www.bytescout.com

How to add stock chart with spreadsheet sdk in C# and ByteScout Data Extraction Suite

How to write a robust code in C# to add stock chart with spreadsheet sdk with this step-by-step tutorial

This sample source code below will display you how to add stock chart with spreadsheet sdk in C#. ByteScout Data Extraction Suite: the set that includes 3 SDK products for data extraction from PDF, scans, images and from spreadsheets: PDF Extractor SDK, Data Extraction SDK, Barcode Reader SDK. It can add stock chart with spreadsheet sdk in C#.

The SDK samples given below describe how to quickly make your application do add stock chart with spreadsheet sdk in C# with the help of ByteScout Data Extraction Suite. Just copy and paste the code into your C# application's code and follow the instructions. Enjoy writing a code with ready-to-use sample codes in C#.

ByteScout provides the free trial version of ByteScout Data Extraction Suite along with the documentation and source code samples.

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:

Program.cs

```
using System.Diagnostics;
using Bytescout.Spreadsheet;
using Bytescout.Spreadsheet.Charts;

namespace CSharp
{
    class Program
    {
        static void Main(string[] args)
        {
            // Create new Spreadsheet object
            Spreadsheet spreadsheet = new Spreadsheet();
            spreadsheet.RegistrationName = "demo";
            spreadsheet.RegistrationKey = "demo";

            // Add new worksheet
            Worksheet sheet = spreadsheet.Workbook.Worksheets.Add("Sample");

            // Add few random numbers
            for (int i = 1; i < 6; i++)
            {
                sheet[i, 0].Value = 39082 + i;
                sheet[i, 0].ValueDataTypeByNumberFormatString = Bytescout.Spreadsheet.
            }

            sheet[1, 1].Value = 41301; sheet[1, 2].Value = 24.3; sheet[1, 3].Value = 2
            sheet[2, 1].Value = 35203; sheet[2, 2].Value = 25.4; sheet[2, 3].Value = 2
            sheet[3, 1].Value = 27908; sheet[3, 2].Value = 23; sheet[3, 3].Value = 19.
            sheet[4, 1].Value = 29567; sheet[4, 2].Value = 17.3; sheet[4, 3].Value = 2
            sheet[5, 1].Value = 25895; sheet[5, 2].Value = 20.4; sheet[5, 3].Value = 1

            // Add charts to worksheet
            Chart stockChart = sheet.Charts.AddChartAndFitInto(7, 1, 26, 9, ChartType.
            stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 3, 5, 3)));
            stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 4, 5, 4)));
            stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 5, 5, 5)));

            stockChart = sheet.Charts.AddChartAndFitInto(7, 10, 26, 18, ChartType.Stoc
            stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 2, 5, 2)));
            stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 3, 5, 3)));
            stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 4, 5, 4)));
            stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 5, 5, 5)));

            stockChart = sheet.Charts.AddChartAndFitInto(28, 1, 46, 9, ChartType.Stock
            stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 1, 5, 1)));
            stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 3, 5, 3)));
            stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 4, 5, 4)));
            stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 5, 5, 5)));

            stockChart = sheet.Charts.AddChartAndFitInto(28, 10, 46, 18, ChartType.Sto
            stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 1, 5, 1)));
            stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 2, 5, 2)));
            stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 3, 5, 3)));
            stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 4, 5, 4)));
            stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 5, 5, 5)));
```

```
            // 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");
        }
    }
}
```

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