Perl Modules Guide: Finance::BankVal::International::GetSWIFT Finance::BankVal::International::GetABA Finance::BankVal::International::IBANvalidate

Contents

Introduction

This document describes the operations and uses of Unified Software's Perl modules, **Finance::BankVal::International::GetSWIFT** which provides full validation of SWIFT BIC codes using the latest SWIFT BIC directory, **Finance::BankVal::International::getABA** which provides full routing number validation using the latest Routing Numbers Database and **Finance::BankVal::International::IBANValidate** which provides full IBAN number validation in accordance with ECBS standards.

The BankVal::International modules handles REST web service calls to Unified Software's BankVal-International web services. The modules handle all aspects of calling the web services requiring one simple call to their exposed method. In the unlikely event of any problems with Unified Software's main server centre, this module will automatically call the backup servers.

Installing this module allows the easy integration of the BankVal International services with other code. This document explains how to install and use the module.

Finding and Installing the Module

The modules can be found on CPAN (Comprehensive PERL Archive Network) at http://search.cpan.org/ , by searching for Finance::BankVal::International.

After downloading the required module(s) it (they) will need to be unpacked and unzipped.

Decompressing/Unpacking the Module

Depending on whether you are on a Linux /UNIX or Windows based system you should follow the respective guide below:

Windows Systems

The package can be unpacked and unzipped in one stage by using a freeware application such as WinZip.

Linux/UNIX Systems

On a Linux/Unix system the operation will take two stages

- First decompress by typing "gzip -d BankVal-UK-0.1.tar.gz "at the command console.
- Next unpack by typing "tar -xof BankVal-UK-0.1.tar2"at the command console.

Build and Install

There is no requirement to compile the BankVal modules, simply place the BankVal folder into any folder on PERL's @INC path. Generally these will be a lib folder e.g. ***C:\Perl\lib*** or ***C:\Perl\site\lib*** on Windows.

Further information on installing modules from CPAN can be found from the following website http://www.cpan.org/modules/INSTALL.html.

Using the Finance::BankVal::International::GetSWIFT Module

Importing the BankVal-International-GetSWIFT Function

To include the getSWIFT module within your code you should:

- First, ensure that the module (with its folder structure) is in a location that is on your PERL @INC path (if you have multiple Unified Software Bank Val modules merge the folders), or add the following line to your code:

```
use lib <<ABSOLUTE PATH TO BANKVAL::INTERNATIONAL::GETSWIFT>>;
```

- Second, import the getSWIFT subroutine by adding the following line to your code:

```
use Finance::BankVal::International::GetSWIFT qw(&getSWIFT);
```

Following these two steps the BankVal-International-getSWIFT service (GetBankDetails2) is accessible from within your code.

Calling the Services

The service can be called by using the following line:

```
my $ans = getSWIFT(@params);
```

Parameters

There are a number of different parameter combinations which can be passed to the GetSWIFT function, these are:

- 1: Format - the response format (either xml, json or csv)
- 2: SWIFT BIC - the BIC code to be validated
- 3: UserID - available from www.unifiedsoftware.co.uk
- 4: PIN - available from www.unifiedsoftware.co.uk

The order of the parameters ***must*** always be as shown above.

The UserID and PIN can also be stored in the LoginConfig.txt file bundled with this module, the use of this file saves passing the PIN and user ID data with each call to getSWIFT, for further information on this see the '*Using the LoginConfig file*' section of this document.

The allowed parameter lists are as follows:

- To call **getSWIFT** and supply UserID and PIN within the call:

```
getSWIFT('$format','$bic','$userID','$PIN');
```

- To call **getSWIFT** if the *LoginConfig.txt* file is being used:

```
getSWIFT('$format','$bic');
```

Using the Finance::BankVal::International::GetABA Module

Importing the Finance-BankVal-International-GetABA Function

To include the getSWIFT module within your code you should:

- First, ensure that the module (with its folder structure) is in a location that is on your PERL @INC path (if you have multiple Unified Software Bank Val modules merge the folders), or add the following line to your code:

```
use lib <<ABSOLUTE PATH TO BANKVAL::INTERNATIONAL::GETABA>>;
```

- Second, import the getABA subroutine by adding the following line to your code:

```
use Finance::BankVal::International::GetABA qw(&getABA);
```

Following these two steps the BankVal-International-getABA service (GetABADetails) is accessible from within your code.

