メシウス株式会社

# Cloud Functions と DioDocs で Excel や PDF ファイルを出力する

※本資料は、弊社ブログに投稿された記事「Cloud Functions と DioDocs で Excel や PDF ファイルを出力する(1)~(3)」の 連載記事をまとめて資料化した内容となります。下記が記事の原文となります。

Cloud Functions と DioDocs で Excel や PDF ファイルを出力する(1) Cloud Functions と DioDocs で Excel や PDF ファイルを出力する(2) Cloud Functions と DioDocs で Excel や PDF ファイルを出力する(3)

# 目次

Cloud Functions と DioDocs で Excel や PDF ファイルを出力する(1)	3
Cloud Functions とは?	3
実装する内容	3
事前準備	3
Google Cloud でプロジェクトの作成、選択、サービスの有効化	3
Cloud SDK のインストールと初期化	3
.NET の開発環境の設定	3
テンプレートパッケージのインストール	3
Cloud Functions アプリケーションを作成	4
NuGet パッケージを追加	5
DioDocs for Excel を使うコードを追加	6
デバッグ実行で確認	7
Google Cloud ヘデプロイ	8
デプロイしたアプリケーションを確認	9
PDF を出力するには?	11
さいごに	12
Cloud Functions と DioDocs で Excel や PDF ファイルを出力する(2)	13
実装する内容	13
事前準備	13
Cloud Functions アプリケーションを作成	13
NuGet パッケージの追加	15
DioDocs for Excel	15
Google.Cloud.Storage.V1(Google Cloud Client Libraries for .NET)	15
Cloud Storage にバケットを作成	16
DioDocs for Excel を使うコードを追加	17
デバッグ実行で確認	17
Google Cloud ヘデプロイ	19
デプロイしたアプリケーションを確認	20
PDF を出力するには?	21
さいごに	22
Cloud Functions と DioDocs で Excel や PDF ファイルを出力する(3)	23
セルに追加するテキストの日本語フォント(DioDocs for Excel)	23
ワークシートを PDF 出力する際の日本語フォント(DioDocs for Excel)	23

Visual Studio で開発する際の注意点		
PDF ドキュメントを保存する際の日本語フォント	(DioDocs for PDF)	

## Cloud Functions と DioDocs で Excel や PDF ファイルを出力する (1)

本記事では、Google Cloud で提供されている FaaS、Cloud Functions で「<u>DioDocs(ディオドック)</u>」を使用した C# (.NET 6、8)の関数アプリケーションを作成し、Excel や PDF ファイルを出力する方法について紹介します。

#### Cloud Functions とは?

<u>Cloud Functions</u>は <u>Google Cloud</u> で提供されている、各種イベントをトリガーに処理を実行するサーバーレス なアプリケーションを作成できるクラウドサービスです。

#### 実装する内容

今回実装する内容は非常にシンプルです。 関数アプリケーションで <u>HTTP 関数</u>を作成します。この関数の実行時 に DioDocs を使用して Excel と PDF ファイルを作成し、HTTP リクエストのクエリパラメータで受け取った文 字列を追加します。その後、作成した Excel と PDF ファイルを関数から HTTP レスポンスで直接ローカルへ出 力する、といった内容です。

#### 事前準備

#### Google Cloud でプロジェクトの作成、選択、サービスの有効化

「<u>クイックスタート: .NET を使用して HTTP Cloud Functions の関数を作成してデプロイする</u>」の「<u>gcloud CLI</u> <u>を使用した GCP プロジェクトの作成</u>」の手順1~4 のとおりに、Google Cloud プロジェクトを作成、選択して 「Cloud Functions API」サービスと「Cloud Build API」サービスを有効にします。

#### Cloud SDK のインストールと初期化

「gcloud CLI を使用した GCP プロジェクトの作成」の手順5「gcloud CLI をインストールして初期化します。」 を実施します。このリンクからは Cloud SDK のトップページにリンクされているだけで少々わかりづらいので すが、 ここからリンクされている「Cloud SDK のインストール」の手順に沿って Cloud SDK をインストール、 初期化します。

これで gcloud CLI が使えるようになったのでその後、手順 6「gcloud コンポーネントを更新してインストール します。」を実行しておきます。

#### .NET の開発環境の設定

「gcloud CLI を使用した GCP プロジェクトの作成」の手順7「開発環境を準備します。」を実施します。

#### テンプレートパッケージのインストール

dotnet new install コマンドを使用して Cloud Functions のプロジェクトテンプレートをインストールしてお きます。「テンプレート パッケージの使用」に記載の以下のコマンドを実行します。 これで Visual Studio から Cloud Functions のプロジェクトを作成できるようになります。

## Cloud Functions アプリケーションを作成

Visual Studio 2022 でプロジェクトテンプレート「Google Cloud Functions HttpFunction (Google LLC)」を選 択して[次へ] をクリックします。

