

ASP(Classic)

- 1. , ,
- 2. HWP, Word, Excel

1. , ,



 TABS Upload5 . ASP

import.asp

```
<%@ CodePage=65001 Language=VBScript %>
<% Option Explicit %>
<%
    On Error Resume Next
    Dim UPLOAD_PATH
    UPLOAD_PATH = "C:\inetpub\wwwroot\upload"

    Dim Upload, fileName
    '
    Set Upload = Server.CreateObject("TABSUpload4.Upload")
    Upload.CodePage = 65001

    '
    Upload.Start "C:\TEMP"
    Upload.Save UPLOAD_PATH, False

    ' ( )
    fileName = Upload.Form("file").ShortSaveName

    ' json
    Response.ContentType = "application/json"
    Response.write("{\"uploadPath\":\"/upload/\" & fileName & \"\"}")
%>
```

2. HWP, Word, Excel



 TABS Upload5 . ASP

import.asp

```
<%@ CodePage=65001 Language=VBScript %>
<% Option Explicit %>
```

```

<%
On Error Resume Next
Dim CONVERTER, FONTS, WORK, UPLOAD_PATH
CONVERTER = "C:\workspace\seimporter\sedocConverter\sedocConverter.exe"
FONTS = "C:\workspace\seimporter\fonts"
WORK = "C:\workspace\seimporter\tmp"
UPLOAD_PATH = "C:\inetpub\wwwroot\upload"

Dim filePath, outputPath, uuid, relativeOutputPath

Dim Upload
' .
Set Upload = Server.CreateObject("TABSEUpload4.Upload")
Upload.CodePage = 65001

' .
Upload.Start "C:\TEMP"
Upload.Save UPLOAD_PATH, False

'
filePath = Upload.Form("file").SaveName

'UUID (unique path)
uuid = CreateGUID()

outputPath = "C:\inetpub\wwwroot\output\" & uuid
relativeOutputPath = "/output/" & uuid

'
Dim wshShell, strCmd, result
'Set wshShell = CreateObject( "WScript.Shell" )
strCmd = CONVERTER & " -pz -f " & FONTS & " "" " & filePath & "" " & outputPath & " " & WORK
'result = wshShell.Run(strCmd, 0, true)
result = Exec(strCmd, 1)

If Not result = 0 Then
    Response.write "Error : " & result
Else
    '
    DeleteExistFile(filePath)
End If

Set wshShell = nothing
Set Upload = Nothing

'document.pb serialized
Dim binText
binText = ReadBinaryFile(outputPath & "\" & "document.pb")

'pb
DeleteExistFile(outputPath & "\" & "document.pb")

' json
Response.ContentType = "application/json"
Response.write("{\"importPath\":\"" & relativeOutputPath & "\", \"serializedData\":\"" & binText & "\"}")

Function Exec(c, t)
    Dim s, e : Set s = CreateObject("WScript.Shell") : Set e = s.Exec(c)
    Do While e.Status = 0
        Call s.Run("waitfor /t 1 OneSecond", 0, True)
        t = t - 1
        If 0 >= t Then
            Call s.Run("taskkill /t /f /pid " & e.ProcessId, 0, True)
            Exit Do
        End If
    Loop
    Set Exec = e
End Function

```

```

'UUID
Function CreateGUID()
    Dim tmpTemp
    tmpTemp = Right(String(4,48) & Year(Now()),4)
    tmpTemp = tmpTemp & Right(String(4,48) & Month(Now()),2)
    tmpTemp = tmpTemp & Right(String(4,48) & Day(Now()),2)
    tmpTemp = tmpTemp & Right(String(4,48) & Hour(Now()),2)
    tmpTemp = tmpTemp & Right(String(4,48) & Minute(Now()),2)
    tmpTemp = tmpTemp & Right(String(4,48) & Second(Now()),2)
    CreateGUID = tmpTemp
End Function

Function DeleteExistFile(filePath)
    Dim fso, result
    Set fso = CreateObject("Scripting.FileSystemObject")
    If fso.FileExists(filePath) Then
        fso.DeleteFile(filePath) ' .
        result = 1
    Else
        result = 0
    End If
    DeleteExistFile = result
End Function

Function ReadBinaryFile(FileName)
    Const adTypeBinary = 1
    Const adTypeText = 2

    'Create Stream object
    Dim BinaryStream
    Set BinaryStream = CreateObject("ADODB.Stream")
    Dim bin, str, cnt

    'Specify stream type - we want To get binary data.
    BinaryStream.Type = adTypeBinary

    'Open the stream
    BinaryStream.Open

    'Load the file data from disk To stream object
    BinaryStream.Position = 0 'Set the stream position to the start
    BinaryStream.LoadFromFile FileName

    cnt = 1
    Do
        bin = BinaryStream.Read(1024)

        if Not IsNull(bin) then
            if cnt = 1 then
                str = str & BinaryToString(bin, 17)
            Else
                str = str & "," & BinaryToString(bin, 1)
            end if
            cnt = cnt + 1
        end if
    Loop While Not IsNull(bin)

    ReadBinaryFile = "[" & str & "]"
    BinaryStream.Close
    Set BinaryStream = Nothing
End Function

Function BinaryToString(Binary, startPosition)
    'Antonin Foller, http://www.motobit.com
    'Optimized version of a simple BinaryToString algorithm.

    Dim cl1, cl2, cl3, pl1, pl2, pl3
    Dim L
    cl1 = startPosition '1 or 17
    cl2 = 1

```

```

c13 = 1
L = LenB(Binary)

Do While c11<=L
    p13 = p13 & CStr(AscB(MidB(Binary,c11,1)))
    If c11 < L Then
        p13 = p13 & ","
    End if
    c11 = c11 + 1
    c13 = c13 + 1
    If c13>300 Then
        p12 = p12 & p13
        p13 = ""
        c13 = 1
        c12 = c12 + 1
        If c12>200 Then
            p11 = p11 & p12
            p12 = ""
            c12 = 1
        End If
    End If
Loop
BinaryToString = p11 & p12 & p13
End Function
%>

```

-
- [Java Spring Framework](#)
 - [Java Servlet](#)
 - [ASP.NET \(C#\)](#)
 - [ASP\(Classic\)](#)
 - [PHP](#)
 - [PHP4](#)
 - [Django](#)
 - [Ruby On Rails](#)
 - [Wordpress plugin](#)