*CamAttendance Integration*

**Introduction**

CamAttendance offers adequate flexibility to integrate CamAttendance device to your HRMS/Attendance system. Every time an employee checks-in or checks-out, data gets stored onto a CamAttendance Cloud platform, which uses webhooks, APIs, and other various ways to communicate and transfer that data to your HRMS/Attendance system. Thus, integration enables your system to receive as well as push data to CamAttendance server.

**API\_Call\_SampleCode**

The sample code enables a user to exchange data over server and client PC through API. This sample code is written in C# that allows you to get data from CamAttendance cloud to your PC and post data from your PC to Cloud.

You can integrate this code to your HRMS for –

* Getting transaction log data for a specific time-period in JSON format
* Getting all employee details of a company
* Getting all details of a particular employee
* Adding a new employee to existing employee list

**API\_Functions Class**

This class contains properties you need to set property values in the calling event. Some functions of this class are described below -

**GetEmpInfo**(string base64EncodedStr)

This function has been used for getting the details of all employees or of a particular employee. The parameter used is Base64 encrypted string Of Account ID, secure Token and current timestamp. You will get data in Json format in jsonString variable. Now, you can parse/save this data as you wish.

**AddEmployeeExample(dynamic** jsonData**, string** base64EncodedStr**)**

This function has been used for adding a new employee along with the relevant information like department, designation, etc. You need to set the path to the property value as defined in this class and then this function helps get data from that text file and push this file data to server to add employee. Before that, you need to put employee detail in that file in JSON format. This function pushes this file data to server to add employee.

**GetTransactionLogExample** (string Base64EncodedToken)

This function has been used for getting transaction data for a specific date period. First you need to set respective property value as defined in this class. Then you need to set a range i.e. *from date* and *to date*. The parameter is Base64 encrypted string Of Account ID, secure Token and current timestamp. You will get data in Json format in text file at the mentioned location as you set in property. Now you can parse/save this data in DB or wherever you wish.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Net.Http.Headers;

using System.Security.Cryptography;

using System.Security.Cryptography.X509Certificates;

using System.Security.AccessControl;

using System.Security.Authentication;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Newtonsoft.Json;

using Newtonsoft.Json.Linq;

namespace API\_Call\_SampleCode

{

class API\_Functions

{

private string \_baseurl = "";

private DateTime \_DT\_from;

private DateTime \_DT\_To;

private string \_AccountId = "";

private string \_secureToken = "";

private string \_PathtoCreateDataFile = "";

private string \_JsonFilePathToAddEmp = "";

private string \_EmpId = "";

private string \_CompanyId = "";

public API\_Functions()

{

GetBaseUrl();

}

public string BaseUrl

{

get { return \_baseurl; }

set { \_baseurl = value; }

}

public DateTime DT\_from

{

get { return \_DT\_from; }

set { \_DT\_from = value; }

}

public DateTime DT\_To

{

get { return \_DT\_To; }

set { \_DT\_To = value; }

}

public string AccountId

{

get { return \_AccountId; }

set { \_AccountId = value; }

}

public string secureToken

{

get { return \_secureToken; }

set { \_secureToken = value; }

}

public string PathtoCreateDataFile

{

get { return \_PathtoCreateDataFile; }

set { \_PathtoCreateDataFile = value; }

}

public string JsonFilePathToAddEmp

{

get { return \_JsonFilePathToAddEmp; }

set { \_JsonFilePathToAddEmp = value; }

}

public string EmpId

{

get { return \_EmpId; }

set { \_EmpId = value; }

}

public string CompanyId

{

get { return \_CompanyId; }

set { \_CompanyId = value; }

}

public void GetEmpInfo(string base64EncodedStr)

{

HttpClientHandler handler = new HttpClientHandler();

HttpClient client = new HttpClient(handler);

string content = string.Empty;

if (EmpId != "") { EmpId = "/" + EmpId; }

string messageUri = BaseUrl + "employee" + EmpId + " ?companyId=" + CompanyId;

HttpWebRequest request = (HttpWebRequest)WebRequest.Create(messageUri);

var BodyPara = base64EncodedStr;

request.Headers.Add("token", base64EncodedStr);

request.ContentType = "application/json";

request.Method = "Get";

HttpWebResponse response = (HttpWebResponse)request.GetResponse();

Stream responseStream = response.GetResponseStream();

string jsonString = null;

using (StreamReader reader = new StreamReader(responseStream))

{

jsonString = reader.ReadToEnd();

Console.WriteLine();

reader.Close();

}

}

public void AddEmployeeExample(dynamic jsonData, string base64EncodedStr)

{

try

{

HttpClientHandler handler = new HttpClientHandler();

HttpClient client = new HttpClient(handler);

string messageUri = BaseUrl + "employee";

HttpWebRequest request = (HttpWebRequest)WebRequest.Create(messageUri);

var BodyPara = base64EncodedStr;

request.Headers.Add("token", base64EncodedStr);

request.ContentType = "application/json";

request.Method = "Post";

request.Accept = "application/json";

using (var streamWriter = new StreamWriter(request.GetRequestStream()))

{

streamWriter.Write(JsonConvert.DeserializeObject(jsonData));

}

HttpWebResponse response = (HttpWebResponse)request.GetResponse();

Stream responseStream = response.GetResponseStream();

string jsonString = null;

using (StreamReader reader = new StreamReader(responseStream))