新しいプロジェクトの作成	- L × テンプレートの検索(S) (Alt+S) ・ すべてクリア(C)
最近使用したプロジェクト テンプレート( <u>R</u> )	C# ・ すべてのブラットフォーム( <u>P</u> ) ・ Google ・
最近アクセスしたテンプレートの一覧は、ここに表示されます。	Google Cloud Functions CloudEvent Function (Google LLC)         A project template for creating a Google Cloud Function to respond to a specific CloudEvent type.         C#       クラウド         Google Cloud Functions CloudEvent Function (Untyped) (Google LLC)         A project template for creating a Google Cloud Function to respond to any CloudEvent.         C#       クラウド         Google
	Google Cloud Functions HttpFunction (Google LLC) A project template for creating a Google Cloud Function to respond to HTTP requests. C# 2ラウド Google
	探しているものが見つからない場合 さらにツールと機能をインストールする 戻る( <u>B</u> ) 次へ( <u>N</u> )

プロジェクト名に「ExcelExportCloudHttpFunction1」を入力して[作成]をクリックします。

		-		>
新しいプロジェクトを構成します				
Google Cloud Functions HttpFunction (Google LLC) C# クラウド Google				
プロジェクト名(J)				
ExcelExportCloudHttpFunction1				
C:¥Users¥kuni¥Desktop •				
ExcelExportCloudHttpFunction1				
	戻る( <u>B</u> )	作反	戈( <u>C</u> )	

## 「ExcelExportCloudHttpFunction1」プロジェクトが作成されます。



## NuGet パッケージを追加

Visual Studio の「NuGet パッケージ マネージャー」から DioDocs for Excel のパッケージ 「<u>DioDocs.Excel.ja</u>」をインストールします。



## DioDocs for Excel を使うコードを追加

DioDocs for Excel で Excel ファイルを作成するコードを追加して HandleAsync を以下のように更新します。

byte[] output;

```
using (var ms = new MemoryStream())
{
    workbook.Save(ms, SaveFileFormat.Xlsx);
    output = ms.ToArray();
}
context.Response.ContentType = "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet";
context.Response.Headers.Add("Content-Disposition", "attachment;filename=Result.xlsx");
await context.Response.Body.WriteAsync(output);
```

DioDocs for Excel で作成した Excel ファイルを MemoryStream に保存し、これを HttpResponse の Body に設 定してダウンロードできるように設定しています。

#### デバッグ実行で確認

作成した Cloud Functions の関数アプリケーションを Visual Studio からデバッグ実行して確認します。ブラウ ザのアドレスバーにコンソールに表示される URL「**http://127.0.0.1:8080**」に、クエリパラメータと文字列 「**?name=DioDocs(ディオドック**)」を追加して、関数アプリケーションを実行します。

⊡ C:¥Users¥kunì¥Desktop¥Excel × + ∨			×
2024-04-05T02:57:42.261Z [Google.Cloud.Functions.Hosting.EntryPoint] [info] Serving function ExcelExport	loudHttpl	Functi	0
2024-04-05T02:57:42.385Z [Microsoft.Hosting.Lifetime] [info] Now listening on: http://127.0.0.1:8080 2024-04-05T02:57:42.386Z [Microsoft.Hosting.Lifetime] [info] Application started. Press Ctrl+C to shut do 2024-04-05T02:57:42.386Z [Microsoft.Hosting.Lifetime] [info] Hosting environment: Production 2024-04-05T02:57:42.386Z [Microsoft.Hosting.Lifetime] [info] Content root path: C:\Users\kuni\Desktop\Exc tpFunction1\ExcelExportCloudHttpFunction1\bin\Debug\net6.0	own. celExport	CloudH	t

実行するとクエリパラメータで渡した文字列「**DioDocs(ディオドック**)」が追加された Excel ファイル 「Result.xlsx」がローカルに出力されます。

🕺 自動保存 💽 🗍 Result.xlsx ・	♀ 検索 (Alt+Q)		<i>ä</i> – o ×
ファイル ホーム 挿入 描画 ページレイアウト 数式 データ 校閲 表示 ヘルプ			ロコメント ピ 共有
A1 $\checkmark$ : $\times \checkmark f_x$ Hello, DioDocs ( $\vec{\tau} \cdot \vec{\tau} \cdot \vec{\tau} \cdot \vec{\tau} \cdot \vec{\tau} \cdot \vec{\tau}$ ) !!			~
A B C D E F G	H I J K L M	N O P Q R	S T
1 Hello, DioDocs(ディオドック)!! 2			
3			
5			
6			
7			
8			
10			
11			
12			
14			
15			
17			
18			
19			
21			
22			
24			
25			
26			
28			
29			
30			
↔ Sheet1 ⊕	· · · · · · · · · · · · · · · · · · ·		
準備完了 🛛 🔀 アクセシビリティ: 検討が必要です			+ 100%

## Google Cloud ヘデプロイ

Visual Studio の「ソリューション エクスプローラー」からプロジェクトを右クリックして、コンテキストメニ ューから [ターミナルで開く] をクリックして「開発者用 PowerShell」を起動します。



gcloud functions deploy コマンドを実行して Google Cloud へ Cloud Functions の関数アプリケーションを デプロイします。

gcloud functions deploy diodocs-excel-export-function1 --gen2 --entry-point
ExcelExportCloudHttpFunction1.Function --runtime dotnet6 --trigger-http --allow-unauthenticated

