メシウス株式会社

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

2024年7月29日

※本資料は、弊社ブログに投稿された記事「Azure 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。

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

目次

Azure Functions と DioDocs で Excel や PDF ファイルを出力する (1)	2
Azure Functions とは	2
実装する内容	2
アプリケーションを作成	2
NuGet パッケージの追加	4
DioDocs for Excel を使うコードを追加	4
DioDocs for PDF を使う関数を追加	5
DioDocs for PDF を使うコードを追加	7
デバッグ実行で確認	8
Azure ヘデプロイ	10
デプロイしたアプリケーションを確認	14
さいごに	15
Azure Functions と DioDocs で Excel や PDF ファイルを出力する(2)	16
実装する内容	16
アプリケーションを作成	16
NuGet パッケージの追加	
Azure Blob Storage を使うコードを追加	19
DioDocs for Excel を使うコードを追加	20
DioDocs for PDF を使う関数を追加	21
Azure Blob Storage を使うコードを追加	22
DioDocs for PDF を使うコードを追加	23
デバッグ実行で確認	24
Azure ヘデプロイ	26
デプロイしたアプリケーションを確認	
さいごに	32
Azure Functions と DioDocs で Excel や PDF ファイルを出力する(3)	33
セルに追加するテキストの日本語フォント(DioDocs for Excel)	33
ワークシートを PDF 出力する際の日本語フォント(DioDocs for Excel)	34
PDF ドキュメントを保存する際の日本語フォント(DioDocs for PDF)	

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

本記事では、Azure Functions で「<u>DioDocs(ディオドック)</u>」を使用した C#(.NET 8)のクラスライブラリを ベースにした関数を作成し、Excel や PDF ファイルを出力する方法について紹介します。

Azure Functions とは

<u>Azure Functions</u>は <u>Microsoft Azure</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。今回は <u>Visual Studio 2022</u> で Azure Functions アプ リケーションを作成し、Azure ヘデプロイして確認します。

実装する内容

今回実装する内容は非常にシンプルです。Azure Functions アプリケーションで<u>HTTPトリガー</u>を使用する関数 を作成します。関数の実行時に DioDocs を使用して Excel と PDF ファイルを作成し、クエリパラメータで受け 取った文字列をそれぞれのファイルへ追加します。その後、作成した Excel と PDF ファイルを <u>FileContentResult</u> で直接ローカルへ出力する、といった内容です。

アプリケーションを作成

以下のドキュメントを参考に Azure Functions アプリケーションを作成していきます。 クイック スタート:Visual Studio を使用して Azure で初めての C# 関数を作成する

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



プロジェクト名 DioDocsFileExportFunctionApp を入力して [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				
Azure Functions C# Azure クラウド				
プロジェクト名(J)				
DioDocsFileExportFunctionApp				
場所(L)				
C:¥Users¥kuni¥Desktop •				
ソリューション名(M) 🚯				
DioDocsFileExportFunctionApp				
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)				
プロジェクト は "C:¥Users¥kuni¥Desktop¥DioDocsFileExportFunctionApp¥DioDocsFileExportFunctionApp¥" で作成されます				
				_
	冥る(B)	次	(∩(N)	

Azure Functions で作成する関数のテンプレートを選択します。Http Trigger を選択して [作成] をクリック します。

	-		×
追加情報			
Azure Functions C# Azure クラウド			
Functions worker 🛈			
.NET 8.0 (長期的なサポート)			
Function 🛈			
Http trigger -			
🖌 ランタイム ストレージ アカウントに Azurite を使用する (AzureWebJobsStorage) 🕕			
コンテナーのサポートを有効にする ①			
Authorization level 🕥			
Function •			
NET Aspire オーケストレーションへの参加 🕕			
戻る(B)		₣成(C)	

DioDocsFileExportFunctionApp プロジェクトが作成されます。



NuGet パッケージの追加

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



DioDocs for Excel を使うコードを追加

DioDocs で Excel ファイルを作成するコードを追加して Function1 を以下のように更新します。

```
public static class Function1
{
   [FunctionName("Function1")]
   public static async Task<IActionResult> Run(
       [HttpTrigger(AuthorizationLevel.Function, "get", "post", Route = null)] HttpRequest req,
       ILogger log)
   {
       log.LogInformation("C# HTTP trigger function processed a request.");
       string name = req.Query["name"];
       string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
       dynamic data = JsonConvert.DeserializeObject(requestBody);
       name = name ?? data?.name;
       string Message = string.IsNullOrEmpty(name)
           ? "Hello, World!!"
           : $"Hello, {name}!!";
       Workbook workbook = new Workbook();
       workbook.Worksheets[0].Range["A1"].Value = Message;
       byte[] output;
       using (var ms = new MemoryStream())
       {
           workbook.Save(ms, SaveFileFormat.Xlsx);
           output = ms.ToArray();
       }
       return new FileContentResult(output, "application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet")
       {
           FileDownloadName = "Result.xlsx"
       };
   }
}
```

DioDocs for PDF を使う関数を追加

ソリューションエクスプローラーから DioDocsFileExportFunctionApp プロジェクトを右クリックして[追加] - [新しい Azure 関数] を選択して、DioDocs で PDF ファイルを作成する関数 Function2 を追加します。



新しい項目の追加 - DioDocsFileExportFunctio	onApp					×
▲ インストール済み	並べ替え	t: 既定 🗸 🔡 📃			検索 (Ctrl+E)	₽-
▲ C# 個の項目 ▶ Web	₽ ₽	クラス	C# 個の項目	•	種類: C# 個の項目	
コード データ	f	Azure 関数	C# 個の項目		プロプエアFIC AZORE 因気を追加しより。	
全般 ASBNET Corp	₽	Class for U-SQL	C# 個の項目			
Avalonia	•0	インターフェイス	C# 個の項目	ï		
SQL Server Storm Items	£15_	コンポーネント クラス	C# 個の項目			
Uno Platform Uno Platform (UWP)	Ð	ADO.NET Entity Data Model	C# 個の項目	l		
XAML for Blazor	8	Code Analysis 規則セット	C# 個の項目			
▶ オンライン	Ð	ComponentOne EF 6.x DbContext Generator	C# 個の項目			
	า	DataSet	C# 個の項目			
	P	editorconfig File (empty)	C# 個の項目			
	P	editorconfig ファイル (.NET)	C# 個の項目			
	Ð	EF 5.x DbContext ジェネレーター	C# 個の項目			
	Ð	EF 6.x DbContext ジェネレーター	C# 個の項目			
		HTML ページ	C# 個の項目	Ŧ		
名前(N): Function2.cs						
					追加(A) キャンセ	211