Calling the Services

The service can be called by using the following line:

```
my $ans = getABA(@params);
```

Parameters

There are a number of different parameter combinations which can be passed to the GetABA function, these are:

- 1: Format - the response format (either xml, json or csv)
- 2: ABA code - the ABA code to be validated
- 3: UserID - available from www.unifiedsoftware.co.uk
- 4: PIN - available from www.unifiedsoftware.co.uk

The order of the parameters ***must*** always be as shown above.

The UserID and PIN can also be stored in the LoginConfig.txt file bundled with this module, the use of this file saves passing the PIN and user ID data with each call to getABA, for further information on this see the '*Using the LoginConfig file*' section of this document.

The allowed parameter lists are as follows:

- To call **getABA** and supply UserID and PIN within the call:

```
getABA('$format','$aba','$userID','$PIN');
```

- To call **getABA** if the *LoginConfig.txt* file is being used:

```
getABA('$format','$aba');
```

Using the Finance::BankVal::International::IBANValidate Module

Importing the Finance-BankVal-International-IBANValidate Function

To include the getSWIFT module within your code you should:

- First, ensure that the module (with its folder structure) is in a location that is on your PERL @INC path (if you have multiple Unified Software Bank Val modules merge the folders), or add the following line to your code:

```
use lib <<ABSOLUTE PATH TO BANKVAL::INTERNATIONAL::IBANVALIDATE>>;
```

- Second, import the BankValUK subroutine by adding the following line to your code:

```
use Finance::BankVal::International::IBANValidate qw(&ibanValidate);
```

Following these two steps the BankVal-International-IBANValidate service is accessible from within your code.

Calling the Services

The service can be called by using the following line:

```
my $ans = ibanValidate(@params);
```

Parameters

There are a number of different parameter combinations which can be passed to the ibanValidate function, these are:

- 1: Format - the response format (either xml, json or csv)
- 2: IBAN code - the international bank account number to be validated
- 3: UserID - available from www.unifiedsoftware.co.uk
- 4: PIN - available from www.unifiedsoftware.co.uk

The order of the parameters ***must*** always be as shown above.

The UserID and PIN can also be stored in the LoginConfig.txt file bundled with this module, the use of this file saves passing the PIN and user ID data with each call to ibanValidate, for further information on this see the '*Using the LoginConfig file*' section of this document.

The allowed parameter lists are as follows:

- To call **ibanValidate** and supply UserID and PIN within the call:

```
ibanValidate('$format','$iban','$userID','$PIN');
```

- To call **ibanValidate** if the *LoginConfig.txt* file is being used:

```
ibanValidate('$format','$iban');
```

Returns from the Services

GetABA (getABAdetails)

BankVal International ABA validation web services validate ABA Routing Numbers by performing a lookup on the ABA Routing Numbers Database and return the name and address for a given routing number in XML, JSON or CSV formats depending on the format parameter passed to the call.

IBANValidate (IBANValidate)

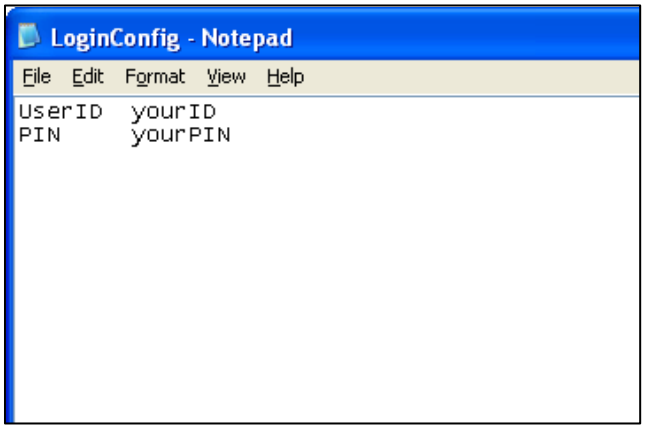
BankVal International IBAN validator is a web service used to validate an IBAN number in full accordance with ECBS standards and returns in XML, JSON or CSV formats depending on the format parameter passed to the call.

GetSWIFT (GetBankDetails2)

BankVal International SWIFT Validation validates SWIFT BICs by checking for their existence in the latest official SWIFT BIC directory and returns in XML, JSON or CSV formats depending on the format parameter passed to the call.

Using the LoginConfig file

If no UserID and PIN information is passed to any of the *BankVal-International* modules, the module will attempt to load information from the **LoginConfig.txt** file. To use this file simply copy it to the same folder that your code is stored in, open it and fill out your UserID and PIN as shown below:



Following this it will be unnecessary to pass the UserID and PIN with the call to the module. If your call is still made with UserID and PIN parameters, then the parameter data will be used rather than the LoginConfig file data.