Google Cloud の Cloud Functions にも以下のようにデプロイが完了した関数アプリケーションが表示されます。

- () Functions - Cloud Functions - 1 × +										- 1	o x		
<del>~</del>	→ C	tons cons	ole.cloud.google.com/functions/li	ist?project=soy-transduce	-419404&support	edpurview=pro	oject				☆ む	=7 🔲	🥹 :
≡	Goo	gle Cloud	<ul> <li>My Project 80113 </li> </ul>	スラッシ	νユ(∕)を使用し	、てリソース、	ドキュメ	ント、ブロダク	トなどを検索	、検索 →	<b>b</b> 1	? :	<u>(</u>
(…)	Clou	d Function	s Functions	+ ファンクションを作成	健 更新		K	uni Ujiie (kuni.ujiie	@mescius.com) がサインインしていま	3	⇒学ぶ	ピリリース	レート
= フ	ィルタ	関数を絞り込	t)									Ø	ш
		環境	名前 个	最終デプロイ日	リージョン	推奨事項	トリガー	ランタイム	割り当てられたメモリ	実行済みの関数		操作	
	$\bigcirc$	2nd gen	diodocs-excel-export-function1	2024/04/05 13:53:21	asia-northeast1		HTTP	.NET 6.0	256 MB	ExcelExportCloudHttpF	unction1.Function	:	

## デプロイしたアプリケーションを確認

ブラウザのアドレスバーに、関数アプリケーションの URL にクエリパラメータと文字列「**?name=DioDocs** (デ ィオドック)」を追加して、関数アプリケーションを実行します。





この関数アプリケーションを実行するとクエリパラメータで渡した文字列「**DioDocs(ディオドック)**」が追加 された Excel ファイル「Result.xlsx」がローカルに出力されます。

▲ 自動保存 (大) 日 Result.xlsx ・	♪ 検索 (Alt+Q)	644.	🍯 🖉 – o 🗙
ファイル ホーム 挿入 描画 ページレイアウト 数式 データ 校閲 表示 ヘルプ			ロコメント ピ 共有
A1 v : × v $f_x$ Hello, DioDocs (ディオドック) !!			
A B C D E F G	H I J K L M	N O P Q	R S T
1 Hello, DioDocs(ディオドック)!!			
3			
4			
5			
7			
8			
9			
10			
12			
13			
14			
16			
17			
18			
20			
21			
22			
24			
25			
26			
28			
29			
30			
31	: • • • • •		
準備完了 🕃 アクセシビリティ: 検討が必要です			

## PDF を出力するには?

Visual Studio の「NuGet パッケージ マネージャー」から DioDocs for PDF のパッケージ

「<u>DioDocs.Pdf.ja</u>」をインストールします。 DioDocs for PDF で PDF ファイルを作成するコードを追加して HandleAsync を以下のように更新します。

```
using GrapeCity.Documents.Pdf;
using GrapeCity.Documents.Text;
using System.Drawing;
using System.IO;
:
public async Task HandleAsync(HttpContext context)
{
 HttpRequest request = context.Request;
 string name = request.Query["name"].ToString();
string Message = string.IsNullOrEmpty(name)
 ? "Hello, World!!"
 : $"Hello, {name}!!";
//GcPdfDocument.SetLicenseKey("");
```

```
GcPdfDocument doc = new GcPdfDocument();
GcPdfGraphics g = doc.NewPage().Graphics;
g.DrawString(Message,
    new TextFormat() { Font = StandardFonts.Helvetica, FontSize = 12 },
    new PointF(72, 72));
byte[] output;
using (var ms = new MemoryStream())
{
    doc.Save(ms, false);
    output = ms.ToArray();
}
context.Response.ContentType = "application/pdf";
context.Response.Headers.Add("Content-Disposition", "attachment;filename=Result.pdf");
await context.Response.Body.WriteAsync(output);
```

# さいごに

}

動作を確認できる Cloud Functions アプリケーションのサンプルはこちらです。 https://github.com/MESCIUSJP/ExcelExportCloudHttpFunction1 https://github.com/MESCIUSJP/PDFExportCloudHttpFunction1

# Cloud Functions と DioDocs で Excel や PDF ファイルを出力する (2)

<u>前回</u>に引き続き、本記事でも Cloud Functions で「<u>DioDocs(ディオドック)</u>」を使用した C#(.NET 6、8)の関 数アプリケーションを作成し、Excel や PDF ファイルを出力する方法について紹介します。

#### 実装する内容

今回も関数アプリケーションで <u>HTTP 関数</u>を作成します。この関数の実行時に DioDocs を使用して Excel と PDF ファイルを作成し、HTTP リクエストのクエリパラメータで受け取った文字列を追加します。

前回は作成した Excel と PDF ファイルを関数から HTTP レスポンスで直接ローカルへ出力していましたが、今回は作成した Excel と PDF ファイルを Cloud Storage へ出力します。

Cloud Storage への保存には Google Cloud API を各言語で使用するための<u>クライアントライブラリ</u>で、.NET ア プリケーション開発向けに提供されている「Google Cloud Client Libraries for .NET」を使用します。

