

Integrating with Visma's Edge Api

Content

Overview.....	2
Swagger pages.....	2
Api url	2
Sign up for access.....	3
Authentication	4
Get data	5
Without parameters.....	5
With parameters	5
With more parameters	6
Get list of data when the result contains pageNumber and isComplete.....	7
Upload data	8
Entities used in examples	9

Overview

This is a simple how-to description for working with the Control Edge Api. All the examples are done with Visual Studio 2022 and net 8.0.

Swagger pages

Environment	Url
Prod	https://controlapi.control.visma.com/swagger
Stage	https://stage-controlapi.control.visma.com/swagger

Api url

Environment	Url
Prod	https://controlapi.control.visma.com
Stage	https://stage-controlapi.control.visma.com

Sign up for access

Sign up for Api access on <https://oauth.developers.visma.com/>

Authentication

Authentication is done by requesting an access token from Visma Connect.

Environment	Authentication url
Prod	https://connect.visma.com/connect/token
Test	https://connect.identity.stagaws.visma.com/connect/token

- The token is valid for one hour.

```
using Newtonsoft.Json;

string? url = "https://connect.identity.stagaws.visma.com/connect/token";
string? client_id = "your clientid";
string? client_secret = "your clientsecret";
string? TenantId = "your tenantid";

using var httpClient = new HttpClient();
httpClient.Timeout = new TimeSpan(0, 5, 0);

var data = new[]
{
    new KeyValuePair<string, string>("client_id", client_id),
    new KeyValuePair<string, string>("client_secret", client_secret),
    new KeyValuePair<string, string>("grant_type", "client_credentials"),
    new KeyValuePair<string, string>("tenant_id", TenantId)
};

var response = await httpClient.PostAsync(url, new FormUrlEncodedContent(data));
var respString = await response.Content.ReadAsStringAsync();
var token = JsonConvert.DeserializeObject<TokenResponse>(respString);

if (!string.IsNullOrEmpty(token.ToString()))
{
    TimeSpan t = TimeSpan.FromSeconds(Convert.ToDouble(token.Expires_in));
    string expires = string.Format("{0:D2}h:{1:D2}m:{2:D2}s:{3:D3}ms",
        t.Hours,
        t.Minutes,
        t.Seconds,
        t.Milliseconds);

    Console.WriteLine("Token: " + token.Access_token + Environment.NewLine + "Expires " + expires);
    Console.WriteLine("Use the token to call the API");
}
else
{
    Console.WriteLine("Token is null or empty" + Environment.NewLine + "Check your settings!");
}

Console.WriteLine("Press any key to exit...");
Console.ReadKey();

internal class TokenResponse
{
    public string? Access_token { get; set; }
    public string? Expires_in { get; set; }
}
```

Get data

Without parameters

```

////////////////////////////////////
/// Get data from the API without parameters
////////////////////////////////////
string? baseurl = "https://stage-controlapi.control.visma.com";
string? endpoint = "/api/account/get-accounts";
string? TenantId = "your tenantid";
string? StorageSpaceId = "your storageid";

HttpResponseMessage response;
string? _url = baseurl + endpoint;

using (var client = new HttpClient())
{
    client.DefaultRequestHeaders.Add("Authorization", "Bearer " + token);
    client.DefaultRequestHeaders.Add("Accept", "application/hal+json");
    client.DefaultRequestHeaders.Add("ContentType", "application/json");
    client.DefaultRequestHeaders.Add("TenantId", TenantId);
    client.DefaultRequestHeaders.Add("StorageSpaceId", StorageSpaceId);

    var requestUri = new Uri(_url);
    response = await client.GetAsync(requestUri);
    var responseMessage = response.Content.ReadAsStringAsync().Result;
    var result = JsonConvert.DeserializeObject<List<Account>>(responseMessage);

    Console.WriteLine("List of accounts");
    foreach (var item in result)
    {
        Console.WriteLine("Account: " + item.AccountNumber + " " + item.Name + " " + item.Created);
    }
}

```

With parameters

```

////////////////////////////////////
/// Get data from the API with parameters
////////////////////////////////////
string? baseurl = "https://stage-controlapi.control.visma.com";
string? endpoint = "/api/account/get-by-id?accountNumber=1012";
string? TenantId = "your tenantid";
string? StorageSpaceId = "your storageid";

string? _url = baseurl + endpoint;

using (var client = new HttpClient())
{
    client.DefaultRequestHeaders.Add("Authorization", "Bearer " + token);
    client.DefaultRequestHeaders.Add("Accept", "application/hal+json");
    client.DefaultRequestHeaders.Add("ContentType", "application/json");
    client.DefaultRequestHeaders.Add("TenantId", TenantId);
    client.DefaultRequestHeaders.Add("StorageSpaceId", StorageSpaceId);

    var requestUri = new Uri(_url);
    response = await client.GetAsync(requestUri);
    var responseMessage = response.Content.ReadAsStringAsync().Result;
    var result = JsonConvert.DeserializeObject<Account>(responseMessage);

    Console.WriteLine("Account 1012");
    Console.WriteLine("Account: " + result.AccountNumber + " " + result.Name + " " + result.Created);
}