関数のテンプレートを選択します。Http Trigger を選択して [追加] をクリックします。

新しい Azure 関数 - Function2	×
Service Bus Queue trigger	Authorization level
🥅 Http trigger	Function
I Http trigger with OpenAPI	
S Timer trigger	
🚥 Queue trigger	
5 Dapr Service Invocation Trigger	
🔊 Kusto output binding	
🔊 Kusto input binding	
🤣 SQL trigger	
🤣 SQL output binding	
🤣 SQL input binding	
🤣 Dapr Topic Trigger	
🐼 Dapr Publish Output Binding	
	追加(d) キャンセル(C)

DioDocs for PDF を使うコードを追加

DioDocs で PDF ファイルを作成するコードを追加して Function2 を以下のように更新します。

```
public static class Function2
{
   [FunctionName("Function2")]
   public static async Task<IActionResult> Run(
       [HttpTrigger(AuthorizationLevel.Function, "get", "post", Route = null)] HttpRequest req,
       ILogger log)
   {
       log.LogInformation("C# HTTP trigger function processed a request.");
       string name = req.Query["name"];
       string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
       dynamic data = JsonConvert.DeserializeObject(requestBody);
       name = name ?? data?.name;
       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));
byte[] output;
using (var ms = new MemoryStream())
{
    doc.Save(ms, false);
    output = ms.ToArray();
  }
return new FileContentResult(output, "application/pdf")
  {
    FileDownloadName = "Result.pdf"
  };
}
```

デバッグ実行で確認

}

作成した Azure Functions アプリケーションをローカルでデバッグ実行して確認します。Visual Studio からデバッグ実行すると以下のコンソールが表示されます。



アプリケーションに含まれる関数の URL は http://localhost:7264/api/Function1、 http://localhost:7264/api/Function2 となっています。この URL にクエリパラメータと文字 列?name=DioDocs を追加して、それぞれの関数をブラウザで実行します。



ローカルのフォルダに保存された Result.xlsx、Result.pdf を確認します。クエリパラメータで渡した文字列 DioDocs が表示されていれば成功です。



Azure ヘデプロイ

作成した Azure Functions アプリケーションを Azure ヘデプロイして確認します。ソリューションエクスプロー ラーから DioDocsFileExportFunctionApp プロジェクトを右クリックして [発行] を選択します。



公開するターゲットは「Azure」を選択します。特定のターゲットは「Azure Function App (Windows)」を選択 します。

公開 _{現在公開している場所}			×
ターゲット		Azure アプリケーションを Microsoft クラウドにホストします	
	.	Docker Container Registry アプリケーションを Docker イメージに対応したサポートされている Container Registry に発行します	
		フォルダー アプリケーションをローカル フォルダーまたはファイル共有に発行します	
	\hookrightarrow	プロファイルのインポート アプリを配置するための発行設定をインポートします	
		戻る(B) 次へ(N) 完了(F) キャンセル(C	-)

			×
公開			
アフリケーションをホストす	るためにと	の Azure サーヒスを使用しますか?	
ターゲット	4	Azure Function App (Windows)	+
特定のターゲット		勤的に入りーリングされ、コートをオンティントで美行するサーハーレス コンビューティングにアノリケーンヨン コートを発行します	£
	\$	Azure Function App (Linux) 動的にスケーリングされ、コードをオンデマンドで実行するサーバーレス コンピューティングにアプリケーション コードを発行しま す	ŧ
	% >	Azure Function App コンテナー アプリケーションを Docker イメージとして Azure Container Registry に発行し、Azure Function App で実行します	t
		Azure コンテナー レジストリ アプリケーションを Docker イメージとして Azure Container Registry に発行します	
		戻る(B) 次へ(N) 完了(F) キャンセル(C	_)

アプリケーションの名前やリソースグループなどを設定して[作成]をクリックします。

4>	Function App (Windo _{新規作成}	x (), insertion with the distance () in the maximum of the distance of the d
名前(N)		
DioDocsFile	eExportFunctionApp20240726181328	
サブスクリプシ	ヨン名(S)	
Visual Studi	io Enterprise サブスクリプション - MPN	•
リソース グルー	プ(G)	
diodocs-rg	(Japan East)	← 新規作成
プランの種類(P)	
消費		•
場所(L)		
Japan East		-
Azure Stora	ge	
diodocs (Ja	pan East)	← 新規作成
Application	Insights(P)	
DioDocsFile	ExportFunctionApp* ()	← 新規作成
エクスポート	(E)	作成(R) キャンセル(C)

公開 ^{既存の Azure 関数を}	選択するか、新規作成します	× Spinner interestions in a proceediments are
ターゲット 特定のターゲット	サブスクリプション名(S) Visual Studio Enterprise サブスクリプション - MPN	•
Functions インスタンス	DioDocsFileExportFunctionApp20240726181 Q	+ 新規作成 [] 心
	戻る(B)	次へ(N) 完了(F) キャンセル(C)

これで公開の準備が完了しました。Visual Studio で [発行] をクリックして作成した Azure Function アプリケ ーションを Azure ヘデプロイします。

📢 ファイル(F) 編集(E) 表示(V) Git(G) プロジェ	ェクト(P) ビルド(B) デバッグ(D) テスト	(S) 分析(N) ツール(T) 拡張機能(X) ウィンドウ(W) ヘルプ(H)		op 🕲 👽 — 🖬 🗙
	g 🔹 Any CPU 🔹 🕨 DioDocs	FileExportFunctionApp 🔹 ▷ 🍏 👻 👼 👼 🖕		郡 GitHub Copilot ៤ 反
DioDacsFileEctionApp:公開 # × 接続済みサービス 公開	DioDocsFileExportFunction Azure Function App (Winds ・ 新しいプロファイル その他のアクジ 公開準備が完了しました。 設定 構成 ターゲット ランタイム すべての設定を表示	App20240726181328 - Zip Deploy pubxml ・ >>>>> /コン・ Release 移植可能	- ♥ €t 発行(U)	yy」- ション 12 スプロ- ラ- ・ ↓ × 替 ● ・ キ ● ● ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
	ホスティング サブスクリブンョン リソース グループ リソース名 ユーザー名 パスワード サイト: https://diodocsfileexportfu	d575d13c-7bf8-4ce5-a30f-017be3a7127a diodocs-rg DioDocsFileExportFunctionApp20240726181328 SDioDocsFileExportFunctionApp20240726181328 		< ソリコーション エクスプローラー Git 変更 プロパティ マーリー× ■ 弊 ♪
	サービスの依存瞬係 ② Azure Application Insights: Di 接続の設定: APPLICATIONINSI ■ Azure Storage: diodocs 接続文字列名: AzureWebJobs	aDocsFileExportFunctionApp GHTS_CONNECTION_STRING Storage	+ ひ 容 … ● 接続されました … ● 接続されました …	
エラー一覧出力				
し 準備完了				个 ソース管理に追加 ▲ 🔲 リポジトリの選択 ▲ 🚨