#### 事前準備

事前準備については前回の内容を参照してください。

## Cloud Functions アプリケーションを作成

Visual Studio 2022 でプロジェクトテンプレート「Google Cloud Functions HttpFunction (Google LLC)」を選 択して[次へ] をクリックします。



		-		×
新しいプロジェクトを構成します				
Google Cloud Functions HttpFunction (Google LLC) C# クラウド Google				
プロジェクト名(J)				
ExcelExportCloudHttpFunction2				
場所(L)				
C:¥Users¥kuni¥Desktop •				
ソリューション名(M) 🕕				
ExcelExportCloudHttpFunction2				
ソリューションとプロジェクトを同じディレクトリに配置する(D)				
	戻る(B)	作	■成(C)	

#### 「ExcelExportCloudHttpFunction2」プロジェクトが作成されます。



## NuGet パッケージの追加

#### **DioDocs for Excel**

Visual Studioの「NuGet パッケージ マネージャー」から DioDocs for Excelのパッケージ「<u>DioDocs.Excel.ja</u>」 をインストールします。



#### Google.Cloud.Storage.V1 (Google Cloud Client Libraries for .NET)

同じ手順で Cloud Storage 用のクライアントアプリのパッケージ「Google.Cloud.Storage.V1」をインストー ルします。



## Cloud Storage にバケットを作成

Cloud Storage に DioDocs for Excel で作成した Excel ファイルの保存先になるバケット [diodocs-export] を、 Google Cloud のコンソールから作成します。

✓ 器 diodocs-export − パケットの詳細 >	< +			- o ×
$oldsymbol{\epsilon}  ightarrow oldsymbol{C}$ $oldsymbol{\epsilon}$ console.cloud.go	oogle.com/storage/browser/diodocs	-export;tab=objects?pr	roject = soy-transducer - 419404 & prefix = & force On Objects Sorting Filtering = false	🖈 🖸   🥑 🗊 🖬 🥹 :
≡ Google Cloud	My Project 80113 🔻	スラッシュ (/) を	を使用してリソース、ドキュメント、プロダクトなどを検索 Q 検索	🔸 🔈 🕦 🤄 🗧 🔶
Cloud Storage	← バケットの詳細			⊖ パスに移動 ● 更新 会学ぶ
<ul> <li>● パケット</li> <li>▲ モニタリング</li> <li>◆ 設定</li> </ul>	diodocs-export □ケーション ストレー asia-northeast1(東京) Standard オブジェクト 設定 フォルレグ ブラウザ ● diodocs-export	-ジクラス 公開アク d 非公開 権限 保護 【く :	さス 保護 なし ライフサイクル オブザーバビリティ インペントリレボート オペレーション バケット 3 diodossexport ファイルをアップロード フォルダをアップロード フォルダを作成 データ転送・ ダウンロード 耐除 名前の接頭部のみでフィルタ 〒 フィルタ オブジェクトとフォルダをフィルタ □ 名前 サイズ 種類 作成日降 ○ ストレージクラス 最終更新 公開アク 表示する行がありません	保留を管理 保持を編集 表示 ライブオブジェクトのみ > Ⅱ 2027 ② パージョン履歴 ② 暗号化 ②
نې Marketplace		I		
€ リリースノート				
<1				

## DioDocs for Excel を使うコードを追加

```
DioDocs for Excel で Excel ファイルを作成するコードを追加して HandleAsync を以下のように更新します。
using Google.Cloud.Functions.Framework;
using Google.Cloud.Storage.V1;
using GrapeCity.Documents.Excel;
using Microsoft.AspNetCore.Http;
using System.IO;
using System.Threading.Tasks;
:
public async Task HandleAsync(HttpContext context)
{
   HttpRequest request = context.Request;
   string name = request.Query["name"].ToString();
   string Message = string.IsNullOrEmpty(name)
       ? "Hello, World!!"
       : $"Hello, {name}!!";
   Workbook workbook = new Workbook();
   workbook.Worksheets[0].Range["A1"].Value = Message;
   using MemoryStream outputstream = new MemoryStream();
   StorageClient sc = StorageClient.Create();
   workbook.Save(outputstream, SaveFileFormat.Xlsx);
   await sc.UploadObjectAsync("diodocs-export", "Result.xlsx", "application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet", outputstream);
}
```

DioDocs for Excel で作成した Excel ファイル [Result.xlsx] を MemoryStream に保存し、これを StorageClient クラスの UploadObjectAsync メソッドを使用して Cloud Storage へアップロードしています。

## デバッグ実行で確認

作成した Cloud Functions の関数アプリケーションを Visual Studio からデバッグ実行して確認します。ブラウ ザのアドレスバーにコンソールに表示される URL「**http://127.0.0.1:8080**」に、クエリパラメータと文字列 「**?name=DioDocs**(**ディオドック**)」を追加して、関数アプリケーションを実行します。



実行するとクエリパラメータで渡した文字列「**DioDocs**(**ディオドック**)」が追加された Excel ファイル 「Result.xlsx」が Cloud Storage に出力されます。

