

# Google calendar et VB6

API Google & Visual basic 6

par bbil ([bbil.developpez.com](http://bbil.developpez.com))

Date de publication : 16/12/2009

Dernière mise à jour :

Grâce aux « Google API » il est possible d'accéder à tout un tas de services Google, dans cet article nous allons voir comment utiliser cette API dans Visual Basic 6.0.

---

I - Introduction.....	3
I-A - Pré-requis.....	3
I-B - Principe général.....	3
II - Descriptif.....	4
II-A - Authentification.....	4
II-B - Envoi d'une commande.....	6
II-C - Créer un calendrier.....	7
II-C-1 - Conversion vers et de UTF-8.....	8
II-D - Liste des calendriers.....	10
II-E - Liste des événements.....	12
II-E-1 - Mise en forme date.....	13
II-F - Afficher un événement.....	14
II-G - Ajouter un événement.....	16
III - L'application exemple à télécharger.....	16
IV - Conclusion.....	18
IV-A - Pour aller plus loin.....	18
IV-B - Remerciements.....	18
IV-C - Vos commentaires.....	18

## I - Introduction

### I-A - Pré-requis

Le pré-requis semble évident, il vous faut un compte Google, si vous n'en possédez pas, rendez-vous ici :

### I-B - Principe général

Pour cet article, j'ai choisi d'utiliser un contrôle **INET** (Microsoft Internet Transfer Control 6.0) pour communiquer avec les serveurs Google.

Après préparation, les requêtes à destination du serveur Google sont «expédiées» grâce à la méthode « Execute » du contrôle Inet.

Le retour du serveur est «récupéré» suite au déclenchement de l'événement *Inet1\_StateChanged (ByVal State As Integer)* lorsque le paramètre State prend la valeur 12 ( *icResponseCompleted =12* ).

Lors de l'apparition de cet événement, la réponse du serveur est stockée dans la chaîne « *stRetHeader* » les données associées sont stockées dans la variable globale « *stReponse* » et un drapeau « *bFinAttente* » est levé.

```
'
' Gestion des réponses du serveur.
'
'
'
Private Sub Inet1_StateChanged(ByVal State As Integer)
' Récupère la réponse du serveur à l'aide de la
' méthode GetChunk lorsque State = 12.
    Select Case State
        ' ... Les autres cas ne sont pas décrits ici.
        Case icRequestSent '6 , sur envoi requête Raz des variables string en retour
            stReponse = ""
            stRetHeader = ""
        Case icError      ' 11

'MsgBox "Une erreur s'est produite lors de la communication avec l'ordinateur hôte", vbCritical, "Erreur"
            stReponse = "#-Erreur-Hote-BadAuthentication#"
            bFinAttente = True
        Case icResponseCompleted '12
            Dim vtData As Variant      ' Variable Data.
            Dim strData As String: strData = ""
            Dim bDone As Boolean: bDone = False
            vtData = Inet1.GetChunk(1024, icString)
            stRetHeader = Inet1.GetHeader()
            DoEvents
            Do While Not bDone
                strData = strData & vtData
                DoEvents
                ' Lecture du segment suivant.
                vtData = Inet1.GetChunk(1024, icString)
                If Len(vtData) = 0 Then
                    bDone = True
                End If
            Loop
            stReponse = strData
            bFinAttente = True
        End Select
    End Sub
```

## II - Descriptif

### II-A - Authentification

Toute action vers les calendriers passe par une authentification auprès des serveurs Google. Dans cet exemple, nous utiliserons la méthode nommée **ClientLogin**. Celle-ci demande d'utiliser le couple d'utilisateur (adresse mail Google, avec le @gmail.com) et le mot de passe de votre compte Google.

Une requête de type « POST » vers l'adresse : <https://www.google.com/accounts/ClientLogin>, va nous permettre d'obtenir une « clef » d'authentification qui sera ensuite utilisée pour les accès aux calendriers.

La requête devra, en outre, être composée des paramètres suivants :

Paramètres	Description
Email	Adresse mail de connexion
Passwd	Mot de passe associé
Source	Identification de l'application émettrice de la forme <i>NomEntreprise-NomApplication-Version</i>
Service	Service Google, pour l'agenda : <b>c1</b>

Dans ce projet, la fonction **GooAuthentification** du module **mdlGooCal** se charge de la mise en forme et de l'envoi de cette requête.

#### Authentification

```

' Authentification
'
Function GooAuthentification(stEmail As String, stPassword As String) As Boolean
    Dim stURL As String
    Dim stSource As String
    Dim stFormData As String
    Dim stHeaders As String
    stSource = App.CompanyName & "-" & App.EXENAME & "-" & App.Major & "." & App.Minor
    stURL = "https://www.google.com/accounts/ClientLogin"
    stFormData = "Email=" & stEmail & "&Passwd=" & stPassword & "&source=" & stSource & "&service=c1"
    stHeaders = "Content-Type:application/x-www-form-urlencoded"
    Inet1.Execute stURL, "POST", stFormData, stHeaders
    'Mise à 0 flag d'attente réponse
    bFinAttente = False
    Do Until bFinAttente = True
        DoEvents
    Loop
    If InStr(1, stRetHeader, "200 OK") > 0 Then
        stAuthCode = Right(stReponse, Len(stReponse) - InStrRev(stReponse, "Auth=") - 4)
        GooAuthentification = True
    Else
        GooAuthentification = False
    End If
End Function

```

La requête est validée si le statut de la réponse est « 200 OK », dans ce cas la réponse est composée de 3 codes alphanumériques SID, LSID, Auth.

Le seul code qui nous intéresse, pour la suite, est le code d'Authentification, placé après la chaîne de caractères « Auth = », celui-ci est extrait de la chaîne stReponse grâce aux fonctions de manipulation de chaînes Right, InstRev et Len .

