

C-DAC

NameMatcher WebAPI

C-DAC

Contents

1. Compare Two Strings	3
2. Compare Two Files by Path.....	5
3. Compare Two Files By Object	8

1. COMPARE TWO STRINGS

URL WITH REQUEST PARAMTERS:

`http://(SERVERNAME)/(APPLICATION NAME)/NameMatch/CompareStrings?FirstName=(FIRST STRING)&SecondName=(SECOND STRING)`

SAMPLE REQUEST

<http://localhost/NameMatcher/NameMatch/CompareStrings?FirstName=CDAC&SecondName=CDAC>

SAMPLE RESPONSE

100

WEBAPI METHOD

```
public int CompareStrings(string FirstName, string SecondName)
```

SAMPLE CODE

```
//jQuery
jQuery.ajax({
    type: "GET",
    dataType: "json",
    url:
    "http://localhost/NameMatcher/NameMatch/CompareStrings?FirstName=CDAC&SecondName=CDAC",
    async: false,
    context: document.body
}).success(function (data) {
    $('#Result').html("Score:"+data+"%");
});
```

```
//C# framework 4.5
private async void getscore()
{
    using (var client = new HttpClient())
    {
        string URIhttp://localhost/NameMatcher/NameMatch/CompareStrings?FirstName=CDAC&SecondName=CDAC,
        var response = await client.GetAsync(URI);
        string content = await response.Content.ReadAsStringAsync();
        label1.Text = "score:" + content.ToString();
    }
}
```

```
//C# Framework 4.0
using System.Net.Http;
using System.IO;

String serviceURL = "http://localhost/NameMatcher/NameMatch";
public String compareTwoString(String firstName, String secondName)
{
    string url = serviceURL + "/CompareStrings?FirstName=" + firstName +
"&SecondName=" + secondName;

    HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
    request.AutomaticDecompression = DecompressionMethods.GZip;

    using (HttpWebResponse response = (HttpWebResponse)request.GetResponse())
    using (Stream stream = response.GetResponseStream())
    using (StreamReader reader = new StreamReader(stream))
    {
        return reader.ReadToEnd();
    }
}
```

```
/Java

private String serviceURL = "http://localhost/NameMatcher/NameMatch";

public String compareTwoString(String firstName, String secondName) throws Exception
{
    String response = "";
    URL url = new URL(serviceURL + "/CompareStrings?FirstName=" + firstName
+ "&SecondName=" + secondName);
    URLConnection conn = url.openConnection();
    BufferedReader br = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
    String line = null;
    while((line = br.readLine()) != null){
        response = response + line;
    }
    return response;
}
```

2. COMPARE TWO FILES BY PATH

URL WITH REQUEST PARAMTERS:

`http:// (SERVERNAME) / (APPLICATION NAME) /NameMatch /CompareTwoFiles?Path1=(FIRST FILE PATH)&Path2=(SECOND FILE PATH)`

Note :

1. ServerName - Name of the server where webapi is hosted
2. "First File path" - Path of the File where the file is located on the server (First need to upload the file on the server or exists on the server)
3. "Second File Path" - Path of the File where the file is located on the server (First need to upload the file on the server or exists on the server)

WEBAPI METHOD

```
public List<object> CompareTwoFiles(string path1, string path2)
```