公開が完了するとメッセージが表示されます。

ペイル(F) 編集(E) 表示(V) Git(G) プロジ ※ ⓒ ▼ ⑨ 鋼 ▼ 醫 ◎ ■ ♥ ▼ ♡ ▼ ○ ▼ Debu	ゴクト(P) ビルド(B) デバッグ(D) テスト(S) g • Any CPU • DioDocsFile	分析(N) ツール(T) 拡張機能(X) ウィンドウ(W) ヘルプ ExportFunctionApp ▼) ダ ↓ ■ ↓ 🛱 🚽 ਦ	(H) ノク 検索 ・ DioDocsFileExportFunction/	App 🔞 👽 — 리 × & GitHub Copilot 년 문
P-DioDocsFileEctionApp: 公開 + × 接続済みサービス 公開	DioDocsFileExportFunctionAp Azure Function App (Windows + 新しいプロファイル その他のアクショ:	p20240726181328 - Zip Deploy.pubxml •	● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	 >> ソリューション I クスプローラー ・ マ ● マ ● マ ● マ ● マ ● マ ● マ ● マ ● マ ● マ
	² 2024(0)/26 0) 1625 に公開加減(0) 移動 設定 構成 ターゲット ランタイム すべての設定を表示 + フランパ	Release P 移植可能 P		
	サブスクリプション リソース グループ リソース名 ユーザー名 パスワード サイト: https://diodocsfileexportfunct	d575d13c-7bf8-4ce5-a30f-017be3a7127a []] diodocs-rg DioDocsFileExportFunctionApp20240726181328 \$DioDocsFileExportFunctionApp20240726181328 @ 		く → → ソリューション エクスプローラー Git 変更 プロパティ → # × 部 弊 ♪
	サービスの弦存関係 ② Azure Application Insights: DioD 接続の設定: APPLICATIONINSIGH ■ Azure Storage: diodocs 接続文字列名: AzureWebJobsStc	ocsFileExportFunctionApp ITS_CONNECTION_STRING rrage	+ 2 27 … ● 接続されました … ● 接続されました …	
エラ				↑ ソース管理に追加 🔺 🔳 リボジトリの遮沢 🔺 🗘

デプロイしたアプリケーションを確認

Azure ポータルでデプロイしたアプリケーションに含まれる関数 Function1、Function2 が表示されます。

木	ム > 関数アプリ >						_
»		FunctionApp	20240726181328 🖈 😁				
	<mark> <i>▶</i> 検索 </mark>	🖸 参照 🜔 最新の	/情報に更新 🔲 停止 📿 再起動 💳 スワップ 🚽 発行プロフ	アァイルの取得 🏷 発行プロフ	ァイルのリセット 🞍 アプリ	コンテンツのダウンロード …	
		へ 要点					
	🧧 アクティビティ ログ	リソース グループ (URL		onapp20240726181328.azure	
	☆ アクセス制御 (IAM)	状態	:実行中	オペレーティング シス	Windows		
	🧼 9Ú	場所 (<u>移動</u>)	: Japan East	App Service プラン			
	∂ 問題の診断と解決	サブスクリプション (.	: <u>Visual Studio Enterprise サブスクリプション - MPN</u>	ランタイム バージョン	4.834.3.22875		
	🏮 Microsoft Defender for Cloud	サブスクリプション I) : d575d13c-7bf8-4ce5-a30f-017be3a7127a				
	🗲 イベント (プレピュー)	タグ (編集)					
	💡 より良い連携 (プレビュー)						
	🔟 ログ ストリーム	関数 メトリック	プロパティ 通知(0)				
	〉 関数	{ } ローカル環境の語	定 > 🖒 最新の情報に更新				
	〉 デプロイメント	9 名前でフィルク・					
	> パフォーマンス						
	〉 設定	名前		トリガー	状態	モニター	
	> App Service プラン				🔗 右òh	1座75年1.ナトビ	
	〉 開発ツール	Punction		nur			
	> API	Function2		нттр	🥑 有効		
	> 監視	·					
	> オートメーション						
	〉 サポート + トラブルシューティング						

関数 Function1 をクリックして「コードとテスト」から「テスト/実行」を選択します。「関数の URL を取得」 が表示されるのでこちらをクリックします。この関数の URL をコピーします。

ホーム > 関数アプリ > DioDocsFileExportFunctionApp20240726181328 >	関数の URL を取得	
Function1 コードとテスト		
DioDocsFileExportFunctionApp20240726181328	_master (ホスト キー)	
コードとテスト 統合 関数キー 呼び出し ログ メトリックス	https://diodocsfileexportfunctionapp20240726181328.azurewebsites.net/api/Function	D
🗄 保存 🏷 破棄 🌔 最新の情報に更新 👂 テスト/実行 💿 関数の URL の取得 📋 無効化 🃋 削除 🌴 アップロード 🚷 リソース	default (ホスト キー)	
────────────────────────────────────	https://diodocsfileexportfunctionapp20240726181328.azurewebsites.net/api/Function	D
r¶ DioDocsFileExportFunctionApp20240726181328 / Function 1 / function ison ∨	default (閒数十一)	
	https://diodocsfileexportfunctionapp20240726181328.azurewebsites.net/api/Function	D
<pre>2 ['"generatedBy": "Nicrosoft.NET.Sdk.Functions.Generator-4.4.0", 3 '"configurationSource": "attributes", 4 '"bindings": [5 ''-' "methods": [6 ''-' ""type": "httpTrigger", 7 ''- "methods": [8 ''-' ""get", 9 ''-' "methods": [8 ''-' "get", 9 ''-' "get", 10 ''-' "authlevel": "function", 11 ''-' "authlevel": "function", 12 ''-' "authlevel": "function", 13 ''-' authlevel": "function", 14 '], 15 ''disabled": false, 16 ''scriptFile": "/bin/DioDocsFileExportFunctionApp.dll", 17 ''entryPoint": "DioDocsFileExportFunctionI.Run" 18 }</pre>		
・ログ 目 App Insights ログ 、 アログレベル ×		
接続されました。現在の[コード + テスト]パネルで関数実行のログを表示しています。この関数のすべてのログを表示するには、[関数]メニュー		
	80B	