Exemple de chaîne en retour (les valeurs sont tronquées) :

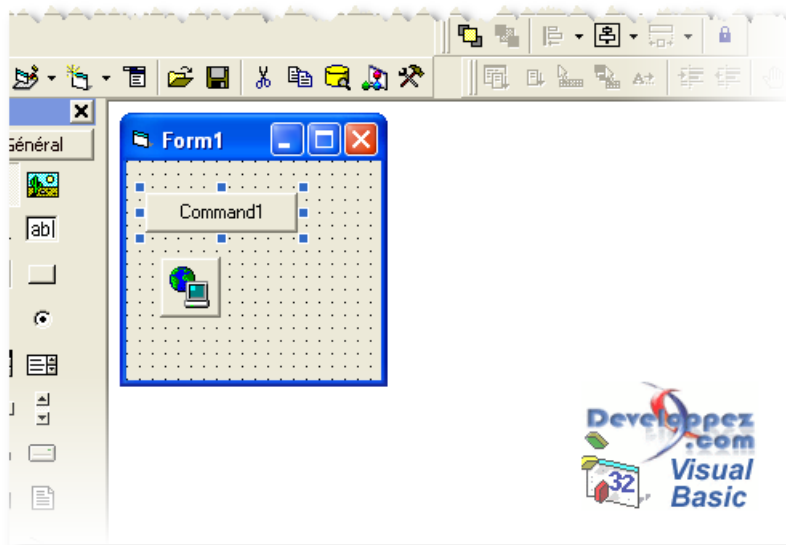
```
SID=dAXAAAI0AdAXDy8GWeSri6DdKGvYkCfo46XNqEc7W7rHapnQPWG
LSID=DdAXAAI8AbP1ACA5Hudu4pjSxOKBp8dAXGcCBE7Cj68W1JbP1EX
Auth=dAXGcl8AAbP1CA5Hudu4pjSxOKBp8dAXGcCBE7Cj68W1JbP1EXK_
```

Pour tester cette cela, créez une FORM, placez-y un contrôle INET les codes *Inet1\_StateChanged* et *GooAuthentication* vu précédemment puis rajoutez un bouton **Command1** pour la connexion et le code suivant :

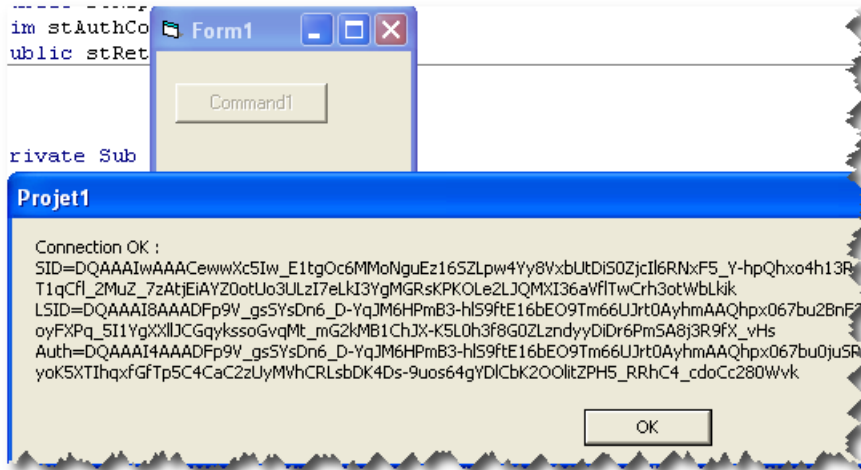
```
Option Explicit
Public bFinAttente As Boolean
Public stReponse As String
Dim stAuthCode As String
Public stRetHeader As String

Private Sub Command1_Click()
    Command1.Enabled = False
    If GooAuthentication("MonCompte@gmail.com", "MotDepasse") Then
        MsgBox "Connection OK : " & vbCrLf & stReponse
        Command1.Caption = "Connecté"
    Else
        MsgBox "Erreur connexion : " & vbCrLf & stReponse
        Command1.Caption = "Connection"
    End If
    Command1.Enabled = True
End Sub
```

Ce qui nous donne :



et après action sur le bouton de connexion :



## II-B - Envoi d'une commande

Une fois le code d'authentification récupéré, l'envoi de requêtes vers les serveurs Google peut être authentifié en plaçant ce code dans l'entête des requête POST ou GET.  
 La fonction **GooEnvoi** permet d'effectuer la mise en forme et l'envoi de cette requête.

```

Procédure d'envoi des requêtes authentifiées
'
' Post d'un message avec code d'identification.
'
Function GooEnvoi(stCde As String, stURL As String, Optional stFormData As String = "") As Boolean
    Dim stHeaders As String

    stHeaders = "Authorization: GoogleLogin auth=" & stAuthCode & _
                "Content-Type:application/atom+xml"

    bFinAttente = False
    Inet1.Execute stURL, stCde, stFormData, stHeaders
    Do Until bFinAttente = True
        DoEvents
    Loop
    GooEnvoi = Mid(stRetHeader, 10, 2) = "20" ' ("HTTP/1.1 201 Created" ou "HTTP/1.1 200 OK" )

End Function
    
```

En retour, les serveurs Google renvoient, en cas de succès, « HTTP/1.1 201 Created... » ou « HTTP/1.1 200 OK... » ou en cas de problème un statut commençant par « HTTP/1.1 40... »

