



# Fiscalog API Aufruf in Visual Basic

Version 1.0, 6/2020

fiscalog Service UG  
(haftungsbeschränkt)  
Ekendonk 16  
47809 Krefeld, DE  
E-Mail: [service@fiscalog.com](mailto:service@fiscalog.com)  
Webseite: [www.fiscalog.com](http://www.fiscalog.com)

## 1 INHALT

2	Einleitung .....	2
2.1	REST Schnittstelle .....	2
2.1.1	Visual Basic.....	2
2.2	Beispiel Programm in Visual Basic .....	3
2.3	Antwort des Aufrufs aus dem Beispiel .....	4

## 2 EINLEITUNG

### 2.1 REST SCHNITTSTELLE

Die Nutzung von REST Schnittstellen erfreut sich wachsender Beliebtheit; hierfür wird das bekannte Hypertext Transfer Protokoll (http) zum Aufrufen von Webseiten im World Wide Web eingesetzt, welches selbst auch nach dem „Representational State Transfer“-Konzept funktioniert.

In den Header eines solchen Aufrufes werden grundlegende Daten geschrieben, z.B. Informationen zum Format der Nachricht, die sich selbst im Body des Aufrufs befindet.

Bei einem Belegaufruf an die Test-API von fiscalog sind im Body Paare aus Schlüsseln und Werten, die Funktionen auslösen oder Belegdaten enthalten.

#### 2.1.1 VISUAL BASIC

Die Programmiersprache Visual Basic, besonders in ihrer Variante Visual Basic .NET von Microsoft wird zur Laufzeit interpretiert und in vielen Anwendungen eingesetzt – zum Beispiel in Registrierkassen.

Für den Aufruf unserer Test-API nach dem REST Prinzip muss über http kommuniziert werden. Dazu gibt es Namespaces, deren Objekte importiert werden können; wir verwenden in unserem Beispiel „System.IO“.

##### 2.1.1.1 BEISPIEL: FUNKTION FÜR REST-AUFRUF

Nachfolgend ist ein kleines Visual Basic Konsolenprogramm aufgeführt, das eine Funktion „REST\_call“ enthält. Dieser werden die URL unserer Test-API sowie der Belegaufruf im JSON-Format als String neben Client-ID und Client-Token übergeben. Beide werden nach Registrierung auf unserer Webseite per E-Mail zugesandt.

Die Funktion liefert entweder die Antwort unserer Test-API oder eine Fehlermeldung z.B. 404, wenn die URL-Adresse falsch geschrieben wurde.

Als Beispiel haben wir einen Funktionsaufruf gewählt, welcher nur aus zwei Wertpaaren besteht („Vorgangstyp“: „Funktionsaufruf“, „Vorgangsbeschreibung“: „TSE\_auflisten“). Über unser eigenes Konto für die Test-API haben wir zuvor verschiedene Cloud-TSE erzeugt, die wir mit diesem Funktionsaufruf auflisten und von der Test-API zurücksenden lassen können.

Die Antwort der Test-API in unserer Konsole ist auf der letzten Seite zu finden.