SAMPLE REQUEST

<http://localhost/NameMatcher/NameMatch/CompareTwoFiles?Path1=E:\\1.txt&SecondName=E:\\2.txt>

```
//C# framework 4.5
private async void CompareTwoFielByPath()
{
    using (var client = new HttpClient())
    {
        string URI = "http:// localhost
/NameMatcher/NameMatch/CompareTwoFiles?Path1=E:\\1.txt&Path2=E:\\2.txt";
        var response = await client.GetAsync(URI);
        if (response.IsSuccessStatusCode)
        {
            var FileByteArrayJsonString = await
response.Content.ReadAsStringAsync();
            // assign the result to the datagridview
            dataGridView1.DataSource =
JsonConvert.DeserializeObject<ResultTable[]>(FileByteArrayJsonString).ToList();
        }
    }
}
```

```
//C# framework 4.0
using Newtonsoft.Json;
using System.Net;
using System.IO;
using System.Runtime.Serialization.Formatters.Binary;
using System.Net.Http;
using System.Web.Script.Serialization;

private String serviceURL = "http://localhost/NameMatcher/NameMatch";

public void compareTwoFilesByPath(String firstFilePath, String secondFilePath)
{
    string url= (serviceURL + "/CompareTwoFiles?Path1=" + firstFilePath +
"&Path2=" + secondFilePath);
    HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
    request.AutomaticDecompression = DecompressionMethods.GZip;

    using (HttpWebResponse response = (HttpWebResponse)request.GetResponse())
    using (Stream stream = response.GetResponseStream())
    using (StreamReader reader = new StreamReader(stream))
    {
        result = reader.ReadToEnd();
        //show the result in the gridview
        var list = JsonConvert.DeserializeObject<ResultTable[]>(result).ToList();
        List<ResultTable>Res_List = list.Cast<ResultTable>().ToList();
        dataGridView1.DataSource = Res_List;
    }
}

//CLASS (used to wrapped the byte array and send using object)
[Serializable]
public class FileByteArray
{
    public byte[] FirstFileBytes { get; set; }
    public byte[] SecondFileBytes { get; set; }
}

//CLASS (used to show the result)
public class ResultTable
{
    public int Index { get; set; }
    public string RecordNo { get; set; }
    public string Name1 { get; set; }
    public string Name2 { get; set; }
    public string Score { get; set; }
}
```

```
//Java
private String serviceURL = "http://localhost/NameMatcher/NameMatch";

public String compareTwoFilesByPath(String firstFilePath, String secondFilePath)
throws Exception {
    String response = "";
    URL url = new URL(serviceURL + "/CompareTwoFiles?Path1=" +
firstFilePath + "&Path2=" + secondFilePath);
    URLConnection conn = url.openConnection();
    BufferedReader br = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
    String line = null;
    while((line = br.readLine()) != null) {
        response = response + line;
    }
    return response;
}
```

SAMPLE RESPONSE

```
[
{"Index":1,"Name1":"GOPALAKRISHNA R Y","Name2":"MR GOPALAKRISHNA R
Y","Score":"100"},
{"Index":2,"Name1":"MANJUNATH GANIGER","Name2":"MANJUNATH
GANIGER","Score":"100"},
{"Index":3,"Name1":"H T SHANTHAKUMAR","Name2":"H T SHANTHA KUMAR S/O
H.K.THAMMEGOWDA","Score":"98"},
{"Index":4,"Name1":"Hanumantaiah","Name2":"HANUMANTAIAH","Score":"100"},
. . .
]
```

3. COMPARE TWO FILES BY OBJECT

We Can compare the files by sending the file in byte array by object to the web api **method**

URL:

WEBAPI METHOD

```
public List<object> CompareTwoFilesByObject (FileByteArray obj)
```

SAMPLE CODE

```
//C# framework 4.5
```

CLASS (used to wrapped the byte array and send using object)

```
public class FileByteArray
{
    public byte[] FirstFileBytes { get; set; }
    public byte[] SecondFileBytes { get; set; }
}
```

CLASS (used to show the result)

```
public class ResultTable
{
    public int Index { get; set; }
    public string RecordNo { get; set; }
    public string Name1 { get; set; }
    public string Name2 { get; set; }
    public string Score { get; set; }
}
```

Function to call webapi and get the result

```
//URL FOR SERVICE
```

```
string URI = "http://localhost/NameMatcher/NameMatch/CompareTwoFilesByObject";
```

```
private async void SendFile()
{
    FileByteArray obj = new FileByteArray();
    obj.FirstFileBytes = File.ReadAllBytes("E:\Ln_En.txt");
    obj.SecondFileBytes = File.ReadAllBytes("E:\Ln_Hn.txt");
    using (var client = new HttpClient())
    {
        var serializedObj = JsonConvert.SerializeObject(obj);
        var content = new StringContent(serializedObj, Encoding.UTF8,
"application/json");
        var response = await client.PostAsync(URI, content);
        if (response.IsSuccessStatusCode)
        {
            var FileByteArrayJsonString = await
response.Content.ReadAsStringAsync();