### Rajoutez le code de cette fonction à votre form, avec en sus 3 contrôles TextBox :

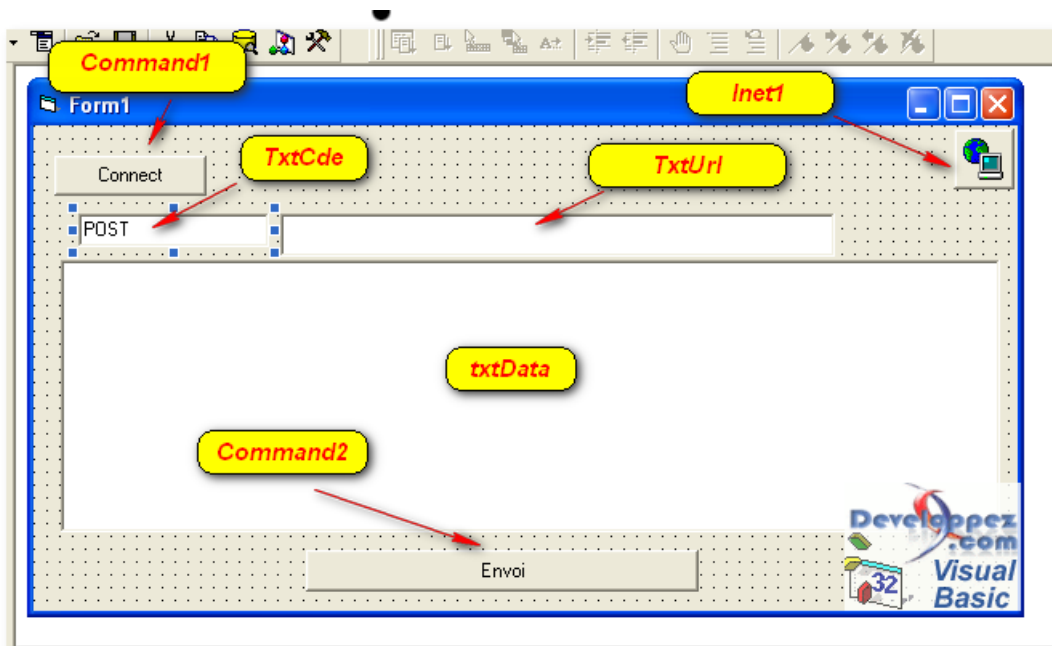
- **txtCde** : pour accueillir le type de commande à envoyer au serveur ( POST, GET)
- **txtUrl** : pour accueillir l'Url de destination
- **txtData** : pour accueillir les données à transmettre.

Un bouton **Command2** permet l'appel de la fonction GooEnvoi :

```

Private Sub Command2_Click()
    If GooEnvoi(txtCde, txtUrl, txtDATA) Then
        MsgBox "Envoi OK :" & vbCrLf & stReponse
    Else
        MsgBox "Réponse NOK :" & vbCrLf & stReponse
    End If
End Sub
    
```

Notre Form se présente alors ainsi :



## II-C - Créer un calendrier

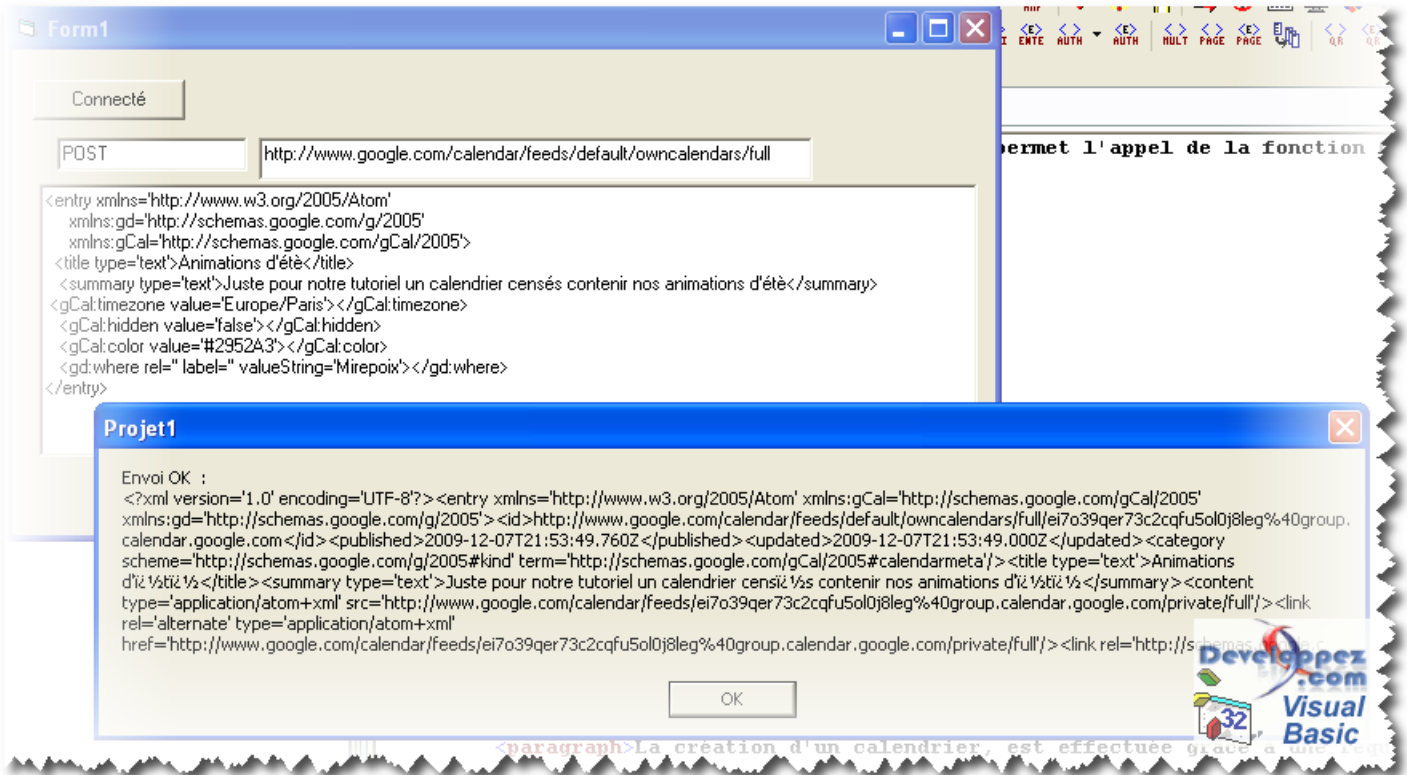
La création d'un calendrier s'effectue grâce à une requête POST vers

<http://www.google.com/calendar/feeds/default/owncalendars/full> ,  
les données utiles à la création étant fournies grâce à un code XML.