{

jsonString = reader.ReadToEnd();

Console.WriteLine();

reader.Close();

}

}

catch (Exception ee) { }

}

public void GetTransactionLogExample(string Base64EncodedToken)

{

StringBuilder sb = new StringBuilder();

try

{

string lastEvaluatedKey = ""; string Size50 = "&size=10";

HttpClientHandler handler = new HttpClientHandler();

HttpClient client = new HttpClient(handler);

string content = string.Empty;

string DateFrom = GetTimeInMiliSec(DT\_from);

string date\_To = GetTimeInMiliSec(DT\_To);

Found:

string messageUri = BaseUrl + "transactions?from=" + DateFrom + "&to=" + date\_To + lastEvaluatedKey + Size50;

HttpWebRequest request = (HttpWebRequest)WebRequest.Create(messageUri);

var BodyPara = Base64EncodedToken;

request.Headers.Add("token", Base64EncodedToken);

request.ContentType = "application/json";

request.Method = "Get";

HttpWebResponse response = (HttpWebResponse)request.GetResponse();

Stream responseStream = response.GetResponseStream();

string jsonString = null;

using (StreamReader reader = new StreamReader(responseStream))

{

jsonString = reader.ReadToEnd();

reader.Close();

}

dynamic dynObj = JsonConvert.DeserializeObject(jsonString);

foreach (var Root in dynObj.Root)

{

foreach (var data in Root)

{

foreach (JProperty keyValue in data)

{

string KeyName = keyValue.Name.ToString();

var KeyVal = keyValue.Value;

if (KeyName == "list")

{

foreach (var data1 in KeyVal)

{

sb.Append("{");

foreach (JProperty keyValue2 in data1)

{

string KeyName2 = keyValue2.Name.ToString();

var KeyVal2 = keyValue2.Value;

sb.Append(KeyName2 + ":" + KeyVal2 + ",");

}

sb.Append("},\r\n");

}

}

if (KeyName == "lastEvaluatedKey")

{

if (KeyVal.ToString() != "")

{

lastEvaluatedKey = "";

lastEvaluatedKey = "&lastEvaluatedKey=" + KeyVal;

goto Found;

}

}

}

}

}

}

catch (Exception ex) { }

finally

{

CreateTextFile(sb);

}

}

public string EncryptString()

{

long currentTimeMillis = (long)DateTime.Now.ToUniversalTime().Subtract(new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc)).TotalMilliseconds;

string base64Decoded = secureToken + "\_" + currentTimeMillis + "\_" + AccountId;

string base64Encoded;

byte[] data = System.Text.ASCIIEncoding.ASCII.GetBytes(base64Decoded);

base64Encoded = System.Convert.ToBase64String(data);

return base64Encoded;

}

public string ReadDataFromTextFile()

{

string JsonData = string.Empty;

if (File.Exists(JsonFilePathToAddEmp))

{

JsonData = File.ReadAllText(JsonFilePathToAddEmp);

}

return JsonData;

}

public string GetTimeInMiliSec(DateTime Time)

{

long TimeInMiliSec = (long)Time.ToUniversalTime().Subtract(new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc)).TotalMilliseconds; ;

return TimeInMiliSec.ToString();

}

public void CreateTextFile(StringBuilder sb)

{

if (!File.Exists(PathtoCreateDataFile))

{

using (StreamWriter sw = File.CreateText(PathtoCreateDataFile))

{

sw.WriteLine(sb.ToString());

}

}

}

public void GetBaseUrl()

{

BaseUrl = File.ReadAllText(Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "Config.txt"));

}

}

}

**JSON File format for adding an employee –**

{

"companyId":"K05",

"email": "skapilq@gmail.com",

"empId": "1111111",

"firstName": "testemp",

"lastName": "test",

"phoneNumber": "9011121213",

"locationId": "1",

"designationId": "DES-001",

"departmentId": "Dep\_002",

"joiningDate":"2020-12-14",

"supervisior":"1001182"

}

You can call the **API\_Function Class Methods** at any event in code as you require. Here, we call at button event and set the property value in the constructor.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace API\_Call\_SampleCode

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

OBJApiInterface.AccountId = "";

OBJApiInterface.secureToken = "";

OBJApiInterface.EmpId = "";

OBJApiInterface.CompanyId = "";

OBJApiInterface.JsonFilePathToAddEmp = "";

OBJApiInterface.PathtoCreateDataFile = @"D://CamAtt.txt";

}

API\_Functions OBJApiInterface = new API\_Functions();

private void btn\_GetTransactionLog\_Click(object sender, EventArgs e)

{

OBJApiInterface.DT\_from = dateTimePicker1.Value;

OBJApiInterface.DT\_To = dateTimePicker2.Value;

string Base64EncodedToken = OBJApiInterface.EncryptString();

OBJApiInterface.GetTransactionLogExample(Base64EncodedToken);

}

private void btn\_AddEmployee\_Click(object sender, EventArgs e)

{

var JsonData = OBJApiInterface.ReadDataFromTextFile();

string Base64EncodedToken = OBJApiInterface.EncryptString();

OBJApiInterface.AddEmployeeExample(JsonData, Base64EncodedToken);

}

private void btn\_GetEmpInfo\_Click(object sender, EventArgs e)

{

string Base64EncodedToken = OBJApiInterface.EncryptString();

OBJApiInterface.GetEmpInfo(Base64EncodedToken);

}

}

}