



# Fexco International Payments

## API Guide and Code Samples

---

Nov 2024

Welcome to the Fexco International Payments API Guide!

## Capabilities Overview

API Name	Description
<b>Address API</b>	AutoAddress - Look up an address in the UK and Ireland using a postcode
<b>Banks API</b>	SWIFT Worldwide BIC Codes - Validate worldwide BIC, clearing codes and IBAN
<b>Client API</b>	Fexco International Payments Data - Get client data
<b>Cost Centre API</b>	Fexco International Payments Data – Get, create, and amend Cost Centre
<b>Payee API</b>	Fexco International Payments Data – Get, create, and amend Payees
<b>Payment API</b>	Fexco International Payments Data – Create, get, approve, and cancel a Payment
<b>Product API</b>	Fexco International Payments Data - Get payment product rules, currencies, and countries
<b>Quotations API</b>	Fexco International Payments Data - Get a quote
<b>Rates API</b>	NetDania (15 second feed, 24/7) - Get a conversion, supports 2,600 currency pairs
<b>Sanctions API</b>	WorldCheck Worldwide Sanction\Politically Exposed Person (PEP) - Check a name for known sanctions

[More Details](#)

## Getting Started with Fexco International Payments API

Follow these steps to begin integrating and using our services.

### 1. Set Up as a Fexco International Payments Client

To access the API, you must first be registered as a Fexco International Payments client. Contact our support team if you are not already set up.

## 2. Request API Credentials

Once registered, request a username and password to access the Fexco International Payments Web API. Our team will provide you with the necessary credentials.

## 3. Authenticate Using Token-Based Authentication

With your username and password, you can authenticate to the Web API using token-based authentication. This ensures secure access to our services.

Here's an example of the login details you'll receive after registration:

**Client Name:** Fexco Tax Business (4429)  
**Email:** johnDoe@fexco.com  
**Password:** FexcoTaxBusinessTest123\$

## 4. Test Our APIs on Swagger

Explore and test our APIs using Swagger! Visit our Swagger site at [this link](#) to verify if your token is active and functioning correctly. You can experiment with the available APIs directly from this interface.

The API is free to use for Fexco International Payments clients! Depending on your requirements, we offer different levels of API access tailored to your needs.

## Appendix 1 – Retrieve a token using a user account (C#)

```
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 jobject = JObject.Parse(responseJson);

return jobject.GetValue("access_token").ToString();
}
}
catch (Exception e)
{
    throw e;
}
}

```

## Appendix 2 – Retrieve Specific Bank Country Details (C#)

```

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"))
        {
            // Return true/false if PostBankDetails method, i.e., OK or NotFound
            return result.ReasonPhrase.ToString();
        }
        else
        {
            HttpResponseMessage response = result;
            var responseString = await response.Content.ReadAsStringAsync();
            return responseString;
        }
    }
}
}

```

### Appendix 3 - Retrieve Exchange Rate details (C#)

```

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

    // Assign values
    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"))
        {
            // Return true/false if PostBankDetails method, i.e., OK or NotFound
            return result.ReasonPhrase.ToString();
        }
        else
        {
            HttpResponseMessage response = result;
            var responseString = await response.Content.ReadAsStringAsync();
            return responseString;
        }
    }
}
}

```