernière version de Microsoft Soap Toolkit est Microsoft Soap Toolkit 3, et ne sera prochainement plus supportée par Microsoft. Rappelons que vous êtes libre d'utiliser le langage de programmation de votre choix sur la plateforme de votre choix.

### À propos de WS-I (Web Services Interoperability)

WS-I est une organisation qui propose une autre norme appelée Basic Profile pour les services web que celle du World Wide Web Consortium.

HYPERPLANNING service web n'est pas compatible WS-I et ce pour plusieurs raisons :

Dans HYPERPLANNING service web, la couche de traitement HTTP ignore tout de la couche de traitement SOAP. Ainsi le code d'erreur HTTP vaut autre chose que 200 (opération réussie) uniquement si une erreur a lieu dans la couche de traitement HTTP. En cas d'erreur dans la couche de traitement SOAP, le code d'erreur HTTP vaut 200 et le document retourné contient un élément `soap:error`. Pour WS-I, ce code devrait être différent de 200 dans ce cas.

Les types tableaux définis par HYPERPLANNING service web le sont par restriction du type soapenc:Array avec l'attribut wsdl:arrayType.

La page de code des réponses de HYPERPLANNING service web est iso-8859-1 en français alors que WS-I n'autorise que UTF-8.

WS-I impose singulièrement la présence d'éléments de traitements (des balises wsi:Claim, wsi="http://ws-i.org/schemas/conformanceClaim/") dans les éléments wsdl:documentation alors que ces derniers, tels des commentaires, devraient être ignorés par les programmes. Les éléments wsdl:documentation fournissent en effet des informations à usage de celui qui écrit le programme qui va consommer le service web.