www.bytescout.com

How to add surface chart with spreadsheet sdk in VB.NET and ByteScout Data Extraction Suite

Learn to code in VB.NET to add surface chart with spreadsheet sdk with this step-by-step tutorial

Add surface chart with spreadsheet sdk is simple to apply in VB.NET if you use these source codes below. ByteScout Data Extraction Suite is the bundle that includes three SDK tools for data extraction from PDF, scans, images and from spreadsheets: PDF Extractor SDK, Data Extraction SDK, Barcode Reader SDK and you can use it to add surface chart with spreadsheet sdk with VB.NET.

This prolific sample source code in VB.NET for ByteScout Data Extraction Suite contains various functions and other necessary options you should do calling the API to add surface chart with spreadsheet sdk. Just copy and paste the code into your VB.NET application's code and follow the instructions. If you want to use these VB.NET sample examples in one or many applications then they can be used easily.

The trial version of ByteScout Data Extraction Suite can be downloaded for free from our website. It also includes source code samples for VB.NET and other programming languages.

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about ByteScout Data Extraction Suite

Explore API Documentation

Get Free Training for ByteScout Data Extraction Suite

Get Free API key for Web API

visit www.ByteScout.com

Source Code Files:

Module1.vb

```
Imports Bytescout.Spreadsheet
Imports Bytescout.Spreadsheet.Charts


Module Module1

    Sub Main()

        ' Create new Spreadsheet object
        Dim spreadsheet As New Spreadsheet()
        spreadsheet.RegistrationName = "demo"
        spreadsheet.RegistrationKey = "demo"

        ' Add new worksheet
        Dim sheet As Worksheet = spreadsheet.Workbook.Worksheets.Add("Sample")

        ' Add a data for Surface chart
        Dim count As Integer = 10
        Dim [step] As Double = 2 * Math.PI / (count - 1)
        For i As Integer = 0 To count - 1
            For j As Integer = 0 To count - 1
                sheet(i, j).Value = Math.Sin([step] * Math.Sqrt(i * i + j * j))
            Next
        Next

        ' Add charts to worksheet
        Dim surfaceChart As Chart = sheet.Charts.AddChartAndFitInto(11, 1, 28, 8, Chart
        For i As Integer = 0 To count - 1
            surfaceChart.SeriesCollection.Add(New Series(sheet.Range(0, i, 9, i)))
        Next

        surfaceChart = sheet.Charts.AddChartAndFitInto(11, 10, 28, 17, ChartType.Surfac
        For i As Integer = 0 To count - 1
            surfaceChart.SeriesCollection.Add(New Series(sheet.Range(0, i, 9, i)))
        Next

        surfaceChart = sheet.Charts.AddChartAndFitInto(29, 1, 46, 8, ChartType.Surface
        For i As Integer = 0 To count - 1
            surfaceChart.SeriesCollection.Add(New Series(sheet.Range(0, i, 9, i)))
        Next

        surfaceChart = sheet.Charts.AddChartAndFitInto(29, 10, 46, 17, ChartType.Surfac
        For i As Integer = 0 To count - 1
            surfaceChart.SeriesCollection.Add(New Series(sheet.Range(0, i, 9, i)))
        Next

        ' Save it as XLS
        spreadsheet.SaveAs("Output.xls")

        ' Close the document
        spreadsheet.Close()

        ' Cleanup
        spreadsheet.Dispose()
```

```
        ' Open generated XLS file in default associated application
        Process.Start("Output.xls")

    End Sub

End Module
```

VIDEO

https://www.youtube.com/watch?v=NEwNs2b9YN8

ON-PREMISE OFFLINE SDK

60 Day Free Trial or Visit ByteScout Data Extraction Suite Home Page
Explore ByteScout Data Extraction Suite Documentation
Explore Samples
Sign Up for ByteScout Data Extraction Suite Online Training

ON-DEMAND REST WEB API

Get Your API Key
Explore Web API Docs
Explore Web API Samples

visit www.ByteScout.com

visit www.PDF.co