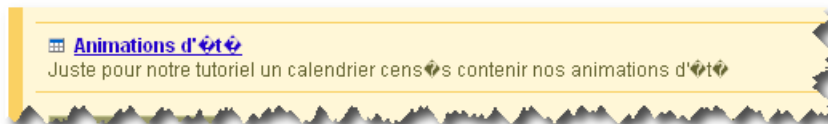
Exemple de code XML utilisé :

```
xml
<entry xmlns='http://www.w3.org/2005/Atom'
  xmlns:gd='http://schemas.google.com/g/2005'
  xmlns:gCal='http://schemas.google.com/gCal/2005'>
  <title type='text'>Animations d'été</title>
  <summary type='text'>Juste pour notre tutoriel un calendrier censés contenir nos animations
d'été</summary>
  <gCal:timezone value='Europe/Paris'></gCal:timezone>
  <gCal:hidden value='false'></gCal:hidden>
  <gCal:color value='#2952A3'></gCal:color>
  <gd:where rel='' label='' valueString='Mirepoix'></gd:where>
</entry>
```

On utilise notre petit programme pour tester ce code, après une action sur le bouton de connexion, les 3 TextBox sont renseignés. Une action sur le bouton Envoi provoque l'appel de GooEnvoi et en retour nous avons :



Il ne nous reste plus alors qu'à vérifier la création sur GoogleCalendar



C'est ici que l'on s'aperçoit d'un problème pour les caractères accentués, les serveurs Google attendant un codage UTF-8 par opposition au codage ANSI utilisé par les chaînes de caractères en VB6.

## II-C-1 - Conversion vers et de UTF-8

Pour gérer ce problème de codage, rajoutez au projet un module dans lequel vous placez le code suivant :

```

Option Explicit
Private Const CP_UTF8 = 65001
Private Declare Function MultiByteToWideChar Lib "kernel32" ( _
ByVal CodePage As Long, ByVal dwFlags As Long, _
ByVal lpMultiByteStr As Long, _
ByVal cchMultiByte As Long, _
ByVal lpWideCharStr As Long, _
ByVal cchWideChar As Long) As Long

Private Declare Function WideCharToMultiByte Lib "kernel32" ( _
ByVal CodePage As Long, _
ByVal dwFlags As Long, _
ByVal lpWideCharStr As Long, _
ByVal cchWideChar As Long, _
ByVal lpMultiByteStr As Long, _
ByVal cchMultiByte As Long, _
ByVal lpDefaultChar As Long, _

```

```

ByVal lpUsedDefaultChar As Long) As Long

Public Function EncodeUtf8(ByVal stA As String) As String
    Dim stBuf As String
    Dim lLg As Long
    If stA = "" Then
        EncodeUtf8 = ""
    Else
        'Récupère la longueur de la chaîne à convertir
        lLg = WideCharToMultiByte(CP_UTF8, 0, StrPtr(stA), -1, 0, 0, 0, 0)
        'Prépare buffer destiner à recevoir la chaîne
        stBuf = Space$(lLg)
        lLg = WideCharToMultiByte(CP_UTF8, 0, StrPtr(stA), -1, StrPtr(stBuf), lLg, 0, 0)
        stBuf = StrConv(stBuf, vbUnicode)
        EncodeUtf8 = Left$(stBuf, lLg - 1)
    End If
End Function

Public Function DecodeUtf8(ByVal stUtf8 As String) As String

    Dim stBuf As String
    Dim lLg As Long
    Dim i As Integer
    'Traitement des caractères accentués ou autres codés sous la forme % suivie du code Ascii
    ' url ou autre...
    For i = 1 To Len(stUtf8)
        If Mid(stUtf8, i, 1) = "%" Then
            Mid(stUtf8, i, 1) = Chr(Val("&H" & Mid(stUtf8, i + 1, 2)))
            Mid(stUtf8, i + 1, 2) = Chr(255) & Chr(255)
        End If
    Next

    stUtf8 = StrConv(stUtf8, vbFromUnicode)

    'Détermine le nombre de caractères de la chaîne à convertir
    lLg = MultiByteToWideChar(CP_UTF8, 0, StrPtr(stUtf8), -1, 0, 0)
    stBuf = Space$(lLg)
    'Conversion
    MultiByteToWideChar CP_UTF8, 0, StrPtr(stUtf8), -1, StrPtr(stBuf), lLg
    'Tronque chaîne suivant sa longueur
    DecodeUtf8 = Left$(stBuf, lLg - 1)
End Function

```

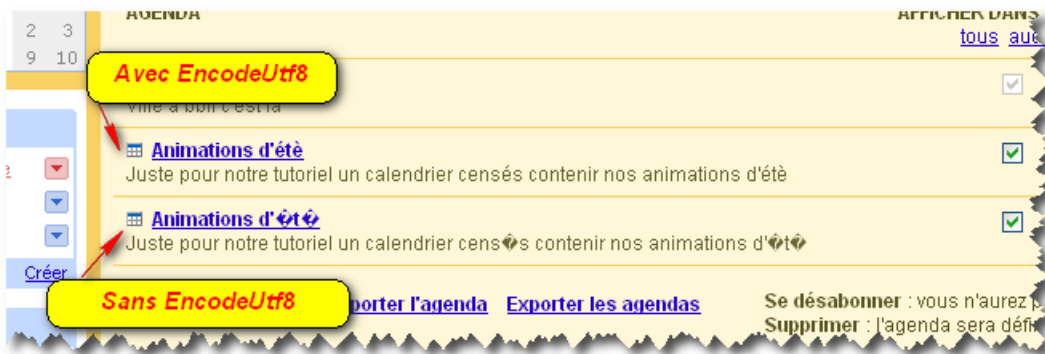
Ainsi nous disposons de deux nouvelles fonctions [EncodeUtf8](#) et [DecodeUtf8](#), utilisons alors la fonction [EncodeUtf8](#) dans notre code précédant de création d'Agenda :

```

Private Sub Command2_Click()
    If GooEnvoi(txtCde, txtUrl, EncodeUtf8(txtDATA)) Then
        MsgBox "Envoi OK :" & vbCrLf & stReponse
    Else
        MsgBox "Réponse NOK :" & vbCrLf & stReponse
    End If
End Sub

```

après exécution le résultat est visible dans GoogleAgenda :