コピーした URL をブラウザに張り付けて、さらにクエリパラメータと文字列&name=DioDocs を追加します。

	副新	しいタブ	x +	-			Ø	×		
\leftarrow	C	۲	xcsfileexportfunctionapp20240726181328.azurewebsites.net/api/Function1?code=pCBRIDRXN0shizPV6mmdcQffXrGwcEhsNOShWNUW85SYAzFuxO_QDA%3D%30 &name=DioDocs	Ф	£≡		ゲスト 🙎			
		۲	https://diodocsfileexportfunctionapp20240726181328.azurewebsites.net/api/Function1?code=pCBRIDRXN0shizPV6mmdcQffXrGwcEhsNOShWNUW85SYAzFuxO_QDA							
			$https://diodocsfileexportfunctionapp20240726181328.azurewebsites.net/api/Function1?code=pCBRIDRXN0shizPV6mmdcQffXrGwcEhsNOShWNUW85SYAzFuxO_QDA%3D%3D&na$							
			ゲストとして閲覧しています							
ダストの閲覧データは、このデバイス上のプロファイルとは分けて 保存されます。ダストが閲覧したすべてのウィンドウを閉じると、 次のように処理されます。										
			Microsoft Edge では次の項目が保存されません.							
			 閲覧履歴 ダウンロード履歴 Cookie およびサイト データ 							
	Microsoft Edge では次の項目が保存されます:									
			 ダウンロードしたファイル 							

関数を実行するとデバッグ実行時と同じように、クエリパラメータで渡した文字列が追加された Excel ファイル がローカルに出力されます。関数 Function2 も同じ手順で確認できます。

さいごに

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

Azure 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 Azure Functions で「<u>DioDocs(ディオドック)</u>」を使用した C#(.NET 8)のクラ スライブラリをベースにした関数を作成し、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

実装する内容は今回も非常にシンプルです。Azure Functions アプリケーションで<u>HTTPトリガー</u>を使用する関数を作成します。関数の実行時に DioDocs を使用して Excel と PDF ファイルを作成し、クエリパラメータで受け取った文字列をそれぞれのファイルへ追加します。

前回は作成した Excel と PDF ファイルを <u>FileContentResult</u>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 Azure Blob Storage へ出力します。

アプリケーションを作成

以下のドキュメントを参考に Azure Functions アプリケーションを作成していきます。 クイック スタート:Visual Studio を使用して Azure で初めての関数を作成する

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



プロジェクト名 DioDocsFileIOFunctionApp 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				
Azure Functions C# Azure クラウド				
プロジェクト名()				
DioDocsFileIOFunctionApp				
場所(L)				
C:¥Users¥kuni¥Desktop -				
ソリューション名(M) ()				
DioDocsFileIOFunctionApp				
	戻る(<u>B</u>)	作	戓(<u>C</u>)	

Azure Functions で作成する関数のテンプレートを選択します。Http Trigger を選択して [作成] をクリック します。

		-		×
追加情報				
Azure Functions C# Azure クラウド				
Functions worker 🕦				
NET 8.0 (長期的なサポート) ▼				
Function 🛈				
Http trigger 🔹				
✔ ランタイム ストレージ アカウントに Azurite を使用する (AzureWebJobsStorage) 🛈				
コンテナーのサポートを有効にする (1)				
Authorization level 🕕				
Function •				
.NET Aspire オーケストレーションへの参加 🛈				
	戻る(B)	作	成(C)	

DioDocsFileIOFunctionApp プロジェクトが作成されます。



NuGet パッケージの追加

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



また、作成した Excel と PDF ファイルを Azure Blob Storage へ出力するために、<u>出力バインドで Azure Blob</u> Storage を使用するので、必要なパッケージ Microsoft.Azure.WebJobs.Extensions.Storage をインストー

ルします。



Azure Blob Storage を使うコードを追加

Azure Blob Storage の出力バインドを設定するコードを追加して Function1 を以下のように更新します。

```
public static class Function1
{
   [FunctionName("Function1")]
   public static async Task<IActionResult> Run(
       [HttpTrigger(AuthorizationLevel.Function, "get", "post", Route = null)] HttpRequest req,
       [Blob("output/result.xlsx", FileAccess.Write)] Stream outputfile,
       ILogger log)
   {
       log.LogInformation("C# HTTP trigger function processed a request.");
       string name = req.Query["name"];
       string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
       dynamic data = JsonConvert.DeserializeObject(requestBody);
       name = name ?? data?.name;
       string responseMessage = string.IsNullOrEmpty(name)
           ? "This HTTP triggered function executed successfully. Pass a name in the query string or in
the request body for a personalized response."
```

```
: $"Hello, {name}. This HTTP triggered function executed successfully.";
    return new OkObjectResult(responseMessage);
}
```

追加した 6 行目のコードですが、出力バインドの属性コンストラクタ[Blob("output/result.xlsx", FileAccess.Write)]では、出力先となる Azure Blob Storage のコンテナとファイル名 output/result.xlsx と、Azure Blob Storage への書き込みアクセス FileAccess.Write を設定しています。

DioDocs for Excel を使うコードを追加

```
DioDocs で Excel ファイルを作成するコードを追加して Function1 を以下のように更新します。
public static class Function1
{
   [FunctionName("Function1")]
   public static async Task<IActionResult> Run(
       [HttpTrigger(AuthorizationLevel.Function, "get", "post", Route = null)] HttpRequest req,
       [Blob("output/result.xlsx", FileAccess.Write)] Stream outputfile,
       ILogger log)
   {
       log.LogInformation("C# HTTP trigger function processed a request.");
       string name = req.Query["name"];
       string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
       dynamic data = JsonConvert.DeserializeObject(requestBody);
       name = name ?? data?.name;
       string Message = string.IsNullOrEmpty(name)
           ? "Hello, World!!"
          : $"Hello, {name}!!";
       Workbook workbook = new Workbook();
       workbook.Worksheets[0].Range["B2"].Value = Message;
       workbook.Save(outputfile);
       return new OkObjectResult("Finished.");
   }
}
```

DioDocs for PDF を使う関数を追加

ソリューションエクスプローラーから DioDocsFileIOFunctionApp プロジェクトを右クリックして[追加]-[新しい Azure 関数]を選択して、DioDocs で PDF ファイルを作成する関数 <mark>Function2</mark> を追加します。



