

Έκδοση Εγγράφου: 1.3

Ημερομηνία Τελευταίας Ενημέρωσης: 14/06/2016

(Υποστήριξη TLS1.2, προαπαιτούμενο framework 4.5 ή μεγαλύτερο)

## Upload zip αρχείων

### Στοιχεία Σύνδεσης

Η κλήση του REST Service για το upload zip αρχείων προϋποθέτει

1. Method HTTP POST στο Url (<https://www1.gsis.gr/myf/oltp/api/post/file>)
2. Ιδιότητες του Request
  - a. ContentType : multipart/form-data;
  - b. Authorization Header με Basic Authentication σε Base64String με username και password τα στοιχεία του Token που έχουν εκδοθεί
3. Ιδιότητες αρχείου
  - a. Content-Disposition: form-data; name="file"; filename="*filename*", όπου *filename*: το όνομα του αρχείου
  - b. Content-Type: application/octet-stream

### Επιστρεφόμενη Τιμή

Το αποτέλεσμα της επιτυχημένης κλήσης είναι string xml της μορφής

```
<result>
  <id>To ID του πακέτου</id>
  <status>OK | PROCESSING | FAIL</status>
  <errors>
    <error>Error Message 1</error>
    <error> Error Message 2</error>
    ....
  </errors>
</result>
```

## Παράδειγμα κλήσης

Στη συνέχεια παρατίθεται κώδικας C# για κλήση του REST service και ανέβασμα του .zip αρχείου

```
//Libraries
/*
using System;
using System.Net;
using System.IO;
using System.Xml.Linq;
*/

//Tls 1.2
System.Net.ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12; //update 1.3

//Custom parameters
string url = "REST_SERVICE_URL";
string sAuthorization = "USERNAME:PASSWORD";
string fileName = "FILENAME.zip";
string filePath = @"FILE_PATH" + fileName;

//Identificate separator
string boundary = "-----" + DateTime.Now.Ticks.ToString("x");
//Encoding
byte[] boundarybytes = System.Text.Encoding.ASCII.GetBytes("\r\n--" + boundary + "\r\n");

//Creation and specification of the request
HttpRequest wr = (HttpRequest)WebRequest.Create(url);
wr.ContentType = "multipart/form-data; boundary=" + boundary;
wr.Method = "POST";
wr.KeepAlive = true;
wr.Credentials = System.Net.CredentialCache.DefaultCredentials;

//Authentication
byte[] toEncodeAsBytes = System.Text.ASCIIEncoding.ASCII.GetBytes(sAuthorization);
```

```

string returnValue = System.Convert.ToBase64String(toEncodeAsBytes);
wr.Headers.Add("Authorization: Basic " + returnValue);

//Writing of the file
Stream rs = wr.GetRequestStream();
rs.Write(boundarybytes, 0, boundarybytes.Length);
byte[] formitembytes = System.Text.Encoding.UTF8.GetBytes(filePath);
rs.Write(formitembytes, 0, formitembytes.Length);
rs.Write(boundarybytes, 0, boundarybytes.Length);

string headerTemplate = "Content-Disposition: form-data; name=\"{0}\";filename=\"{1}\"\\r\\nContent-Type: {2}\\r\\n\\r\\n";
string header = string.Format(headerTemplate, "file", fileName, "application/octet-stream");
byte[] headerbytes = System.Text.Encoding.UTF8.GetBytes(header);
rs.Write(headerbytes, 0, headerbytes.Length);

FileStream fileStream = new FileStream(filePath, FileMode.Open, FileAccess.Read);
byte[] buffer = new byte[4096];
int bytesRead = 0;
while ((bytesRead = fileStream.Read(buffer, 0, buffer.Length)) != 0)
{
    rs.Write(buffer, 0, bytesRead);
}
fileStream.Close();

byte[] trailer = System.Text.Encoding.ASCII.GetBytes("\\r\\n--" + boundary + "--\\r\\n");
rs.Write(trailer, 0, trailer.Length);
rs.Close();
rs = null;

WebResponse wresp = null;
try
{
    //Get the response
    wresp = wr.GetResponse();
    Stream stream2 = wresp.GetResponseStream();
    StreamReader reader2 = new StreamReader(stream2);
    string responseData = reader2.ReadToEnd();
    //Parse xml
    XDocument xdoc = XDocument.Parse(responseData);
}