## II-D - Liste des calendriers

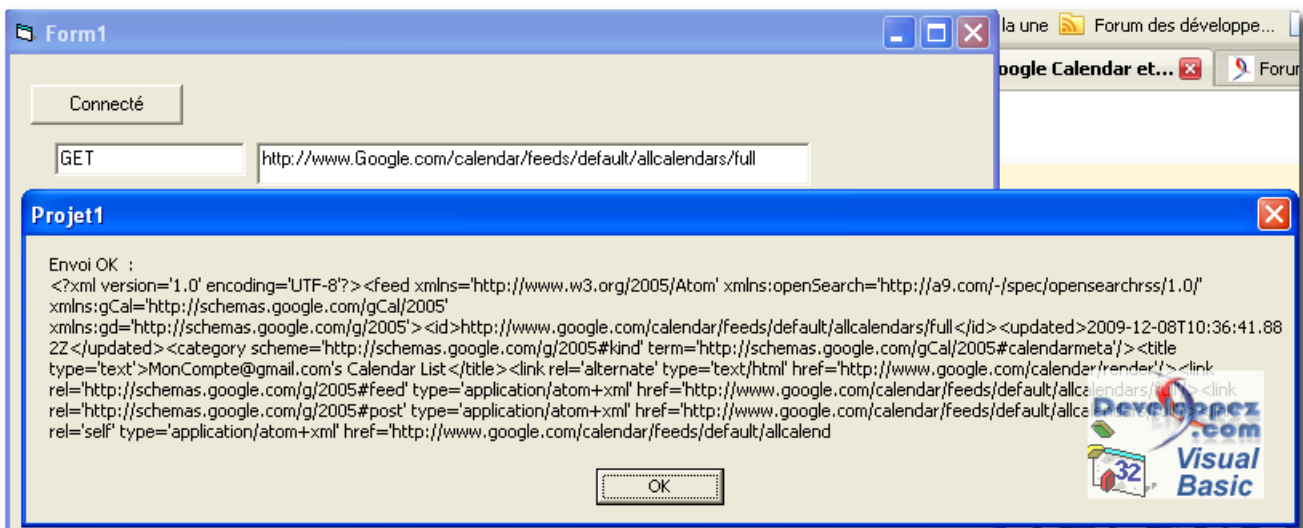
Il existe 3 types de calendrier Google :

- 1 **Calendrier primaire** : c'est celui qui est automatiquement créé lors de l'ouverture d'un compte Google Agenda.
- 2 **Calendriers secondaires** : ce sont tous les autres calendriers créés par l'utilisateur du compte Google.
- 3 **Calendriers importés** : ce sont les calendriers créés et partagés par d'autres utilisateurs Google, auxquels notre compte Google est abonné.

Deux requêtes « GET authentifiée » permettent de récupérer cette liste de calendriers :

- 1 <http://www.Google.com/calendar/feeds/default/allcalendars/full> : pour tous les calendriers (primaire, secondaire et importés)
- 2 <http://www.Google.com/calendar/feeds/default/owncalendars/full> : pour tous les calendriers primaires et secondaires ainsi que les calendriers importés dont les droits d'accès sont positionnés à « owner » (propriétaire).

En réponse à cette interrogation, la requête doit avoir, en retour, un statut « HTTP/1.1 200 OK... » et pour les données un XML renvoie la liste des calendriers.



On retrouve le même problème de codage, la fonction DecodeUtf8 permet de retrouver le bon codage dans l'application :

\_\_\_\_\_

```
stReponse = DecodeUtf8 (stReponse)
```

### Retour XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:openSearch="http://a9.com/-/spec/
opensearchrss/1.0/" xmlns:gCal="http://schemas.google.com/gCal/2005" xmlns:gd="http://
schemas.google.com/g/2005">
  <id>http://www.google.com/calendar/feeds/default/allcalendars/full</id>
  <updated>2009-12-08T11:22:23.416Z</updated>
  <category scheme="http://schemas.google.com/g/2005#kind" term="http://schemas.google.com/
gCal/2005#calendarmeta" />
  <title type="text">MonCompte@gmail.com's Calendar List</title>
  <link rel="alternate" type="text/html" href="http://www.google.com/calendar/render" />
  <link rel="http://schemas.google.com/g/2005#feed" type="application/atom+xml" href="http://
www.google.com/calendar/feeds/default/allcalendars/full" />
  <link rel="http://schemas.google.com/g/2005#post" type="application/atom+xml" href="http://
www.google.com/calendar/feeds/default/allcalendars/full" />
  <link rel="self" type="application/atom+xml" href="http://www.google.com/calendar/feeds/default/
allcalendars/full" />
  <author>
    <name>MonCompte@gmail.com</name>
    <email>MonCompte@gmail.com</email>
  </author>
  <generator version="1.0" uri="http://www.google.com/calendar">Google Calendar</generator>
  <openSearch:startIndex>1</openSearch:startIndex>
  <entry>
    <id>http://www.google.com/calendar/feeds/default/allcalendars/full/fgbpki21fdgfu3pimhq9ate0tc
%40group.calendar.google.com</id>
    <published>2009-12-08T11:22:23.410Z</published>
    <updated>2009-12-08T07:53:29.000Z</updated>
    <category scheme="http://schemas.google.com/g/2005#kind" term="http://schemas.google.com/
gCal/2005#calendarmeta" />
    <title type="text">Animations d'été</title>
    <summary type="text">Juste pour notre tutoriel un calendrier censés contenir nos animations
d'été</summary>
    <content type="application/atom+xml" src="http://www.google.com/calendar/feeds/
fgbpki21fdgfu3pimdq9ate0tc%40group.calendar.google.com/private/full" />
    <link rel="alternate" type="application/atom+xml" href="http://www.google.com/calendar/feeds/
fgbpki21fdgfudq3pimqq9ate0tc%40group.calendar.google.com/private/full" />
    <link rel="http://schemas.google.com/gCal/2005#eventFeed" type="application/atom+xml" href="http://
www.google.com/calendar/feeds/fgbpki21fdgfu3pimhq9ate0tc%40group.calendar.google.com/private/full" />
    <link rel="http://schemas.google.com/acl/2007#accessControlList" type="application/
atom+xml" href="http://www.google.com/calendar/feeds/fgbpki21fdgfu3pimhq9ate0tc
%40group.calendar.google.com/acl/full" />
    <link rel="self" type="application/atom+xml" href="http://www.google.com/calendar/feeds/default/
allcalendars/full/fgbpki21fdgfu3pimsqh9ate0tc%40group.calendar.google.com" />
    <link rel="edit" type="application/atom+xml" href="http://www.google.com/calendar/feeds/default/
allcalendars/full/fgbpki21qsdgfu3pimhq9ate0tc%40group.calendar.google.com" />
    <author>
      <name>Animations d'été</name>
    </author>
    <gCal:accesslevel value="owner" />
    <gCal:color value="#2952A3" />
    <gCal:hidden value="false" />
    <gd:where valueString="Mirepoix" />
    <gCal:selected value="false" />
    <gCal:timezone value="Europe/Paris" />
    <gCal:timesCleaned value="0" />
  </entry>
</feed>
```

L'XML retourné contient, tout d'abord, des informations générales sur le compte gmail, ensuite pour chaque calendrier une entrée : <entry> dans laquelle on peut retrouver les informations du calendrier.

Après avoir ajouté au projet, la référence à "Microsoft XML", (6.0 par exemple) et la déclaration de la variable `xmlCals`, tout en haut de votre module, afin que celle-ci soit accessible à toutes les fonctions :