▼ 聞 diodocs-export - パケットの詳細	x +	- o x
$\leftrightarrow$ $\rightarrow$ $\mathbf{C}$ $\mathbf{c}$ console.cloud.g	joogle.com/storage/browser/diodocs-export.tab=objects?forceOnBucketsSortingFiltering=true&project=soy-transducer-419404&prefix=&forceOnObjectsSortin	☆ ひ: ଡ ₹ 0 🥹 :
≡ Google Cloud 🏼 🗧	My Project 80113 ▼ スラッシュ (/) を使用してリソース、ドキュメント、プロダクトなどを検索 Q 検索	🔸 🔈 🗘 🧭 ፥ 🧐
Cloud Storage	← バケットの詳細	○ パスに移動 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●
<ul> <li>● パケット</li> <li>☆ 設定</li> </ul>	diodocs-export         ロケーション       ストレーシクラス       公開アクセス       保護         オブシエクト       設定       推測       発送期       なし         オブシエクト       設定       推測       保護       ライフサイクル       オブザーバビリティ       インペントリレボート       オペレーション         レ       パグット > diodocsexport 6	5 <b>ライブオブジェクトのみ →</b> ストレージ <i>ク</i> ラス 最終更 9 Standard 2024/( <b>生</b> :
알 Marketplace		
官 リリースノート		
<۱		

×	自動保存(	▶77 🗄	Result.xls	sx ∽					検索 (Alt+Q	)
ファイ	ルホーム	挿入 描	画 ページし	レイアウト 数	対式 データ	校閲表示	示 開発 ∕	ヘルプ		
A1	~	$ $ : $\times \checkmark$	<i>fx</i> Hello	, DioDocs	(ディオド	ック)!!				
	А	В	С	D	E	F	G	Н	I	J
1	lello, Diol	Docs(ディ	オドック)	!!						
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										

## Google Cloud ヘデプロイ

Visual Studio の「ソリューション エクスプローラー」からプロジェクトを右クリックして、コンテキストメニ ューから [ターミナルで開く] をクリックして「開発者用 PowerShell」を起動します。



gcloud functions deploy コマンドを実行して Google Cloud へ Cloud Functions の関数アプリケーションを

デプロイします。

gcloud functions deploy diodocs-excel-export-function2 --gen2 --entry-point ExcelExportCloudHttpFunction2.Function --runtime dotnet6 --trigger-http --allow-unauthenticated

Google Cloud の Cloud Functions にも以下のようにデプロイが完了した関数アプリケーションが表示されます。

• •	(**) Functions – Cloud Functions – 1 × +														ø	×
÷ +	$\leftarrow \rightarrow \mathbb{C}$ $\approx$ console.cloud.google.com/functions/list?project=soy-transducer-419404 $\bigstar$													=≀ [	I 🥹	
≡	Goo	gle Cloud	S• My Project 80113 🔻	スラッシ	シュ(/)を使用し	ィフリソース、	ドキュメン	、ト、プロダク	トなどを検索	Q 検索	+	>-	Ļ	?	: (	
(…)	(…) Cloud Functions Functions 中ファンクションを作成 C 更新											<b>\$</b> #	泛	ピリリ	-27-	-ト
<b>テ</b> フ	ィルタ	関数を絞りi	10												0	ш
		環境	名前 个	最終デプロイ日	リージョン	推奨事項	トリガー	ランタイム	割り当てられたメモリ	実行済みの関数				操作		
	0	2nd gen	diodocs-excel-export-function2	2024/04/05 17:37:40	4/04/05 17:37:40 asia-northeast1 HTTP .NET 6.0 256 MB ExcelExportC							oudHttpFunction2.Function				

## デプロイしたアプリケーションを確認

ブラウザのアドレスバーに、関数アプリケーションの URL にクエリパラメータと文字列「**?name=DioDocs** (デ ィオドック)」を追加して、関数アプリケーションを実行します。

<ul> <li></li></ul>	■ ● ● : ② : ●登 y Cloud Run ● elevanot-function2
■ Google Cloud       My Project 80113 ▼       スラッシュ (/) を使用してリソース、ドキュメント、プロダクトなどを検索       Q.検索       ◆ □       Q.         「・・・ Cloud Functions       ← ファンクションの詳細       / 編集       前所除       団 □ □ -         ③ diodocs-excel-export-function2       毎 2 世代       (デプロイ特熱: 2024/04/05 17.37.40)       URL: https://asia-northeast1-acy-transducer-419404.cloudfunctions.net/diodocs-excel-export-function2 ピー       ●	② : 😳
(…) Cloud Functions       ← ファンクションの詳細       / 編集 前所除 「□ビー         ② diodocs-excel-export-function2       (#2世代)       (デブロイ特熱: 2024/04/05 17:37:40)       URL: https://asia-northeast1-soy-transducer-419404.cloudfunctions.net/diodocs-excel-export-function2(2)       「       ●	◆学ぶ C y Cloud Run € el-export-function2
G diodocs-excel-export-function2 (第2世代) (デブロイ時熱: 2024/04/05 17:37:40)     URL: <a href="https://asia-northeast1-soy-transducer-419404-cloudfunctions.net/diodocs-excel-export-function2">https://asia-northeast1-soy-transducer-419404-cloudfunctions.net/diodocs-excel-export-function2     To      Powered &amp; diodocs-excel-export-function2     To      To      Powered &amp; diodocs-excel-export-function2     To      To</a>	y Cloud Run @
HTTPS      HTTPS      Intro://asia-portheast1-sov-transducer-419404 cloudfunctions net/diadocs-	