```



```
        dataGridView1.DataSource =  
        JsonConvert.DeserializeObject<ResultTable[]>(FileByteArrayJsonString).ToList();  
    }  
  
    }  
  
}
```

```
//C# Framework 4.0  
  
using Newtonsoft.Json;  
using System.Net;  
using System.IO;  
using System.Runtime.Serialization.Formatters.Binary;  
using System.Net.Http;  
using System.Web.Script.Serialization;  
Function to call webapi and get the result  
Public void Comparefilebypath()  
{  
  
    FileByteArray obj = new FileByteArray();  
    obj.FirstFileBytes = File.ReadAllBytes("E:\Ln_En.txt");  
    obj.SecondFileBytes = File.ReadAllBytes("E:\Ln_Hn.txt");  
    JavaScriptSerializer serializer = new JavaScriptSerializer();  
    string output = serializer.Serialize(obj);  
    string strUri =  
    "http://localhost/NameMatcher/CompareTwoFilesByObject";  
    Uri uri = new Uri(strUri);  
    WebRequest request = WebRequest.Create(uri);  
    request.Method = "POST";  
    request.ContentType = "application/json; charset=utf-8";  
  
    JavaScriptSerializer jsonSerializer = new JavaScriptSerializer();  
    string serOut = jsonSerializer.Serialize(obj);  
  
    using (StreamWriter writer = new StreamWriter(request.GetRequestStream()))  
    {  
        writer.Write(serOut);  
    }  
  
    WebResponse response = request.GetResponse();  
    Stream reader = response.GetResponseStream();  
  
    StreamReader sReader = new StreamReader(reader);  
    string outResult = sReader.ReadToEnd();  
    sReader.Close();  
  
    //show the result in the gridview  
  
    var list = JsonConvert.DeserializeObject<ResultTable[]>(outResult).ToList();  
    List<ResultTable> Res_List = list.Cast<ResultTable>().ToList();  
    dataGridView1.DataSource = Res_List;  
  
}
```

CLASS (used to wrapped the byte array and send using object)

```
[Serializable]
public class FileByteArray
{
    public byte[] FirstFileBytes { get; set; }
    public byte[] SecondFileBytes { get; set; }
}
```

//CLASS (used to show the result)

```
public class ResultTable
{
    public int Index { get; set; }
    public string RecordNo { get; set; }
    public string Name1 { get; set; }
    public string Name2 { get; set; }
    public string Score { get; set; }
}
```

```
//Java
```

```
//URL FOR SERVICE
```

```
private String serviceURL = "http://10.208.11.138:80/NameMatcher/NameMatch";
```

```
//FileByteArray class
```

```
public class FileByteArray
{
    public byte[] FirstFileBytes;
    public byte[] SecondFileBytes;
    public byte[] getFirstFileBytes() {
        return FirstFileBytes;
    }
    public void setFirstFileBytes(byte[] firstFileBytes) {
        FirstFileBytes = firstFileBytes;
    }
    public byte[] getSecondFileBytes() {
        return SecondFileBytes;
    }
    public void setSecondFileBytes(byte[] secondFileBytes) {
        SecondFileBytes = secondFileBytes;
    }
}
```

```
//CLIENT API FOR CALLING SERVICE TO COMPARE TWO FILES
```

```
public String compareTwoFilesByObject(String firstFilePath, String secondFilePath)
throws Exception {
    String response = "";
```

```
//Creating URL and opening HttpURLConnection for post request
URL url = new URL(serviceURL + "/CompareTwoFilesByObject");
HttpURLConnection conn = (HttpURLConnection) url.openConnection();
conn.setRequestMethod("POST");
conn.setDoOutput(true);
conn.setRequestProperty("charset", "utf-8");
conn.setRequestProperty("Content-Type", "application/json");

//creating object of FileByteArray
//needs following dependency : apache commons-io to convert file
//into byte array
FileByteArray fileBytes = new FileByteArray();
fileBytes.setFirstFileBytes(IOUtils.toByteArray(new FileInputStream(new
File(firstFilePath))));
fileBytes.setSecondFileBytes(IOUtils.toByteArray(new FileInputStream(new
File(secondFilePath))));

//serializing fileBytes to JSON String
//needs following dependency : jackson-all to seriliazе object
//into json content
ObjectMapper objectMapper = new ObjectMapper();
String jsonContent = objectMapper.writeValueAsString(fileBytes);

//adding url parameters to output stream
DataOutputStream wr = new DataOutputStream(conn.getOutputStream());
wr.writeBytes(jsonContent);
wr.flush();
wr.close();

//reading response from service
BufferedReader br = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
String line = null;
while((line = br.readLine()) != null) {
    response = response + line;
}
return response;
}
```

SAMPLE RESPONSE

```
[  
  
{"Index":1,"Name1":"GOPALAKRISHNA R Y","Name2":"MR GOPALAKRISHNA R  
Y","Score":"100"},  
{"Index":2,"Name1":"MANJUNATH GANIGER","Name2":"MANJUNATH  
GANIGER","Score":"100"},  
{"Index":3,"Name1":"H T SHANTHAKUMAR","Name2":"H T SHANTHA KUMAR S/O  
H.K.THAMMEGOWDA","Score":"98"},  
{"Index":4,"Name1":"Hanumantaiiah","Name2":"HANUMANTAIAH","Score":"100"},  
.  
.  
.  
]
```

Note:

1. The 2 files together should not exceed the limit of 2GB, when using this method.
2. The files need not be present on the server