新しい Azure 関数 - Function2	×
Service Bus Queue trigger	Authorization level
Http trigger	Function •
🔤 Http trigger with OpenAPI	
🕙 Timer trigger	
🚥 Queue trigger	
Blob trigger	
📧 Event Grid trigger	
📧 Event Hub trigger	
📧 IoT Hub trigger	
📧 Service Bus Topic trigger	
55 Durable Functions Orchestration	
🜌 Cosmos DB Trigger	
5 SendGrid	
▼7. ci in	
	追加(<u>d</u>) キャンセル(<u>C</u>)

Azure Blob Storage を使うコードを追加

Azure Blob Storage の出力バインドを設定するコードを追加して Function2 を以下のように更新します。 public static class Function2 { [FunctionName("Function2")] public static async Task<IActionResult> Run([HttpTrigger(AuthorizationLevel.Function, "get", "post", Route = null)] HttpRequest req, [Blob("output/result.pdf", FileAccess.Write)] Stream outputfile, ILogger log) { log.LogInformation("C# HTTP trigger function processed a request."); string name = req.Query["name"]; string requestBody = await new StreamReader(req.Body).ReadToEndAsync(); dynamic data = JsonConvert.DeserializeObject(requestBody); name = name ?? data?.name; string responseMessage = string.IsNullOrEmpty(name)

```
? "This HTTP triggered function executed successfully. Pass a name in the query string or in
the request body for a personalized response."
    : $"Hello, {name}. This HTTP triggered function executed successfully.";
    return new OkObjectResult(responseMessage);
  }
}
```

```
先程と同じように、追加した 6 行目のコードでは出力バインドの属性コンストラクタ
[Blob("output/result.pdf", FileAccess.Write)]では、出力先となる Azure Blob Storage のコンテナと
ファイル名 output/result.pdf と、Azure Blob Storage への書き込みアクセス FileAccess.Write を設定し
ています。
```

DioDocs for PDF を使うコードを追加

DioDocs で PDF ファイルを作成するコードを追加して Function2 を以下のように更新します。

```
public static class Function2
{
   [FunctionName("Function2")]
   public static async Task<IActionResult> Run(
       [HttpTrigger(AuthorizationLevel.Function, "get", "post", Route = null)] HttpRequest req,
       [Blob("output/result.pdf", FileAccess.Write)] Stream outputfile,
       ILogger log)
   {
       log.LogInformation("C# HTTP trigger function processed a request.");
       string name = req.Query["name"];
       string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
       dynamic data = JsonConvert.DeserializeObject(requestBody);
       name = name ?? data?.name;
       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));
```

```
doc.Save(outputfile, false);
    return new OkObjectResult("Finished.");
}
```

デバッグ実行で確認

今回は Azure Function アプリケーションの関数で作成する Excel と PDF ファイルの出力先として、関数の出力 バインドで設定している Azure Blob Storage を使用しています。ローカルでの確認には <u>Azurite エミュレータ</u> ーを使用します。<u>Microsoft Azure Storage Explorer</u>で [Blob Containers] を右クリックして表示されるメニュ ーから [BLOB コンテナーの作成] を選択します。



output というコンテナを作成します。



作成した Azure Functions アプリケーションをローカルでデバッグ実行して確認します。Visual Studio からデバッグ実行すると以下のコンソールが表示されます。



アプリケーションに含まれる関数の URL は http://localhost:7071/api/Function1、

http://localhost:7071/api/Function2 となっています。この URL にクエリパラメータと文字 列?name=DioDocs を追加して、それぞれの関数をブラウザで実行します。





Microsoft Azure Storage Emulator のコンテナーoutput に保存された Result.xlsx、Result.pdf を確認しま す。クエリパラメータで渡した文字列 DioDocs が表示されていれば成功です。

自動保存 💽 🎵 📙	२ ५ ऱ ऱ	Result.xlsx	- Excel	R	検索								Kuni Uji	ie 🍘	m –	٥	×
ファイル ホーム 挿入	ページ レイアウト 豊	対式 データ 校閲	表示 開発	ヘルプ											icì 共有	כאב ק	۰۲
A1	- I X 🗸	f_x Hello, Dio	locs!!														*
A B 1 Hello, Dio Docs!! 2 3 3 4 5 6 7 8 9 9	C D	E F	G	H I	J	K	L	M	N	0	P	Q	R	S	T	U	
11 Result.pdf - Adobe Acroba ファイル (P) 編集 (E) 表示(V) ホーム、 V)ール	t Reader DC (32-bit) 署名(S) ウインドウ(W) / Result odf	リレブ(<u>H</u>)													_	0	* ×
	Resultput			1 / 1	- - -	(H)	159% ¥				3				2		- 9 -
	Hello, Die	oDocs!!	U .				• 67C2			[₽ • ↓					C'A		^

Azure ヘデプロイ

作成した Azure Functions アプリケーションを Azure ヘデプロイして確認します。ソリューションエクスプロー ラーから DioDocsFileIOFunctionApp プロジェクトを右クリックして [発行] を選択します。



公開するターゲットは「Azure」を選択します。特定のターゲットは「Azure Function App (Windows)」を選択 します。

公開 現在公開している場所			×
ターゲット	۸	Azure アプリケーションを Microsoft クラウドに発行します	
		Docker Container Registry アプリケーションを Docker イメージに対応したサポートされている Container Registry に発行します	
		フォルダ ー アプリケーションをローカル フォルダーまたはファイル共有に発行します	
	G	プロファイルのインポート アプリを配置するための発行設定をインポートします	
		戻る(B) 次へ(N) 完了(F) キャンセル(C)

公開 アプリケーションをホストす	るためにど	の Azure サービスを使用しますか?	x
ターゲット 特定のターゲット		Azure Function App (Windows) 動的にスケーリングされ、コードをオンデマンドで実行するサーバーレス コンピューティングにアプリケーション コードを発行し: す	ŧ
	<i>«</i>	Azure Function App (Linux) 動的にスケーリングされ、コードをオンデマンドで実行するサーバーレス コンピューティングにアプリケーション コードを発行し; す	±
	Azure Function App コンテナー アプリケーションを Docker イメージとして Azure Container Registry に発行し、 Azure Function App で実行します		
	_	Azure コンテナー レジストリ アブリケーションを Docker イメージとして Azure Container Registry に発行します	
		戻る(B) 次へ(N) 完了(F) キャンセル(C))