```

With more parameters

```
////////////////////////////////////  
/// Get data from the API with 2 parameters  
////////////////////////////////////  
string? baseurl = "https://stage-controlapi.control.visma.com";  
string? endpoint = "/api/invoicelog/get-invoicelog-by-Id?invnumber=654321&invserie=F1";  
string? TenantId = "your tenantid";  
string? StorageSpaceId = "your storageid";  
  
string? _url = baseurl + endpoint;  
  
using (var client = new HttpClient())  
{  
    client.DefaultRequestHeaders.Add("Authorization", "Bearer " + token);  
    client.DefaultRequestHeaders.Add("Accept", "application/hal+json");  
    client.DefaultRequestHeaders.Add("ContentType", "application/json");  
    client.DefaultRequestHeaders.Add("TenantId", TenantId);  
    client.DefaultRequestHeaders.Add("StorageSpaceId", StorageSpaceId);  
  
    var requestUri = new Uri(_url);  
    response = await client.GetAsync(requestUri);  
    var responseMessage = response.Content.ReadAsStringAsync().Result;  
    var result = JsonConvert.DeserializeObject<Invlog>(responseMessage);  
  
    Console.WriteLine("Order : " + result.InvoiceNumber + ", Serie : " + result.Serie);  
  
    foreach (var item in result.Rows)  
    {  
        Console.WriteLine("Row: " + item.Freetext + " " + item.Account + " " + item.Quantity + " " + item.Amount);  
    }  
}
```

Get list of data when the result contains pageNumber and isComplete.

- Always start with page=1.
- Check the result. If isComplete equals false, then query the endpoint with page + 1 until isComplete equals true.
- The result will be of maximum 5000 rows.

```

////////////////////////////////////
/// Get list of data from the API if the response contains pageNumber and isComplete
/// Aslong as the isComplete flag is false, there are more pages to get
////////////////////////////////////
string? baseurl = "https://stage-controlapi.control.visma.com";
string? endpoint = "/api/person/get-customers?page=1";
string? TenantId = "your tenantid";
string? StorageSpaceId = "your storageid";

string? _url = baseurl + endpoint;

using (var client = new HttpClient())
{
    client.DefaultRequestHeaders.Add("Authorization", "Bearer " + token);
    client.DefaultRequestHeaders.Add("Accept", "application/hal+json");
    client.DefaultRequestHeaders.Add("ContentType", "application/json");
    client.DefaultRequestHeaders.Add("TenantId", TenantId);
    client.DefaultRequestHeaders.Add("StorageSpaceId", StorageSpaceId);

    var requestUri = new Uri(_url);
    response = await client.GetAsync(requestUri);
    var responseMessage = response.Content.ReadAsStringAsync().Result;
    var result = JsonConvert.DeserializeObject<Person>(responseMessage);

    Console.WriteLine("List of type person");
    Console.WriteLine("PageNumber: " + result.PageNumber + " IsComplete: " + result.IsComplete);

    foreach (var item in result.Items)
    {
        Console.WriteLine("Person: " + item.Type + " " + item.Id + " " + item.Name);
    }
}

