

- **COMPARE TWO STRINGS**

URL WITH REQUEST PARAMTERS:

http://(SERVERNAME)/(APPLICATION NAME)/NameMatch/CompareStrings?FirstName=(FIRST STRING)&SecondName=(SECOND STRING) & ApplicationID=(APPLICATIONID)&DepartmentCode=(DEPARTMENTCODE)&TransactionID=(TRANSACTIONID)

SAMPLE REQUEST

[...http://10.10.29.178/WebAPIGok/NameMatch/CompareStrings?FirstName=CDAC&SecondName=CDAC&ApplicationID=DMO001&DepartmentCode=DMO&TransactionID=DMO098034hdjjnvkvLA](http://10.10.29.178/WebAPIGok/NameMatch/CompareStrings?FirstName=CDAC&SecondName=CDAC&ApplicationID=DMO001&DepartmentCode=DMO&TransactionID=DMO098034hdjjnvkvLA)

SAMPLE RESPONSE

100

WEBAPI METHOD

```
public int CompareStrings(string ApplicationID, string DepartmentCode, string TransactionID, string FirstName, string SecondName)
```

SAMPLE CODE

```
//C#  
  
public async Task<JsonResult> CompareTwoString(string ApplicationID, string DepartmentCode, string TransactionID, string FirstName, string SecondName)  
{  
    try  
    {  
        HttpClientHandler handler = new HttpClientHandler();  
        handler.AutomaticDecompression = System.Net.DecompressionMethods.GZip;  
        var FileByteArrayJsonString = "";  
        using (var client = new HttpClient(handler))  
        {  
            client.Timeout = TimeSpan.FromMinutes(60);  
            string url = URI + "CompareStrings?ApplicationID=" + ApplicationID.Trim() + "&DepartmentCode=" + DepartmentCode.Trim() + "&TransactionID=" + TransactionID.Trim() + "&FirstName=" + FirstName + "&SecondName=" + SecondName;  
            var response = await client.GetAsync(url);  
            if (response.IsSuccessStatusCode)  
            {  
                FileByteArrayJsonString = await
```

```

response.Content.ReadAsStringAsync();
        }
        client.Dispose();
    }
    return Json(FileByteArrayJsonString, JsonRequestBehavior.AllowGet);
}
catch (Exception ex)
{

    throw;
}

}

```

JSON RESULT

1 (If Response has error message)

```

{
  "Score": -1,
  "IsSuccess": false,
  "ErrorMessage": "ApplicationID & Transaction ID not unique",
  "ApplicationID": "GOK001",
  "DepartmentCode": "GOK",
  "TransactionID": "GOK1234567890987654",
  "CallType": "CompareStrings"
}

```

2 (If Response has score without error)

```

{
  "Score": 100,
  "IsSuccess": true,
  "ErrorMessage": null,
  "ApplicationID": "GOK001",
  "DepartmentCode": "GOK",
  "TransactionID": "GOK1234654327652135",
  "CallType": "CompareStrings"
}

```

Note :

1. Application ID = 6 digit no. [NNNNNN] (DepartmentCode + ApplicationID)
2. Department Code = Digit/Character
3. Transaction ID = 19 Digit/Character
 [Application ID(6)+ DDMMYY(6)+SqenceNo(7)]

- **COMPARE TWO FILES BY PATH**

URL WITH REQUEST PARAMTERS:

[http:// \(SERVERNAME\) / \(APPLICATION NAME\) /NameMatch /CompareTwoFiles?Path1=\(FIRST FILE PATH\)&Path2=\(SECOND FILE PATH\) \) & ApplicationID=\(APPLICATIONID\)&DepartmentCode=\(DEPARTMENTCODE\)&TransactionID=\(TRANSACTIONID\)](http://(SERVERNAME)/(APPLICATION NAME)/NameMatch /CompareTwoFiles?Path1=(FIRST FILE PATH)&Path2=(SECOND FILE PATH) & ApplicationID=(APPLICATIONID)&DepartmentCode=(DEPARTMENTCODE)&TransactionID=(TRANSACTIONID))

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 ApplicationID, string DepartmentCode, string TransactionID, string path1, string path2)
```

SAMPLE REQUEST 112%%111121122112112112112112112112

<http://localhost/NameMatcher/NameMatch/CompareTwoFiles?Path1=E:\1.txt&SecondName=E:\2.txt&ApplicationID=GOK001&DepartmentCode=GOK&TransactionID= GOK0012802190000001>

SAMPLE RESPONSE

```
{
  "ScoreList": [{
    "Index": 1,
    "RecordNo": "1",
    "Name1": "anant rajaram kulkarni",
    "Name2": "□ □ □□□□□□□□",
    "Score": "19"
  },{..},..],
  "Response": {
    "Score": 0,
    "IsSuccess": true,
    "ErrorMessage": null,
    "ApplicationID": "GOK001",
    "DepartmentCode": "GOK",
    "TransactionID": "11111111111111111111",
    "CallType": "CompareTwoFilesByPath"
  }
}
```

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; }
    public string ApplicationID { get; set; }
    public string DepartmentCode { get; set; }
    public string TransactionID { 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; }
}
```

CLASS (used to show the result)

```
public class FileCompareScoreDetails
{
    public FileCompareScoreDetails()
    {
        ScoreList = new List<ResultTable>();
    }
    public List<ResultTable> ScoreList { get; set; }
    public ResponseInt Response { 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");
}
```

```

HttpClientHandler handler = new HttpClientHandler();
handler.AutomaticDecompression = System.Net.DecompressionMethods.GZip;
using (var client = new HttpClient(handler))
{
    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();
        FileCompareScoreDetails model =
JsonConvert.DeserializeObject<FileCompareScoreDetails>(FileByteArrayJsonString);
        dataGridView1.DataSource = model.ScoreList;
    }
}
}
}

```

SAMPLE RESPONSE

```

{
  "ScoreList": [{
    "Index": 1,
    "RecordNo": "1",
    "Name1": "anant rajaram kulkarni",
    "Name2": "□ □ □□□□□□□□",
    "Score": "19"
  },{..},..],
  "Response": {
    "Score": 0,
    "IsSuccess": true,
    "ErrorMessage": null,
    "ApplicationID": "GOK001",
    "DepartmentCode": "GOK",
    "TransactionID": "111111111111111111",
    "CallType": "CompareTwoFilesByPath"
  }
}

```

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