## 2.2 BEISPIEL PROGRAMM IN VISUAL BASIC

```
Imports System.IO
Imports System.Net

Module fiscalog_REST_call

    Public Const Client_ID = "### Ihre Client-ID aus der Registrierungsgmail ###"
    Public Const Client_Token = "### Ihr Client-Token aus der Registrierungsgmail ###"

    Public Function REST_call(URL, JSON_Belegaufruf, Client_ID, Client_Token)
        Try
            Dim API_call As HttpWebRequest
            Dim API_response As HttpResponseMessage
            Dim API_stream As StreamReader
            API_call = HttpWebRequest.Create(URL)
            API_call.Method = "POST"
            API_call.ContentType = "application/json"
            API_call.Accept = "application/json"
            API_call.Headers.Add("Client-ID", Client_ID)
            API_call.Headers.Add("Client-Token", Client_Token)
            Dim API_body As String = JSON_Belegaufruf
            API_call.GetRequestStream.Write(System.Text.Encoding.UTF8.GetBytes(API_body), 0, System.Text.Encoding.UTF8.GetBytes(API_body).Count)

            API_response = API_call.GetResponse
            API_stream = New System.IO.StreamReader(API_response.GetResponseStream)
            Return API_stream.ReadToEnd

        Catch ex As Exception
            Return "Fehlermeldung: " & ex.Message
        End Try

    End Function

    Sub Main()

        Dim JSON_Belegaufruf As String
        Dim REST_response As String

        JSON_Belegaufruf = "{""Vorgangstyp"": ""Funktionsaufruf"",
""Vorgangsbeschreibung"":""TSE_auflisten""}"

        REST_response = REST_call("http://api.fiscalog.eu/sign/v1/", JSON_Belegaufruf,
Client_ID, Client_Token)

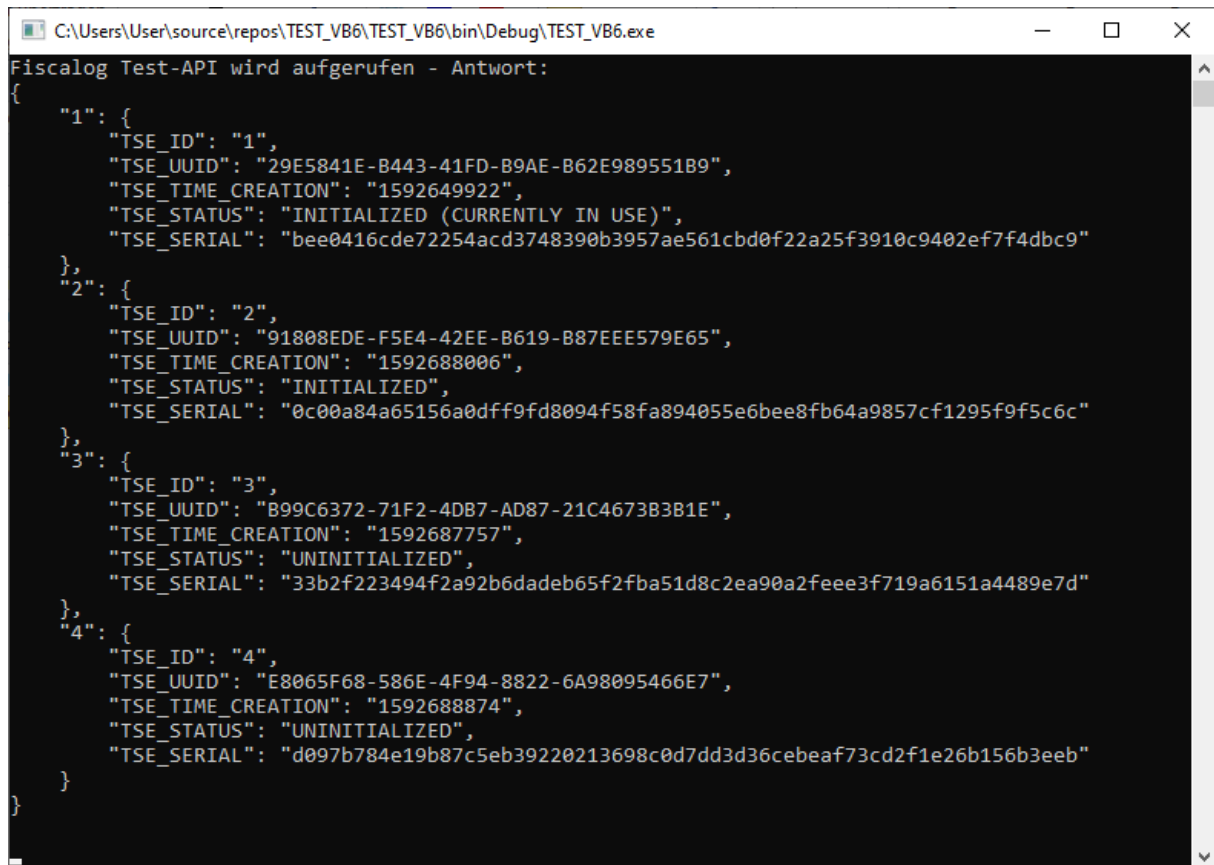
        Console.WriteLine("Fiscalog Test-API wird aufgerufen - Antwort:")
        Console.WriteLine(REST_response)
        Console.ReadKey(True)
    End Sub

END MODULE
```

## 2.3 ANTWORT DES AUFRUFS AUS DEM BEISPIEL

Die fiscallog Test-API sendet eine Liste erzeugter Cloud-TSE zurück, von denen die erste an die Kasse angebunden ist und für Signaturen verwendet wird: „INITIALIZED (CURRENTLY IN USE)“.

Wenn syntaktisch korrekte und DSFinV-K konforme Belegaufrufe an die Schnittstelle übermittelt werden, besteht die Antwort aus Signaturdaten (z.B. Informationen, die auf den Kassensbon zu drucken sind, und andere).



```
Fiscallog Test-API wird aufgerufen - Antwort:
{
  "1": {
    "TSE_ID": "1",
    "TSE_UUID": "29E5841E-B443-41FD-B9AE-B62E989551B9",
    "TSE_TIME_CREATION": "1592649922",
    "TSE_STATUS": "INITIALIZED (CURRENTLY IN USE)",
    "TSE_SERIAL": "bee0416cde72254acd3748390b3957ae561cbd0f22a25f3910c9402ef7f4dbc9"
  },
  "2": {
    "TSE_ID": "2",
    "TSE_UUID": "91808EDE-F5E4-42EE-B619-B87EEE579E65",
    "TSE_TIME_CREATION": "1592688006",
    "TSE_STATUS": "INITIALIZED",
    "TSE_SERIAL": "0c00a84a65156a0dff9fd8094f58fa894055e6bee8fb64a9857cf1295f9f5c6c"
  },
  "3": {
    "TSE_ID": "3",
    "TSE_UUID": "B99C6372-71F2-4DB7-AD87-21C4673B381E",
    "TSE_TIME_CREATION": "1592687757",
    "TSE_STATUS": "UNINITIALIZED",
    "TSE_SERIAL": "33b2f223494f2a92b6dadeb65f2fba51d8c2ea90a2feee3f719a6151a4489e7d"
  },
  "4": {
    "TSE_ID": "4",
    "TSE_UUID": "E8065F68-586E-4F94-8822-6A98095466E7",
    "TSE_TIME_CREATION": "1592688874",
    "TSE_STATUS": "UNINITIALIZED",
    "TSE_SERIAL": "d097b784e19b87c5eb39220213698c0d7dd3d36cebeaf73cd2f1e26b156b3eeb"
  }
}
```