```

Upload data

- When uploading data to the Api wait for the response before uploading the next new entry in the database.

```

////////////////////////////////////
/// Upload data to the API.
/// Create a voucherlog
////////////////////////////////////
string? baseurl = "https://stage-controlapi.control.visma.com";
string? endpoint = "/api/voucherlog/create-voucherlog";
string? TenantId = "your tenantid";
string? StorageSpaceId = "your storageid";

HttpResponseMessage response;
string? _url = baseurl + endpoint;

using (var client = new HttpClient())
{
    client.DefaultRequestHeaders.Add("Authorization", "Bearer " + token);
    client.DefaultRequestHeaders.Add("Accept", "application/hal+json");
    client.DefaultRequestHeaders.Add("ContentType", "application/json");
    client.DefaultRequestHeaders.Add("TenantId", TenantId);
    client.DefaultRequestHeaders.Add("StorageSpaceId", StorageSpaceId);

    var data = new Voulog
    {
        BusinessUnit = "AB1",
        Serie = "E01",
        VouNo = 0,
        Type = 5,
        VoucherDate = new DateTime(2024, 1, 26),
        FreeText = "AAAAA",
        TransferLog = true,
        DeleteAfterTransfer = false,
        Rows = new List<VouRowLog>
        {
            new VouRowLog
            {
                Account = "1710",
                Amount = -100,
                Date = new DateTime(2024, 1, 26),
                FreeText = "AAAAA",
                VatCode = 0
            },
            new VouRowLog
            {
                Account = "6530",
                Amount = 100,
                Date = new DateTime(2024, 1, 26),
                FreeText = "AAAAA",
                VatCode = 0
            }
        }
    };

    var jsonString = JsonConvert.SerializeObject(data, Formatting.Indented, new IsoDateTimeConverter { DateTimeFormat = "yyyy-MM-dd" });
    var content = new StringContent(jsonString, Encoding.UTF8, "application/json");
    var requestUri = new Uri(_url);
    response = await client.PostAsync(requestUri, content);
    var responseMessage = response.Content.ReadAsStringAsync().Result;
    var result = JsonConvert.DeserializeObject<VouLogResponse>(responseMessage);

    Console.WriteLine(result.LogNo);
    Console.WriteLine(result.LogDate);
    Console.WriteLine(result.VouNo);
    Console.WriteLine(result.Status);
    Console.WriteLine(result.LogNo);
    Console.ReadLine();
}

```


Entities used in examples

```

public class Account
{
    [JsonProperty("accountNumber")]
    public string? AccountNumber { get; set; }
    [JsonProperty("name")]
    public string? Name { get; set; }
    [JsonProperty("created")]
    public DateTime Created { get; set; }
}
public class Invlog
{
    [JsonProperty("businessUnit")]
    public string? BusinessUnit { get; set; }

    [JsonProperty("serie")]
    public string? Serie { get; set; }

    [JsonProperty("invoiceNumber")]
    public int InvoiceNumber { get; set; }

    [JsonProperty("yourOrderNo")]
    public string? YourOrderNo { get; set; }

    [JsonProperty("rows")]
    public List<InvRowLog>? Rows { get; set; }
}
public class InvRowLog
{
    [JsonProperty("freetext")]
    public string? Freetext { get; set; }

    [JsonProperty("account")]
    public string? Account { get; set; }

    [JsonProperty("quantity")]
    public decimal Quantity { get; set; }

    [JsonProperty("amount")]
    public decimal Amount { get; set; }
}
public class Person
{
    [JsonProperty("pageNumber")]
    public int PageNumber { get; set; }
    [JsonProperty("isComplete")]
    public bool IsComplete { get; set; }

    [JsonProperty("list")]
    public List<PersonItem> Items { get; set; } = new List<PersonItem>();
}
public class PersonItem
{
    [JsonProperty("type")]
    public int Type { get; set; }
    [JsonProperty("id")]
    public string? Id { get; set; }
    [JsonProperty("name")]
    public string Name { get; set; }
}

```

```

public class Voulog
{
    [JsonProperty("businessUnit")]
    public string? BusinessUnit { get; set; }
    [JsonProperty("serie")]
    public string? Serie { get; set; }
    [JsonProperty("vouNo")]
    public int? VouNo { get; set; }
    [JsonProperty("type")]
    public int Type { get; set; }
    [JsonProperty("voucherDate")]
    public DateTime? VoucherDate { get; set; }
    [JsonProperty("freeText")]
    public string? FreeText { get; set; }
    [JsonProperty("transferLog")]
    public bool? TransferLog { get; set; }
    [JsonProperty("deleteAfterTransfer")]
    public bool? DeleteAfterTransfer { get; set; }
    [JsonProperty("rows")]
    public List<VouRowLog>? Rows { get; set; }
}

public class VouRowLog
{
    [JsonProperty("account")]
    public string? Account { get; set; }
    [JsonProperty("amount")]
    public decimal? Amount { get; set; }
    [JsonProperty("date")]
    public DateTime? Date { get; set; }
    [JsonProperty("freeText")]
    public string? FreeText { get; set; }
    [JsonProperty("vatCode")]
    public int VatCode { get; set; }
}

public class VouLogResponse
{
    [JsonProperty("logno")]
    public int? LogNo { get; set; }
    [JsonProperty("logdate")]
    public DateTime? LogDate { get; set; }
    [JsonProperty("vouno")]
    public int? VouNo { get; set; }
    [JsonProperty("status")]
    public int? Status { get; set; }
}

```