```
Dim xmlCals As MSXML2.IXMLDOMNodeList ' Liste des agendas.
```

```
Private Sub cdAfflst_Click()
    AfficheListe True
End Sub
```

La fonction suivante permet la mise à jour de cette liste :

#### Extraction liste des calendriers

```
Sub AfficheListe (bProprietaire As Boolean)
    Dim xmlEntry As IXMLDOMElement
    Dim xmlReponse As New MSXML2.DOMDocument ' Réponse du serveur google.

    Dim stUrl As String
    lstCals.Clear
    stUrl = IIf(bProprietaire, "http://www.google.com/calendar/feeds/default/owncalendars/full" _
, "http://www.google.com/calendar/feeds/default/allcalendars/full")
    If GooEnvoi("GET", stUrl) Then
        If xmlReponse.loadXML(stReponse) Then
            Set xmlCals = xmlReponse.selectNodes("/feed/entry")
            For Each xmlEntry In xmlReponse.selectNodes("/feed/entry")
                lstCals.AddItem DecodeUtf8(xmlEntry.selectSingleNode("title").Text)
            Next
        End If
    End If
End Sub
```

Sur la "form" un nouveau bouton **cdLAfflst** et une listbox **lstCals**



## II-E - Liste des évènements

Pour extraire la liste des évènements, on utilise le lien trouvé dans la liste précédente des calendriers.

Sous la balise <content>

```
<entry>
(...)
<content type="application/atom+xml" src="http://www.google.com/calendar/feeds/MonMail1%40gmail.com/private/full" />
(□)
```

Lors d'un clic sur l'un des éléments de la liste des calendriers, une requête est exécutée afin d'en récupérer les évènements :

```
Private Sub LstCals_Click()
    Dim stUrl As String
    stUrl = xmlCals(LstCals.ListIndex).selectSingleNode("content").Attributes.getNamedItem("src").Text
    ListeEvents stUrl
End Sub
```

La procédure [ListeEvents](#) se charge d'envoyer la requête au serveur Google et ensuite de la mise en forme des données reçues:

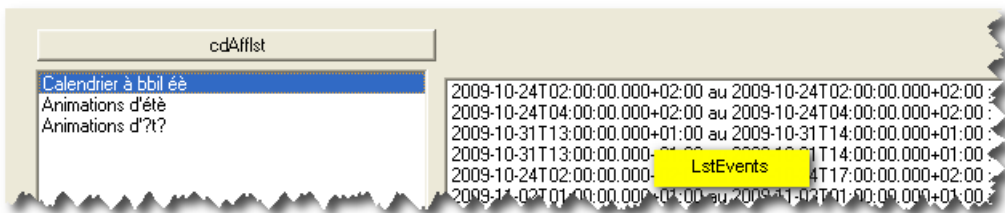
```

Lister les évènements
'
' Affiche la liste des événements du calendrier sélectionné

Private Sub ListeEvents(stUrl As String)
    Dim xmlDocEvents As New MSXML2.DOMDocument
    Dim xmlEntry As IXMLDOMElement
    Dim xmlEvents As MSXML2.IXMLDOMNodeList
    Dim DateDebut As String
    Dim DateFin As String
    Dim stTitre As String
    LstEvents.Clear
    If GooEnvoi("GET", stUrl) Then
        LstEvents.Clear
        If xmlDocEvents.loadXML(stReponse) Then
            Set xmlEvents = xmlDocEvents.selectNodes("/feed/entry")
            For Each xmlEntry In xmlEvents
                DateDebut =
                (xmlEntry.selectSingleNode("gd:when").Attributes.getNamedItem("startTime").Text)
                DateFin =
                (xmlEntry.selectSingleNode("gd:when").Attributes.getNamedItem("endTime").Text)
                LstEvents.AddItem DateDebut & " au " & DateFin & " : " & stTitre
            Next
        End If
    End If
End Sub

```

sur la form une listBox : [LstEvents](#) est renseignée avec date de début et fin de chacun des éléments;



## II-E-1 - Mise en forme date

Une fonction permet de convertir le format de date utilisé au format Date de VB

```

' Conversion Date/heure
' ' Récupère la date au format xs:DateTime et la convertit..
'
Function xsDateTime(st As String) As Date
    Dim t As Date
    If Len(st) > 12 Then
        t = TimeValue(Mid(st, 12, 8))
    End If
    xsDateTime = DateSerial(Left(st, 4), Mid(st, 6, 2), Mid(st, 9, 2)) + t
End Function

```

Utilisez cette fonction dans la procédure ListeEvents précédente en modifiant les lignes :

```

(..)
    Dim DateDebut As Date
    Dim DateFin As Date
    (..)

```

```

DateDebut =
xsDateTime(xmlEntry.selectSingleNode("gd:when").Attributes.getNamedItem("startTime").Text)
DateFin =
xsDateTime(xmlEntry.selectSingleNode("gd:when").Attributes.getNamedItem("endTime").Text)
(..)

```

## II-F - Afficher un événement

L'Xml reçu, en retour, comporte, tout d'abord, une partie « commune » reprenant les paramètres du calendrier :

xml