アプリケーションの名前やリソースグループ、使用する Azure Storage などを設定して [作成] をクリックします。

Function App (Windows) _{新規作成}	X Microsoft account
名前(<u>N</u>)	
DioDocsFileIOFunctionApp20210629161753	
サブスクリプション名(<u>S</u>)	
Visual Studio Enterprise サブスクリプション - MPN	•
リソース グループ(G)	
diodocs-rg (Japan East)	✔ 新規(<u>N</u>)
プランの種類(<u>P)</u>	
消費	-
場所(L)	
Japan East	-
Azure Storage	
diodocs (Japan East)	✔ 新規(<u>N</u>)
エクスポート(<u>E</u>)	作成(<u>R</u>) キャンセル(<u>C</u>)

公開 ^{既存の Azure} 関数を	選択するか、新規作成します	X Microsoft account
ターゲット	サブスクリプション名(<u>S</u>)	
特定のターゲット	Visual Studio Enterprise サブスクリプション - MPN 表示(V)	*
Functions インスタンス	リソースの種類	•
	検索(<u>R</u>) DioDocsFilelOFunctionApp20210629161753	
	関数アプリ	+ ¢
	 ▲ ● 関数アプリ ▲ ◆ DioDocsFileIOFunctionApp20210629161753 (消費) ▶ ● デプロイスロット 	
	戻る(<u>B</u>)	次へ(N) 完了(F) キャンセル(C)

これで公開の準備が完了しました。Visual Studio で [発行] をクリックして作成した Azure Function アプリケ ーションを Azure ヘデプロイします。

 ✓ ファイルE 編集(E) ○ • ○ 1 谷 • ☆ 論 	表示(V) Git(G) プロジェクト(P) 🍟 り - ペ - Debug - Ar	ビルド(B) デパッグ(D) テスト(S) 分析(N) ツール(D) 拡張機能(ny CPU - ▶ DioDocsFilelOFunctionApp - 1 () () ⇒	8) ウィンドウ(W) ヘルプ(H) 検索 (Ctrl+Q)	👂 DioDocsFileIOFunctionApp 🛛 🥥 — 🗗 🗙 Live Share 🕅
DioDocsFileIOFunction 実施済みサービス 金属 サーバー 127ズローシー ツール減少9ス Cloud Explorer WS Explorer	App:公開 * × DioDocsFileOFunction Azure Function App (M + 新規 その他のアクション・ ① 公開準備が完了しました。 設定 構成 ターゲット ランタイム すべての設定を表示	App20210629161753 - Zip Deploy.pubxml ▼ Vindows) Release ▲ 移植可能 ▲	ং র্হার্য যে	
otet	ホスティング サブスクリプション リソース グループ リソース名 ユーザー名 パスフード サイト: https://diodocsfileiofu	c841263b-189e-4993-8eeb-2c27ce6cb92f		GitHub <mark>ソリューション Iク チーム Iクスプロー Git 変更 通知</mark> プロパティ マ サ ×
	サービスの依存開係 Application Insights		+ C 啓 … ▲ 場成 ~ …	
エラー一覧 出力	-		新加加 486 CTT CC 2) ・・・	
∠」準隔元」				

公開が完了するとメッセージが表示されます。



デプロイしたアプリケーションを確認

Azure Functions アプリケーションを実行する前に、ローカルでの確認と同じく関数の出力バインドで設定して いる Azure Blob Storage のコンテナーoutput を作成しておきます。

\equiv Microsoft Azure		0検索 (G+/)	D D	🖨 🕸 ? R
ダッシュボード > diodocs				
aliodocs コンテナー	x ···			
	+ ナデナ- 台 アクセスレベルを変更します り コンテナ	を復元する 🗸 🜔 更新 📔 📋 削除		
■ 概要	^ プレフィックスによるコンテナーの検索			削除されたコンテナーを表示する
 アクティビティ ログ タグ 	名前	最終変更日時	パプリック アクセス レベル	リース状態
♪ 問題の診断と解決	\$logs	2021/6/4 15:20:59	プライベート	利用可能
⁸ アクセス制御 (IAM)	azure-webjobs-hosts	2021/6/7 16:48:58	プライベート	利用可能
■ データ移行	azure-webjobs-secrets	2021/6/7 16:48:57	プライベート	利用可能
イベント	output	2021/6/29 15:04:14	プライベート	利用可能
🚡 Storage Explorer (プレビュー)				
データ ストレージ				
🔳 コンテナー				
🧃 ファイル共有				
■ † 1-				
🚍 テーブル				

Azure ポータルでデプロイしたアプリケーションに含まれる関数 Function1、Function2 が表示されます。

- ホーム > 関数アプリ >									
	Kite DioDocsFileIOFunc Ritterの Ritterの	ctionApp20240729155904 🖉 🖈							
	<mark>▶</mark> 検索 ◇ 《	○ 《 □ 参照 ○ 最新の情報に更新 □ 停止 ○ 再起動 二 スワップ ④ 発行プロファイルの取得 ○ 発行プロファイルのリセット ● アプリコンテンツのダウンロード … へ 要点							
	🎸 概要								
	📄 アクティビティ ログ	リソース グループ (: <u>diodocs-r</u> g	URL		tionapp20240729155904.azureweb				
	🏡 アクセス制御 (IAM)	状態 : 実行中	オペレーティング シス	: Windows					
		場所 (移動) : Japan East	App Service プラン	: JapanEastPlan (Y1: 0)					
	∂ 問題の診断と解決	サブスクリプション (: <u>Visual Studio Enterprise サブスクリプション - MPN</u>	ランタイム バージョン	: 4.834.3.22875					
	🏮 Microsoft Defender for Cloud	サブスクリプション ID :d575d13c-7bf8-4ce5-a30f-017be3a7127a							
	🗲 イベント (プレビュー)	タヴ (編集) : タグの追加							
	💡 より良い連携 (プレビュー)								
	🔤 ログ ストリーム	関数 メトリック プロパティ 通知(0)							
	> 関数	{} ローカル環境の設定 > 💍 最新の情報に更新							
	〉 デプロイメント								
	> パフォーマンス								
	〉 設定	名前	トリガー	状態	モニター				
	〉 App Service プラン		LITTO	⊘ 右动	I型が出したと				
	> 開発ツール		nur						
	> API		HTTP	🕑 有効					
	> 監視								
	> オートメーション								
	〉 サポート + トラブルシューティング								

関数 Function1 をクリックして「コードとテスト」を選択します。「関数の URL を取得」が表示されるので こちらをクリックします。



コピーした URL をブラウザに張り付けて、さらにクエリパラメータと文字列&name=DioDocs を追加します。



関数 Function1 を実行するとデバッグ実行時と同じように、クエリパラメータで渡した文字列が追加された Excel ファイル Result.xlsx が Azure Blob Storage のコンテナ output に出力されます。関数 Function2 も同 じ手順で確認できます。

