



FEXCO Corporate Payments

API Code Samples

07.08.2019



Initial steps to use FCP API

1. The client must be setup as a Fexco Corporate Payments Client.
2. The client must then request a username and password for access to the Fexco Corporate Payments Web API.
3. The client then uses this username and password to access the Web API (using Token based authentication)

Use of the Web API is free to Fexco Corporate Payments Clients!

There are different levels of access to the API based on client requirements.

Login Details for Client (Sample of what is sent to Client)

Client	Fexco Tax Business (4429)
Email	johnDoe@fexco.com
Password	FexcoTaxBusienssTest123\$

API Technical/Code

The API uses the regular Authentication: Bearer <token> method.

The sample code (C#) below (Appendix 1) is used to retrieve a token using a user account.

The sample code (C#) below (Appendix 2) is used to retrieve specific Bank Country Details

The sample code(C#) below (Appendix 3) is used to retrieve Exchange Rate details

Appendix 1

```
private static async Task<string> GetAPIToken(string userName, string password, string apiBaseUri)
{
    try
    {
        using (var client = new HttpClient())
        {
            //setup client
            client.BaseAddress = new Uri(apiBaseUri);
            client.DefaultRequestHeaders.Accept.Clear();
            client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

            //setup login data
            var formContent = new FormUrlEncodedContent(new[]
            {
                new KeyValuePair<string, string>("grant_type", "password"),
                new KeyValuePair<string, string>("username", userName),
                new KeyValuePair<string, string>("password", password),
            });

            var result = await client.PostAsync("/Token", formContent)
                .ConfigureAwait(continueOnCapturedContext: false);

            HttpResponseMessage responseMessage = result;

            //get access token from response body
            var responseJson = await responseMessage.Content.ReadAsStringAsync();
            var jsonObject = JObject.Parse(responseJson);
            return jsonObject.GetValue("access_token").ToString();
        }
    }
    catch (Exception e)
    {
        throw e;
    }
}
```

Appendix 2

```
protected void btnGetBankCountryDetails_Click(object sender, EventArgs e)
{
    //Make the call
    Var response = GetRequest(this.txtToken.Text, this.txtBaseURI.Text, this.txtBankPath.Text,
"/GetBankCountryRules?bankCountry=250").Result;
    this.txtBankCountryDetails.Text = response;
}

static async Task<string> GetRequest(string token, string apiBaseUri, string requestPath, string methodDetails)
{
    using (var client = new HttpClient())
    {
        //setup client
        client.BaseAddress = new Uri(apiBaseUri);
        client.DefaultRequestHeaders.Accept.Clear();
        client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));
        client.DefaultRequestHeaders.Add("Authorization", "Bearer " + token);

        //make request
        var result = await client.GetAsync(requestPath + methodDetails)
            .ConfigureAwait(continueOnCapturedContext: false);

        if (methodDetails.Contains("PostBankDetails")) //need to return true/false if postbankdetails method i.e. ok
or not found
        {
            return result.ReasonPhrase.ToString();
        }
        else
        {
            HttpResponseMessage response = result;
            var responseString = await response.Content.ReadAsStringAsync();
            return responseString;
        }
    }
}
```

Appendix 3

```
protected void btnGetRateDetails_Click(object sender, EventArgs e)
{
    string strClientRef;
    string strBaseCurrency;
    string strParameterList;

    strClientRef = this.txtClientRef.Text;
    strBaseCurrency = this.txtBaseCurrency.Text;

    strParameterList = "/GetRates?clientRef=" + strClientRef + "&baseCurrency=" + strBaseCurrency;
    //Make the call
    var response = GetRequest(this.txtToken.Text, this.txtBaseURI.Text, this.txtRatePath.Text,
strParameterList).Result;
    this.txtRateDetails.Text = response;
}

static async Task<string> GetRequest(string token, string apiBaseUri, string requestPath, string methodDetails)
{
    using (var client = new HttpClient())
    {
        //setup client
        client.BaseAddress = new Uri(apiBaseUri);
        client.DefaultRequestHeaders.Accept.Clear();
        client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));
        client.DefaultRequestHeaders.Add("Authorization", "Bearer " + token);

        //make request
        var result = await client.GetAsync(requestPath + methodDetails)
            .ConfigureAwait(continueOnCapturedContext: false);

        if (methodDetails.Contains("PostBankDetails")) //need to return true/false if postbankdetails method i.e. ok
or not found
        {
            return result.ReasonPhrase.ToString();
        }
        else
        {
            HttpResponseMessage response = result;
            var responseString = await response.Content.ReadAsStringAsync();
            return responseString;
        }
    }
}
```

Fexco

Iveragh Road

Killorglin, Co. Kerry

Ireland

T: +353 66 976 1258

www.fexco.com