この関数アプリケーションを実行するとクエリパラメータで渡した文字列「**DioDocs**(**ディオドック**)」が追加 された Excel ファイル「Result.xlsx」が Cloud Storage に出力されます。

	😁 diodocs-export – パケットの詳細 🔅	< +	-	o x
÷	→ C 😁 console.cloud.g	pogle.com/storage/browser/diodocs-export.tab=objects?forceOnBucketsSortingFiltering=true&project=soy-transducer-419404&prefix=&forceOn	hObjectsSortin 🛧 🖸   🥑 🐬 🔳	🥹 i
≡	Google Cloud	My Project 80113 ▼ スラッシュ (/) を使用してリソース、ドキュメント、プロダクトなどを検索 Q 検	* L <u></u>	: 🥺
	Cloud Storage	← バケットの詳細	⊖ パスに移動 C更新	⇒学ぶ
	バケット	diodocs-export		
ன் \$	モニタリング 設定	ロケーション ストレージクラス 公開アクセス 保護 asia-northeast1 (東京) Standard 非公開 なし オブジェクト 設定 権限 保護 ライフサイクル オブザーバビリティ インベントリレポート オペレー	->=>	
		ト バケット > diodocs-export □ ファイルをアップロード フォルダを作成 データ転送 ◆ 保留を管理 保持を編集 ダウン 名前の接頭額のみでフィルタ ◆ マイルタ オブジェクトとフォルダをフィルタ	パード 削除 表示 ライブオブジェクトのみ ▼	ш
		名前         サイズ 種類         作品	成日時 😧 ストレージ クラス 最終更	IJ
		Image: S.9 KB         application/vnd.openxmlformats-officedocument.spreadsheetml.sheet         202	24/04/05 17:52:15 Standard 2024/	( <b>± i</b>

×	自動	保存(	77	E	Result	xlsx 🗸					Q	検索 (Alt+Q	)
ファ	イルホ	х-Д	挿入	描画	ī ~	ジ レイアウ	ット 数	文式 データ	校閲 表	示 開発	ヘルプ		
A1		~	$\colon \times$	$\checkmark$ .	fx He	ello, Dic	Docs	(ディオド	ック)!!				
	А		В		С		D	E	F	G	Н	I	J
1	Hello,	Dio	ocs (	ディス	ナドック	·) !!							
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													

## PDF を出力するには?

Visual Studio の「NuGet パッケージ マネージャー」から DioDocs for PDF のパッケージ

「<u>DioDocs.Pdf.ja</u>」をインストールします。 DioDocs for PDF で PDF ファイルを作成するコードを追加して HandleAsync を以下のように更新します。

using Google.Cloud.Functions.Framework;

using Google.Cloud.Storage.V1;

using GrapeCity.Documents.Pdf;

- using GrapeCity.Documents.Text;
- using Microsoft.AspNetCore.Http;

```
using System.Drawing;
using System.IO;
using System.Threading.Tasks;
:
public async Task HandleAsync(HttpContext context)
{
   HttpRequest request = context.Request;
   string name = request.Query["name"].ToString();
   string Message = string.IsNullOrEmpty(name)
       ? "Hello, World!!"
       : $"Hello, {name}!!";
   GcPdfDocument doc = new GcPdfDocument();
   GcPdfGraphics g = doc.NewPage().Graphics;
   g.DrawString(Message,
       new TextFormat() { Font = StandardFonts.Helvetica, FontSize = 12 },
       new PointF(72, 72));
   using MemoryStream outputstream = new MemoryStream();
   StorageClient sc = StorageClient.Create();
   doc.Save(outputstream, false);
   await sc.UploadObjectAsync("diodocs-export", "Result.pdf", "application/pdf", outputstream);
}
```

# さいごに

動作を確認できる Cloud Functions アプリケーションのサンプルはこちらです。 https://github.com/MESCIUSJP/ExcelExportCloudHttpFunction2 https://github.com/MESCIUSJP/PDFExportCloudHttpFunction2

# Cloud Functions と DioDocs で Excel や PDF ファイルを出力する (3)

<u>前回</u>と<u>前々回</u>の記事では <u>Cloud Functions</u> で「<u>DioDocs(ディオドック)</u>」を使用した C#(.NET 6、8)の関数ア プリケーションを作成し、Excel や PDF ファイルを出力する方法について紹介しました。

今回は Cloud Functions で DioDocs を利用する際に、日本語フォントを使用する Tips を紹介します。

#### セルに追加するテキストの日本語フォント(DioDocs for Excel)

セルに追加するテキストの日本語フォントを設定したい場合は、Font プロパティを使用します。

Workbook workbook = new Workbook();
workbook.Worksheets[0].Range["A1"].Font.Name = "MS P明朝";