```

<?xml version="1.0" encoding="UTF-8" ?>
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:openSearch="http://a9.com/-/spec/opensearchrss/1.0/"
xmlns:gml="http://www.opengis.net/gml" xmlns:georss="http://www.georss.org/georss"
xmlns:batch="http://schemas.google.com/gdata/batch"
xmlns:gCal="http://schemas.google.com/gCal/2005" xmlns:gd="http://schemas.google.com/g/2005">
  <id>http://www.google.com/calendar/feeds/MonMaill%40gmail.com/private/full</id>
  <updated>2009-10-29T16:20:03.000Z</updated>
  <category scheme="http://schemas.google.com/g/2005#kind" term="http://schemas.google.com/g/2005#event" />
  <title type="text">Calendrier à bbil éé</title>
  <subtitle type="text">Ville à bbil c'est la</subtitle>
  <link rel="alternate" type="text/html" href="http://www.google.com/calendar/embed?src=MonMaill@gmail.com" />
  <link rel="http://schemas.google.com/g/2005#feed" type="application/atom+xml" href="http://www.google.com/calendar/feeds/MonMaill%40gmail.com/private/full" />
  <link rel="http://schemas.google.com/g/2005#post" type="application/atom+xml" href="http://www.google.com/calendar/feeds/MonMaill%40gmail.com/private/full" />
  <link rel="http://schemas.google.com/g/2005#batch" type="application/atom+xml" href="http://www.google.com/calendar/feeds/MonMaill%40gmail.com/private/full/batch" />
  <link rel="self" type="application/atom+xml" href="http://www.google.com/calendar/feeds/MonMaill%40gmail.com/private/full?max-results=25" />
  <author>
    <name>MonNom</name>
    <email>MonMaill@gmail.com</email>
  </author>
  <generator version="1.0" uri="http://www.google.com/calendar">Google Calendar</generator>
  <openSearch:totalResults>10</openSearch:totalResults>
  <openSearch:startIndex>1</openSearch:startIndex>
  <openSearch:itemsPerPage>25</openSearch:itemsPerPage>
  <gd:where valueString="VilleAabil" />
  <gCal:timezone value="Europe/Paris" />
  <gCal:timesCleaned value="0" />
  <entry>

```

Puis, pour chaque événement, une balise <entry> reprend les propriétés particulières à l'événement.

xml

```

<entry>
  <id>http://www.google.com/calendar/feeds/MonMaill%40gmail.com/private/full/9U28jc2kphrv65s63ukh7ogdes</id>
  <published>2009-10-24T19:33:04.000Z</published>
  <updated>2009-10-24T19:33:04.000Z</updated>
  <category scheme="http://schemas.google.com/g/2005#kind" term="http://schemas.google.com/g/2005#event" />
  <title type="text">mon evenement</title>
  <content type="text">Text3</content>
  <link rel="alternate" type="text/html" href="http://www.google.com/calendar/event?eid=NmI0OGU2NpjMmtwaHJvZnNjN1a2g3b2dkZXMgYmJpbC5iaWxAbQ" title="alternate" />
  <link rel="self" type="application/atom+xml" href="http://www.google.com/calendar/feeds/MonMaill%40gmail.com/private/full/9U28jc2kp5s63ukhhrv67ogdes" />
  <link rel="edit" type="application/atom+xml" href="http://www.google.com/calendar/feeds/MonMaill%40gmail.com/private/full/9U2rv65s63ukh7ogd8jc2kphes/63392095984" />

```

xml

```

<author>
<name>MonNom</name>
<email>MonMaill@gmail.com</email>
</author>
<gCal:guestsCanModify value="false" />
<gCal:guestsCanInviteOthers value="true" />
<gCal:guestsCanSeeGuests value="true" />
<gCal:anyoneCanAddSelf value="false" />
<gd:comments>
<gd:feedLink href="http://www.google.com/calendar/feeds/MonMaill%40gmail.com/private/
full/6bc24kphrr63ukh7ogdes/comments" />
</gd:comments>
<gd:eventStatus value="http://schemas.google.com/g/2005#event.confirmed" />
<gd:visibility value="http://schemas.google.com/g/2005#event.default" />
<gd:transparency value="http://schemas.google.com/g/2005#event.opaque" />
<gCal:uid value="9U28jcp3u2v65s63u2kkh7ogdes@google.com" />
<gCal:sequence value="0" />
<gd:when startTime="2009-10-24T02:00:00.000+02:00" endTime="2009-10-24T17:00:00.000+02:00" />
<gd:who rel="http://schemas.google.com/
g/2005#event.organizer" valueString="bbil" email="MonMaill@gmail.com" />
<gd:where valueString="lol" />
</entry>

```

Le suivant code permet ensuite de mettre à jour la zone événement sélectionné :

Détails de l'événement

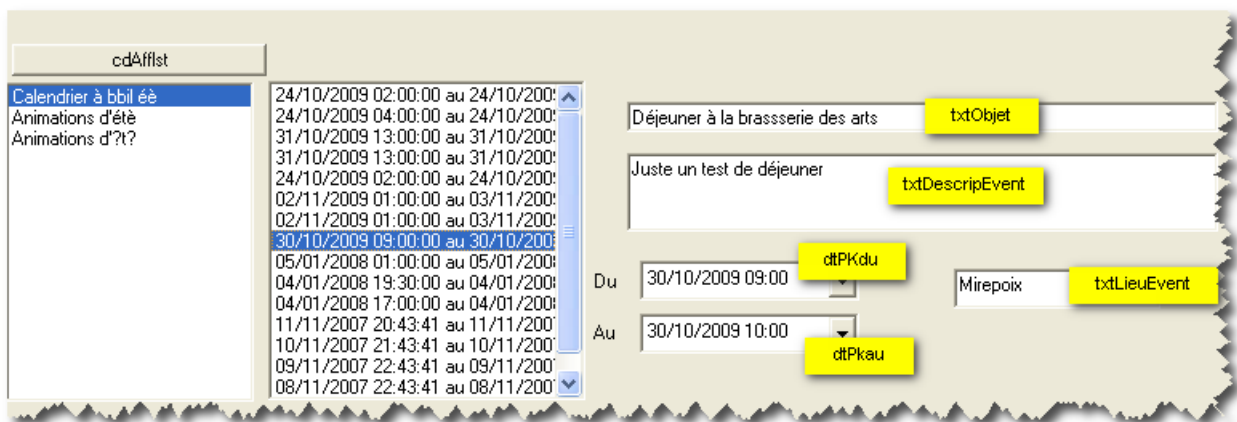
```