🙏 output - Microsoft Azure 🛛 🗙	+								
C a portal.azure.com/#blade/Microsoft_Azure_Storage/ContainerMenuBlade/overview/storageAccountId/%2Fsubscriptions%2Fc841263b-189e-4993-8eeb-2c27ce6cb92f%2FresourceGroups%2Fd									
\equiv Microsoft Azure	シ リソース、サービス、ドキュメントの検索 (G+/)			D. 🖓 🗘	®? &				
output - ر جرد									
	← アップロード 台 アクセス レベルを変更します ご 更新 前 前除 ご 層の変更 ダ	リースの取得 🔗 リースの解約 👁 ス	ナップショットの表示 百じ	スナップショットの作成					
■ 概要	認証方法: アクセス キー (Azure AD のユーザー アカウントに切り替える)								
乃 問題の診断と解決	wayer: output								
🏡 アクセス制御 (IAM)	プレフィックスによる BLOB の検索 (大文字と小文字を区別)				tれた Blob を表示				
設定	⁺ マフルターの追加								
Shared Access Signature	名前	変更日時	アクセス層	BLOB の種類	サイズ				
📍 アクセス ポリシー	esult.pdf	2021/6/29 16:59:27	ホット (推定)	プロック BLOB	62.08 KiB				
🔢 วือパティ	📄 📄 result.xlsx	2021/6/29 16:58:09	ホット (推定)	プロック BLOB	6 KiB				
 メタデータ 									

さいごに

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

Azure 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 Azure Functions で「<u>DioDocs(ディオドック)</u>」を使用した C#(.NET 8)のクラスラ イブラリをベースにした関数を作成し、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 Azure 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";

É	動保存 💽 🕇	2 🖪 9	~ CI ~ -	;		Result.xlsx	.		Q		
ファ	^ッ イル ホー	ム 挿入	ページ レイ	イアウト 数	式 データ	校閲	表示 開発	き ヘルプ			
Ľ	▲ DATICIE ~ ※ ダ B I U		~ 11		1 • A^ A ≡ ≡ ≫ •			設新り返して全体を表示			
貼り			• III •	• 🖽 • 💁 • 🗛 •				😅 セルを結合	して中央捕		
クሁ	ップボード 15		フォン	\	Ŀ		配	置			
A1			•	X 🗸	fx ZA	」にちは、世	世界!				
	А	В	С	D	E	F	G	Н	I		
1	こんにちは	世界!									
2											
3											
4											
5											
6											
7											
8											
9											
10											

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)

DioDocs for Excel では FontsFolderPath プロパティで参照するフォルダを設定していない場合、Azure Functions アプリケーションの実行環境 (Windows)の C:¥Windows¥Fonts にインストールされている日本語フォントであれば Font プロパティで設定するだけです。

「PDF エクスポート時に使用されるフォントについて」を見る

例えば、Font プロパティで「メイリオ」を設定してワークシートを PDF 出力した場合、Azure Functions の実 行環境の C:¥Windows¥Fonts には meiryo.ttc と meiryob.ttc が含まれていますので、追加で設定する必要は 無くそのまま利用することができます。

```
K Diagnostic Console
                    × +
                                                                                                                           0
                                                                                                                        🖻 🌣 😕 🖬 🕜
    4
                   Your app is currently in read only mode because you are running from a package file. To make any changes, please update the content in your zip file and
                  WEBSITE_RUN_FROM_PACKAGE app setting
                 ... / Fonts 🛨 🛛 299 items 🖌 🏫
                                               ±∕0
                          hangalb.ttf
                                                                            2016/5/25 15:34:44
                                                                                              182 KB
                 ±∕0
                          marlett.ttf
                                                                            2016/7/16 22:18:43
                                                                                              27 KB
                 ±/0
                          heiryo.ttc
                                                                            2016/7/14 15:17:50
                                                                                               9339 KB
                 ±/0
                         h meiryob.ttc
                                                                            2016/7/14 15:17:50
                                                                                              9549 KB
                 ±/0
                          hicross.ttf
                                                                            2016/7/16 22:18:43
                                                                                              822 KB
                 1/0
                          hingliu.ttc
                                                                            2016/7/16 22:19:21
                                                                                              26802 KB
                 ±∕0
                                                                                               35929 KB
                          hingliub.tto
                                                                            2016/7/16 22:18:43
                 1/O
                          mmrtext.ttf
                                                                            2016/7/16 22:18:25
                                                                                              285 KB
                  ajalla.tt
                  ajallab.ttf
                  malgun.ttf
                    unbd.ttf
                  algunsl.ttf
                   ngal.ttf
                   galb.tt
                  marlett.ttf
                   irvob.tt
public static class Function4
{
    [FunctionName("Function4")]
    public static async Task<IActionResult> Run(
         [HttpTrigger(AuthorizationLevel.Function, "get", "post", Route = null)] HttpRequest req,
         ILogger log)
     {
         log.LogInformation("C# HTTP trigger function processed a request.");
          string name = req.Query["name"];
         string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
         dynamic data = JsonConvert.DeserializeObject(requestBody);
         name = name ?? data?.name;
```

```
string Message = string.IsNullOrEmpty(name)
       ? "こんにちは、世界!"
       : $"こんにちは、{name}!";
   Workbook workbook = new Workbook();
   workbook.Worksheets[0].Range["A1"].Font.Name = "メイリオ";
   workbook.Worksheets[0].Range["A1"].Value = Message;
   byte[] output;
   using (var ms = new MemoryStream())
   {
       workbook.Save(ms, SaveFileFormat.Pdf);
       output = ms.ToArray();
   }
   return new FileContentResult(output, "application/pdf")
   {
       FileDownloadName = "Result.pdf"
   };
}
```



「IPAex フォント」のようなアプリケーションの実行環境 (Windows)の C:¥Windows¥Fonts に含まれていない 日本語フォントを利用する場合は、日本語フォントをアプリケーションのコンテンツとして配置し、 FontsFolderPath プロパティで参照する方法があります。

