How to create spreadsheet in PHP and ByteScout Spreadsheet SDK

The tutorial below will demonstrate how to create spreadsheet in PHP

The code below will help you to implement an PHP app to create spreadsheet. ByteScout Spreadsheet SDK: the SDK to create, read, modify and calculate spreadsheets. Formula calculations are supported, import and export to and from JSON, CSV, XML, databases, arrays. It can create spreadsheet in PHP.

Fast application programming interfaces of ByteScout Spreadsheet SDK for PHP plus the instruction and the code below will help you quickly learn how to create spreadsheet. Follow the instructions from the scratch to work and copy the PHP code. Enjoy writing a code with ready-to-use sample PHP codes.

You can download free trial version of ByteScout Spreadsheet SDK from our website to see and try many others source code samples for PHP.

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about ByteScout Spreadsheet SDK

Explore API Documentation

Get Free Training for ByteScout Spreadsheet SDK

Get Free API key for Web API

visit www.ByteScout.com

Source Code Files:

HelloWorld.php

```
<?php

// this PHP script creates XLS file using Spreadsheet SDK and streams it into the brows

/*

IMPORTANT:
to use COM objects in PHP need to enable it by adding the following line into php.ini:

extension=php_com_dotnet.dll

If you are usiung EasyPHP DevServer then to edit php.ini just right-click on the tray i
*/

try
{

// create spreadsheet object
$document = new COM("Bytescout.Spreadsheet.Spreadsheet");

// Add new worksheet
$worksheet = $document->Workbook->Worksheets->Add("123HelloWorld");

// get cell value
$cell = $worksheet->Cell(0,0); // you can also use worksheet.Cell_2("A1") as well. Cel

// set cell value
$cell->Value = "Hello, World!";

// generate and get xls document as an array of bytes
$XLSDocumentArray = $document->GetAsBytesArrayXLS();

// now set headers to properly tell what kind of content we stream into browser
header("Content-Type: vnd.ms-excel");

// set file name to save into
header("Content-Disposition: attachment; filename=\"HelloWorld.xls\"");

// now write the array with content into the output
foreach ($XLSDocumentArray as $byte) echo chr($byte);

}
catch(Exception $err) // catch exception if any
{
header("Content-Type: text/plain"); // switch to plain text output type
echo($err->getMessage()); // write error message to the output
}


?>
```

HowToUseViaDOTNETclass.txt

```
You may use the SDK from PHP on Windows using DOTNET() class in PHP

1) Install the DLL into the GAC (global assembly cache)

 a) Open "c:\Windows\assembly" in Windows Explorer and drag-and-drop the dll into this
 OR
 b) install via command line:

"%programfiles%\Microsoft SDKs\Windows\v6.0\bin\gacutil.exe" -i "Bytescout.Spreadsheet

NOTE: you may need to change %programfiles%\Microsoft SDKs\Windows\v6.0\bin\ into
%programfiles%\Microsoft SDKs\Windows\v6.0A\bin\
or into
%programfiles%\Microsoft SDKs\Windows\v7.0A\bin\

If you are still getting "File not found" then run the search for gacutil.exe in "Progr

2) Now run Windows Explorer and open "C:\WINDOWS\assembly" to view installed .NET assem
3) Find the installed and right-click on it and select "Properties"
4) In the Properties window double-click on "Assembly Name" and copy it to your PHP coc
5) In the Properties window double-click on "Public Key Token" and copy it to your PHP
6) In the Properties window double-click on "Version" value and copy it to your PHP coc

from (4), (5) and (6) you should compose the following string:

$dotnetobj = new DOTNET("ASSEMBLYNAME, Version=VERSIONVALUE, Culture=neutral, PublicKey
"ASSEMBLYNAME.CLASSNAME");

Where
ASSEMBLYNAME = the assembly name  (like Bytescout.Spreadsheet)
VERSIONVALUE = version value you got on step (5)
PUBLICKEYVALUE = public key value you got on step (4)
CLASSNAME = class name from the dll you want to create (for example, "Spreadsheet")

Example for Spreadsheet SDK version 2.4.0.1370:

$dotnetobj = new DOTNET("Bytescout.Spreadsheet, Version=2.4.0.1370, Culture=neutral, Pu
"Bytescout.Spreadsheet.Spreadsheet");

Then you may use $dotnetobj and call its methods like this:
$dotnetobj->SomeMethod() (see the documentation of the SDK for available classes and me
```

VIDEO

https://www.youtube.com/watch?v=nm_7I0PN1TY