```

```
    if (xdoc.Element("result").Element("status").Value == "OK")
    {
        Console.WriteLine("OK. ID: " + xdoc.Element("result").Element("id").Value);
    }
    else if (xdoc.Element("result").Element("status").Value == "PROCESSING")
    {
        Console.WriteLine("PROCESSING. ID: " + xdoc.Element("result").Element("id").Value);
    }
    else if (xdoc.Element("result").Element("status").Value == "FAIL")
    {
        Console.WriteLine("FAIL. ID: " + xdoc.Element("result").Element("id").Value + ". See errors.");
    }
}
catch (Exception ex)
{
    string s = ex.Message;
}
finally
{
    if (wresp != null)
    {
        wresp.Close();
        wresp = null;
    }
    wr = null;
}

Console.ReadLine();
```

## Κατάσταση Απεσταλμένων Αρχείων

### Στοιχεία Σύνδεσης

Η κλήση του REST Service για το upload zip αρχείων προϋποθέτει

1. Method HTTP GET στο Url (<https://www1.gsis.gr/myf/oltp/api/post/status/<ID του πακέτου>>)
  - a. Παράμετρος id που αντιστοιχεί στο ID του πακέτου που έχει επιστραφεί κατά την αποστολή του αρχείου όπως περιγράφηκε πιο πάνω
2. Ιδιότητες του Request
  - a. Authorization Header με Basic Authentication σε Base64String με username και password τα στοιχεία του Token που έχουν εκδοθεί

### Επιστρεφόμενη Τιμή

Το αποτέλεσμα της επιτυχημένης κλήσης είναι string xml της μορφής

```
<result>  
  <id>To ID του πακέτου</id>  
  <status>OK | PROCESSING | FAIL</status>  
  <errors>  
    <error>Error Message 1</error>  
    <error> Error Message 2</error>  
    ....  
  </errors>  
</result>
```

### Παράδειγμα κλήσης

Στη συνέχεια παρατίθεται κώδικας C# για κλήση του REST service

```

//Libraries
/*
using System;
using System.Net;
using System.IO;
using System.Xml.Linq;
*/

//Tls 1.2
System.Net.ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12; //update 1.3

//Custom parameters
string url = "REST_SERVICE_URL";
string sAuthorization = "USERNAME:PASSWORD";
string id = "ID_OF_UPLOAD";//ID from upload service: get it from xdoc.Element("result").Element("id").Value

//Creation and specification of the request
System.Net.HttpWebRequest wr = (HttpWebRequest)WebRequest.Create(url + id);
wr.Method = "GET";
wr.KeepAlive = true;
wr.Credentials = System.Net.CredentialCache.DefaultCredentials;

//Authentication
byte[] toEncodeAsBytes = System.Text.ASCIIEncoding.ASCII.GetBytes(sAuthorization);
string returnValue = System.Convert.ToBase64String(toEncodeAsBytes);
wr.Headers.Add("Authorization: Basic " + returnValue);

WebResponse wresp = null;
try
{
    //Get the response
    wresp = wr.GetResponse();
    Stream stream2 = wresp.GetResponseStream();
    StreamReader reader2 = new StreamReader(stream2);
    string responseData = reader2.ReadToEnd();
    //Parse xml
    XDocument xdoc = XDocument.Parse(responseData);
}

```

```
if (xdoc.Element("result").Element("status").Value == "OK")
{
    Console.WriteLine("OK. ID: " + xdoc.Element("result").Element("id").Value);
}
else if (xdoc.Element("result").Element("status").Value == "PROCESSING")
{
    Console.WriteLine("PROCESSING. ID: " + xdoc.Element("result").Element("id").Value);
}
else if (xdoc.Element("result").Element("status").Value == "FAIL")
{
    Console.WriteLine("FAIL. ID: " + xdoc.Element("result").Element("id").Value + ". See errors.");
}
}
catch (Exception ex)
{
    Console.WriteLine("Something went wrong: " + ex.Message);
}
finally
{
    if (wresp != null)
    {
        wresp.Close();
        wresp = null;
    }
    wr = null;
}

Console.ReadLine();
```