自動保存 ● カ) 日	9 ~ ~ ~ <del>-</del>		Re	sult.xlsx - E	Excel		₽ 検	索 (Al
ファイル <b>ホーム</b> 挿入	ページレ1	(アウト 数:	式 データ	校閲	表示開発	き ヘルプ		
	月朝	~ 11 ~	A A	ΞΞ	ð97 ~	いがあって	全体を表示す	3
	<u>U</u> ~   <u>+</u> ~	<u></u>	✓ ア →	≣≡≡	<u>←</u> = →=	豆 セルを結合	して中央揃え	~
クリップボード 🗔	フォント		E1		百百	置		Ŀ
A1		X 🗸	fx こん	にちは、世	t界!			
A B	С	D	E	F	G	Н	I	
1 こんにちは 世界!								
2								
3								
4								

セルではなくシート全体のフォントを設定したい場合はこちらのナレッジベースを参考にしてください。 「シート全体のフォントを設定する方法」を見る

#### ワークシートを PDF 出力する際の日本語フォント(DioDocs for Excel)

Cloud Functions で .NET 6、8 のランタイムが含まれる関数アプリケーションの実行環境 OS は、「Ubuntu 22.04」 になっています。こちらには日本語を表示可能なフォント「Droid Sans Fallback」が含まれてはいますが、そ の他の日本語フォントは含まれていません。そのため、デフォルトの状態で PDF 出力を実行すると指定したフ ォントで表示されなかったり、文字化けが発生したり、文字列そのものが表示されない現象が発生します。

「Linux 環境で PDF エクスポートすると文字化けが発生する」を見る

そこで DioDocs for Excel を使用して PDF 出力を実行する場合は、あらかじめ使用する日本語フォントを追加し

ておく必要があります。本記事では「IPAex フォント」を使用します。

Cloud Functions では、関数アプリケーションのプロジェクト内のフォルダをそのまま参照することができます。 プロジェクト内に任意のフォルダ(「diodocs-fonts」フォルダなど)を作成して使用するフォントファイルを格 納します。



そのフォルダを Workbook.FontsFolderPath プロパティで設定します。

```
public async Task HandleAsync(HttpContext context)
{
    HttpRequest request = context.Request;
    string name = request.Query["name"].ToString();
    string Message = string.IsNullOrEmpty(name)
        ? "こんにちは、世界!"
        : $"こんにちは、世界!";
    Workbook.FontsFolderPath = "./diodocs-fonts";
    Workbook.Worksheets[0].Range["A1"].Font.Name = "IPAex ゴシック";
    workbook.Worksheets[0].Range["A2"].Font.Name = "IPAex 明朝";
    workbook.Worksheets[0].Range["A2"].Value = Message;
    Workbook.Worksheets[0].Range["A2"].Value = Message;
```

byte[] output;

```
using (var ms = new MemoryStream())
{
    workbook.Save(ms, SaveFileFormat.Pdf);
    output = ms.ToArray();
}
context.Response.ContentType = "application/pdf";
context.Response.Headers.Add("Content-Disposition", "attachment;filename=Result.pdf");
await context.Response.Body.WriteAsync(output);
```

## Visual Studio で開発する際の注意点

}

Visual Studio では Function.cs などのソースファイルを保存する際の文字コードはデフォルトが「Shift-JIS」と なっています。このままデプロイして実行すると Message にコード上で追加した日本語文字列が文字化けして しまいます。この文字化けを解消するために Visual Studio でソースファイルを保存する際は「UTF-8」を設定 して保存するようにしてください。

▶ 名前を付けてファイルを保存				X
$\leftrightarrow$ $\rightarrow$ $\checkmark$ $\uparrow$	ExcelEx > ExcelExportCloudHttpFunction1 >	~ C	우 ExcelExportCloudHttpFunc	ti 🖣
整理・ 新しいフォルダー				?
🔤 晋 - 個人用		更新日時	種類サイズ	
	bin	2022/11/17 16:06	ファイル フォルダー	
	diodocs-fonts	2022/11/17 11:25	ファイル フォルダー	
		2022/11/17 17:41	ファイル フォルター	6
	ExcelExportCloudHttpFunction1.csproj	2022/11/17 13:49		1
🔀 ピクチャ 🛛 🖈 💻	Function.cs	2022/11/17 17:34	C# ソース ノアイル	3
🎦 ミュージック 🛛 🖈				
🎦 ビデオ 🛛 🖈				a
PdfExportCloudHt				
ExcelFxnortCloud				
ファイル名(N): C:¥User		∉ExcelExportCloudHttpFur	nction1¥Function.cs	$\sim$
ファイルの種類(T): すべてのフ	ファイル (*.*)			
ヘ フォルダーの非表示		上書	き保存(S) 🔻 キャンセル	
		Dath "	上書き保存(S)	.:
A 31 32	Workbook.FontsFold	erPath = .	エンコード付きで保存(V)	