まず、以下のように「IPAex ゴシック」のフォント ipaexg.ttf をアプリケーションのプロジェクトに追加しま す。



このフォントが含まれる fonts フォルダへのパスを FontsFolderPath プロパティに設定します。この設定により Font プロパティで設定した「IPAex ゴシック」が利用できます。

```
public static class Function4
```

```
{
    [FunctionName("Function4")]
    public static async Task<IActionResult> Run(
```

```
[HttpTrigger(AuthorizationLevel.Function, "get", "post", Route = null)] HttpRequest req,
Microsoft.Azure.WebJobs.ExecutionContext context,
```

```
ILogger log)
{
   log.LogInformation("C# HTTP trigger function processed a request.");
   string name = req.Query["name"];
   string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
   dynamic data = JsonConvert.DeserializeObject(requestBody);
   name = name ?? data?.name;
   string Message = string.IsNullOrEmpty(name)
       ? "こんにちは、世界!"
       : $"こんにちは、{name}!";
   Workbook workbook = new Workbook();
   Workbook.FontsFolderPath = Path.Combine(context.FunctionAppDirectory, "fonts");
   workbook.Worksheets[0].Range["A1"].Font.Name = "IPAex ゴシック";
   workbook.Worksheets[0].Range["A1"].Value = Message;
   byte[] output;
   using (var ms = new MemoryStream())
   {
       workbook.Save(ms, SaveFileFormat.Pdf);
       output = ms.ToArray();
   }
   return new FileContentResult(output, "application/pdf")
   {
       FileDownloadName = "Result.pdf"
   };
}
```

}

Result.pdf - Adobe Acrobat Reader DC (32-bit) ファイル (F) 編集 (F) 表示(V) 素合(S) ウィンドウ(W) ヘルプ(H)		- 0 ×
ホーム ツール Result.pdf ×		⑦ 4 ログイン
🖺 슈 🖶 🔍	⑦ ④ 1 / 1 ▶ ① ○ ⊕ 159% · ☐ ⊡ ☐ · □	<i>2</i> 🖂 🎝
こんにちは、世界!	X書のプロパゲイ × E要 世をユジティ ブネント カスタム 詳細設定 この文書で使用しているフォント E P MeesGobic (想め込みサブセット) 範囲: Thurbype (CD) エンコーディング : Identsy+4	

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「作成したワークシートを PDF 出力する際の日本語フォントを設定する (DioDocs for Excel)」と同じような動作になります。

Azure Functions アプリケーションの実行環境 (Windows)の C:¥Windows¥Fonts にインストールされている日 本語フォントであれば FontName プロパティで設定するだけです。

例えば、FontName プロパティで「メイリオ」を設定して PDF ドキュメントを作成した場合、Azure Functions の実行環境の C:¥Windows¥Fonts には meiryo.ttc と meiryob.ttc が含まれていますので、追加で設定する必 要は無くそのまま利用することができます。

```
public static class Function3
```

```
{
    [FunctionName("Function3")]
    public static async Task<IActionResult> Run(
        [HttpTrigger(AuthorizationLevel.Function, "get", "post", Route = null)] HttpRequest req,
        ILogger log)
    {
        log.LogInformation("C# HTTP trigger function processed a request.");
        string name = req.Query["name"];
        string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
        dynamic data = JsonConvert.DeserializeObject(requestBody);
    }
}
```

```
name = name ?? data?.name;
   string Message = string.IsNullOrEmpty(name)
       ? "こんにちは、世界!"
       : $"こんにちは、{name}!";
   GcPdfDocument doc = new GcPdfDocument();
   GcPdfGraphics g = doc.NewPage().Graphics;
   g.DrawString(Message,
       new TextFormat() { FontName = "メイリオ", FontSize = 12 },
       new PointF(72, 72));
   byte[] output;
   using (var ms = new MemoryStream())
   {
      doc.Save(ms, false);
      output = ms.ToArray();
   }
   return new FileContentResult(output, "application/pdf")
   {
       FileDownloadName = "Result.pdf"
   };
}
```

}

ホーム ツール Result.pdf ×		?	4	ログ	イン
中 日 Q	(↑) (↓) 1 / 1 🖡 (⊕) (−) (+) 159% - 📙 🐼 🛃 🛃 - 🐺		0		20
こんにちは、世界	xextDiff x xextDiff xextDiff xextDiff xextDiff				~

「IPAex フォント」のようなアプリケーションの実行環境 (Windows)の C:¥Windows¥Fonts に含まれていない 日本語フォントを利用する場合は、日本語フォントをアプリケーションのコンテンツとして配置し、Font プロ パティで参照する方法があります。

```
public static class Function3
```

```
{
```

```
[FunctionName("Function3")]
```

```
public static async Task<IActionResult> Run(
```

```
[HttpTrigger(AuthorizationLevel.Function, "get", "post", Route = null)] HttpRequest req,
Microsoft.Azure.WebJobs.ExecutionContext context,
```

```
ILogger log)
```

```
{
```

```
log.LogInformation("C# HTTP trigger function processed a request.");
```

```
string name = req.Query["name"];
```

```
string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
dynamic data = JsonConvert.DeserializeObject(requestBody);
name = name ?? data?.name;
```

```
string Message = string.IsNullOrEmpty(name)
```

```
? "こんにちは、世界!"
```

```
: $"こんにちは、{name}!";
```

```
GcPdfDocument doc = new GcPdfDocument();
```

```
GcPdfGraphics g = doc.NewPage().Graphics;
```

}

```
Font font = Font.FromFile(Path.Combine(context.FunctionAppDirectory, "fonts", "ipaexg.ttf"));
g.DrawString(Message,
    new TextFormat() { Font = font, FontSize = 12 },
    new PointF(72, 72));
byte[] output;
using (var ms = new MemoryStream())
{
    doc.Save(ms, false);
    output = ms.ToArray();
  }
return new FileContentResult(output, "application/pdf")
{
    FileDownloadName = "Result.pdf"
    };
}
```

💫 Result_pdf - Adobe Acrobat Reader DC (32-bit) ファイル (F) 編集 (E) 表示(V) 署名(S) ウィンドウ(W) ヘルプ(H)			-	0	×
ホーム ツール Result.pdf ×		?	.	ログ	ん
🖺 🗘 🖶 🔍 🕜	↓ 1 / 1 k (¹) () () () (159% · k (k) (k))		ð.	\bowtie	20
こんにちは、世界!	文書のプリパティ × 構成 でもよりディーブネント 力スタム 詳細設定 この文書で使用しているフォント ・ ・ ・ ・ ● Pearsonic (豊め込みサブゼット) 構成: TrueType (CD) エンコーディング: :denthy+H				

日本語フォントの配置方法は「作成したワークシートを PDF 出力する際の日本語フォントを設定する (DioDocs

for Excel)」と同じです。

本記事では Azure Functions で DioDocs を利用する際に、日本語フォントを使用する Tips を紹介しましたが、 Azure Functions の実行環境として Windows を選択した場合の Tips になります。Azure Functions の実行環境 として Linux を選択した場合については今後の記事で紹介したいと思います。

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