'
'   Clic sur la liste d'évènements, mise à jour champ "évènements sélectionnés"
'
Private Sub LstEvents_Click()
Dim xmlEntry As IXMLDOMElement
Set xmlEntry = xmlEvents.Item(LstEvents.ListIndex)
txtObjet = DecodeUtf8(xmlEntry.selectSingleNode("title").Text)
txtDescripEvent = DecodeUtf8(xmlEntry.selectSingleNode("content").Text)
txtLieuEvent =
DecodeUtf8(xmlEntry.selectSingleNode("gd:where").Attributes.getNamedItem("valueString").Text)
dtPKdu = xsDateTime(xmlEntry.selectSingleNode("gd:when").Attributes.getNamedItem("startTime").Text)
dtPkau = xsDateTime(xmlEntry.selectSingleNode("gd:when").Attributes.getNamedItem("endTime").Text)
End Sub

```

### Sur la Form, 5 nouveaux contrôles :

- TextBox : **txtObjet**, **txtObjet**, **txtDescripEvent**, **txtLieuEvent**
- DTPicker ("Microsoft Windows Common Controls") : **dtPKdu** , **dtPkau**



## II-G - Ajouter un évènement

Pour l'ajout d'un évènement, une requête, de type POST, est envoyée à l'url précédemment récupérée avec l'attribut « src » de la balise <content>, la requête est accompagnée des données nécessaires à la création de l'évènement.

Le code VB suivant permet de générer cette requête :

### Ajout d'un évènement dans le calendrier sélectionné

```

'-----
' Ajout d'un évènement
'
Private Sub cdAjEvent_Click()
    Dim stFormData As String
    Dim stURL As String

    Dim stDeb As String
    Dim stFin As String
    stDeb = Format(dtPKdu.Value, "yyyy-mm-ddTHH:MM:00.000Z")
    stFin = Format(DTPkau.Value, "yyyy-mm-ddTHH:MM:00.000Z")

    stURL = xmlCals(lstCals.ListIndex).selectSingleNode("content").Attributes.getNamedItem("src").Text
    stFormData = "<entry xmlns='http://www.w3.org/2005/Atom' & _
    " xmlns:gd='http://schemas.google.com/g/2005'>" & _
    " <category scheme='http://schemas.google.com/g/2005#kind' & _
    " term='http://schemas.google.com/g/2005#event'></category>" & _
    " <title type='text'>" & EncodeUtf8(TxtObjet) & "</title>" & _
    " <content type='text'>" & EncodeUtf8(txtDescripEvent) & "</content>" & _
    " <gd:transparency" & _
    " value='http://schemas.google.com/g/2005#event.opaque'>" & _
    " </gd:transparency>" & _
    " <gd:eventStatus" & _
    " value='http://schemas.google.com/g/2005#event.confirmed'>" & _
    " </gd:eventStatus>" & _
    " <gd:where valueString='" & EncodeUtf8(TxtLieuEvent) & "'></gd:where>" & _
    " <gd:when startTime='" & stDeb & "' & _
    " endTime='" & stFin & "'></gd:when>" & _
    " </entry>"
    If GooEnvoi("POST", stURL, stFormData) Then
        MsgBox "Ajout ok"
    Else
        MsgBox "Erreur "
    End If
End Sub

```

Exemple de données accompagnant la requête POST :

```

xml
<entry xmlns='http://www.w3.org/2005/Atom' xmlns:gd='http://schemas.google.com/g/2005'>
<category scheme='http://schemas.google.com/g/2005#kind' term='http://schemas.google.com/
g/2005#event'></category>
<title type='text'>Une nouvelle actualité </title>
<content type='text'>Juste pour notre test le commentaire associé </content>
<gd:transparency value='http://schemas.google.com/g/2005#event.opaque' </gd:transparency>
<gd:eventStatus value='http://schemas.google.com/g/2005#event.confirmed' </gd:eventStatus>
<gd:where valueString='Paris'></gd:where>
<gd:when startTime='2009-10-24T02:00:00.000Z' endTime='2009-10-24T02:00:00.000Z'></gd:when> </entry>

```

## III - L'application exemple à télécharger

Vous retrouverez, dans l'application suivante, tout le code décrit au paragraphe précédent.

bbil@gmail.com

n

Liste calendriers (Selectionnez en un)

Tous les calendriers

Tous Nos calendriers

Calendrier à bbil èè  
Cal2  
vb Redaction-developpez.com

Créer

onné

ndrier à bbil

le de calendrier

Evènements du calendrier sélectionné

24/10/2009 04:00:00 au 24/10/2009 04:00:00 :  
02/11/2009 01:00:00 au 03/11/2009 01:00:00 :  
02/11/2009 01:00:00 au 03/11/2009 01:00:00 :  
31/10/2009 13:00:00 au 31/10/2009 14:00:00 :  
31/10/2009 13:00:00 au 31/10/2009 14:00:00 :  
30/10/2009 09:00:00 au 30/10/2009 10:00:00 :  
24/10/2009 02:00:00 au 24/10/2009 17:00:00 :  
24/10/2009 02:00:00 au 24/10/2009 02:00:00 :  
05/01/2008 01:00:00 au 05/01/2008 01:00:00 :  
04/01/2008 19:30:00 au 04/01/2008 20:30:00 :  
04/01/2008 17:00:00 au 04/01/2008 18:00:00 :  
11/11/2007 20:43:41 au 11/11/2007 21:43:41 :

ctionné /ou/ à créer

Encore un évènement

Et voilà

/2009 01:00

Lieu

Paris

Ajoute Event

/2009 01:00

 **Les sources VB6 du projet GooCalendar****IV - Conclusion****IV-A - Pour aller plus loin**

Cet article ne constitue qu'une base pour la manipulation des calendriers Google, pour aller plus loin, rendez-vous sur :

 [http://code.google.com/intl/fr/apis/calendar/data/2.0/developers\\_guide\\_protocol.html](http://code.google.com/intl/fr/apis/calendar/data/2.0/developers_guide_protocol.html)

**IV-B - Remerciements**

Tous mes remerciements à **OhMonBato** et **Hédhili Jaïdane** pour leur relecture.

**IV-C - Vos commentaires**

Pour vous permettre de commenter cet article et ce code, une discussion est ouverte :