~ 保存オプションの詳細設定		?	×
C:¥Users¥kuni¥Desktop ¥ExcelExportCloudHttpFunction1¥ExcelExport	CloudHttpFuncti	on1¥Funct	ion.cs
エンコード(E):			
Unicode (UTF-8 シグネチャ付き) - コードページ 650	01		~
行の終わり(L):			
現在の設定			$\sim$
	OK	キャン	セル

「IPAex フォント」を設定したセルに日本語の文字列を持つ Excel ワークブックを PDF ファイルへ出力すると、 日本語が文字化けしたりすることなく「IPAex フォント」が設定されて正しく文字列が表示されていることが確 認できます。

Result.pdf - Adobe Acro	obat Reader (64-bit)															
ファイル (F) 編集 (E) 表示	示(V) 署名(S) ウィンドウ(W)	ヘルプ(H)														
ホーム ツール	Result.pdf	×														
🖹 🕁 🏟	🖶 🔍	$\bigcirc$		1 / 1	k		Θ	$\oplus$	158% 🔻	₽ I I I I I I I I I I I I I I I I I I I	↓	F	(Jac	Ð	Ŵ	Q
	- / - + -	D' D	(	<b>_</b> *	L I & .	. <b>–</b> )										
	こんにちは、	DioDo	cs (·	アイオ	「トッ	19)	!									
	こんにちは、	DioDo	cs (	ディオ	ナドッ	ック)	!									
	文書のプロパティ										×					
	概要 セキュリティ フォント	カスタム 詳細	設定													
	🖃 🍸 IPAexGothic (埋め	込みサブセット)														
	種類 : TrueTyp	e (CID)														
•	エンコーディング	: Identity-H														
	🖃 <table-of-contents> IPAexMincho (埋め</table-of-contents>	め込みサブセット)														
	種類 : TrueTyp	e (CID)														
	エンコーディング	: Identity-H														

#### PDF ドキュメントを保存する際の日本語フォント(DioDocs for PDF)

DioDocs for PDF で作成した PDF ドキュメントで日本語フォントを利用する場合も、「ワークシートを PDF 出力する際の日本語フォント(DioDocs for Excel)」と同じ手順で日本語フォントを追加します。

以下のコードのように Font.FromFile メソッドで日本語フォントのファイルを設定して、「IPAex ゴシック」を 設定した日本語の文字列を持つ PDF ドキュメントを出力すると、日本語が文字化けしたりすることなく「IPAex ゴシック」が設定されて正しく文字列が表示されていることが確認できます。

```
public async Task HandleAsync(HttpContext context)
   HttpRequest request = context.Request;
   string name = request.Query["name"].ToString();
   string Message = string.IsNullOrEmpty(name)
       ? "こんにちは、世界!"
       : $"こんにちは、{name}!";
   Font font1 = Font.FromFile(Path.Combine("diodocs-fonts", "ipaexg.ttf"));
   GcPdfDocument doc = new GcPdfDocument();
   GcPdfGraphics g = doc.NewPage().Graphics;
   g.DrawString(Message,
      new TextFormat() { Font = font1, FontSize = 12 },
      new PointF(72, 72));
   byte[] output;
   using (var ms = new MemoryStream())
   {
       doc.Save(ms, false);
       output = ms.ToArray();
   }
   context.Response.ContentType = "application/pdf";
   context.Response.Headers.Add("Content-Disposition", "attachment;filename=Result.pdf");
   await context.Response.Body.WriteAsync(output);
```

{

}

♪ Resul ファイル	lt.pdf - Ad (F) 編集	dobe Ac 〔(E) 赤	crobat Re 毫示(V)	eader (64 署名(S)	4-bit) ウィンドウ(\	N) ///	プ(H)																	
木	ムッ	ール		Re	sult.pdf	×																		
₿		എ		୍			$(\uparrow)$			1 / 1		k	$\bigcirc$	Θ	$\oplus$	158%	*	₽	$\downarrow$	F		Ó.	Ŵ	Q
				2	んに	ちに	よ、	Dio	D	ocs	(	デ・	ィオ	ドッ	ック)	) !								
						.e= .														 				
					又害の刀口	ハティ セキュリティ	フォン	► <u></u> лス9 <i>Ц</i>	ム詩	細設定											×			
					この文書で	で使用してい	いるフォン	ŀ																
					- 1	PAexGo 種類	thic (埋&	が込みサブセ pe (CID)	፻፶ኮ)															
•						17	コーディング	ヴ:Identity-	-H															

なお、こちらも Visual Studio で開発する場合は、日本語の文字化けを回避するためにソースファイルの文字コ ードを「Shift-JIS」から「UTF-8」に変更しておく必要があります。

弊社 Web サイトでは、製品の機能を気軽に試せるデモアプリケーションやトライアル版も公開していますので、こちらもご確認いただければと思います。

- デモアプリケーション (DioDocs for Excel) を試す
- デモアプリケーション(DioDocs for PDF)を試す
- トライアル版をダウンロードして試す

また、ご導入前の製品に関するご相談やご導入後の各種サービスに関するご質問など、お気軽にお問合せくだ さい。

- 問合せ先を確認する
- 個別相談会(Web 会議)について確認する