メシウス株式会社

AWS Lambda 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「AWS Lambda と DioDocs で Excel や PDF ファイルを出力する(1)~(3)」の連載 記事をベースに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。

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

目次

AWS Lambda と DioDocs で Excel や PDF ファイルを出力する(1)	2
AWS Lambda とは	2
実装する内容	2
AWS Toolkit for Visual Studio のセットアップ	2
AWS Lambda アプリケーションを作成	2
NuGet パッケージの追加	4
Amazon API Gateway を使うコードを追加	5
DioDocs for Excel を使うコードを追加	5
デバッグ実行で確認	6
AWS ヘデプロイ	7
トリガーの追加	10
API Gateway のバイナリメディアタイプを設定	11
デプロイしたアプリケーションを確認	
PDF を出力するには?	16
さいごに	17
AWS Lambda と DioDocs で Excel や PDF ファイルを出力する(2)	
実装する内容	
AWS Lambda アプリケーションを作成	
NuGet パッケージの追加	
Amazon S3 にバケットを作成	
Amazon API Gateway を使うコードを追加	21
DioDocs for Excel を使うコードを追加	21
デバッグ実行で確認	23
AWS ヘデプロイ	24
トリガーの追加	27
デプロイしたアプリケーションを確認	
PDF を出力するには?	32
さいごに	
AWS Lambda と DioDocs で Excel や PDF ファイルを出力する(3)	35
セルに追加するテキストの日本語フォント(DioDocs for Excel)	35
ワークシートを PDF 出力する際の日本語フォント(DioDocs for Excel)	36
PDF ドキュメントを保存する際の日本語フォント(DioDocs for PDF)	41

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

本記事では、AWS Lambda で「<u>DioDocs(ディオドック)</u>」を使用した C#(.NET 8)の Lambda 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。

AWS Lambda とは

AWS Lambda は <u>Amazon Web Services</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。

AWS Lambda は<u>NET 8 をサポート</u>しており、C#で .NET 8 ベースの Lambda 関数を作成できます。今回は <u>AWS</u> <u>Toolkit for Visual Studio</u>を使用して Visual Studio 2022 で Lambda 関数を作成し、AWS ヘデプロイして確認し てみます。

実装する内容

今回実装する内容は非常にシンプルです。AWS Lambda アプリケーションで <u>Amazon API Gateway</u>から HTTP リクエストを受け取る Lambda 関数を作成します。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 Amazon API Gateway に渡して 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。

AWS Toolkit for Visual Studio のセットアップ

Visual Studio 2022 への AWS Toolkit for Visual Studio のインストールと AWS の認証情報の設定は以下を参考 にして準備しておきます。 <u>AWS Toolkit for Visual Studio のインストールとセットアップ</u> AWS IAM 認証情報

AWS Lambda アプリケーションを作成

以下のドキュメントを参考に AWS Lambda アプリケーションを作成していきます。 基本 AWS Lambda プロジェクト

Visual Studio 2022 でプロジェクトテンプレート「AWS Lambda Project (.NET Core – C#)」を選択して [次へ] をクリックします。



プロジェクト名に「ExcelExportAWSLambda1」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				
AWS Lambda Project (.NET Core - C#) C# AWS クラウド Serverless				
プロジェクト名(J)				
ExcelExportAWSLambda1				
場所(L)				
C:¥Users¥kuni¥Desktop¥ •				
ソリューション名(M) 🚯				
ExcelExportAWSLambda1				
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¥ExcelExportAWSLambda1¥ExcelExportAWSLambda1¥" で作成され ます				
	戻る(B)	11	F成(C)	

AWS Lambda Project のテンプレートを選択します。「Empty Function」を選択して [Finish] をクリックします。

🎁 New AWS Lambda C# Project			-		×
Select Bluer	Orint hts of the C# project for your AWS L	ambda function.			
Blueprints are Lambda projects that con desired scenario and customize as need Filter:	tain getting started code for your functions ed.	and a test project. Choose a blueprint tha	it best align	s with your	
Empty Function	Empty Top-level Function	Native AOT (.NET 8)			
Setup the project and test project to create a Lambda function from scratch.	Lambda function using C# Top- level statements	A project configured for deployment using .NET 8's Native AOT feature.			
Custom	Custom, TopLevelStatements	NativeAOT			
Powertools for AWS Lambda	Custom Runtime Function	Simple Application Load Bal			- 11
Setup the project and test project to create a Lambda function with Powertools for AWS Lambda from scratch.	Use Lambda Custom Runtime feature to build Lambda functions using .NET 8.	A skeleton Lambda function to get started responding to Application Load Balancer events.			
Powertools, Custom	Custom	ALB, Simple			
Simple DynamoDB Function	Simple Kinesis Firehose Func	Simple Kinesis Function			
A skeleton Lambda function to get started responding to Amazon	A skeleton Lambda function to get started responding to Amazon	A skeleton Lambda function to get started responding to Amazon			
		Close Back	Next	Finish	

「ExcelExportAWSLambda1」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 Amazon API Gateway のイベントを処理するための パッケージ「Amazon.Lambda.APIGatewayEvents」と DioDocs for Excel のパッケージ 「DioDocs.Excel.ja」をインストールします。

00 7711/(F)	編集(E) 表示(V) Git(G)	プロジェクト(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)		rtAWSLambda1	(0) €	- • ×
		Debug Any CPU Mock Lambda lest lool •		_			_	ER
ー NuGet: Excel 参照 ス	lortAWSLambda1 🔋 X インストール済み 🔺	更新プログラム	NuGe	et パッケージ マネー	-ジャー: ExcelExportA	ہ ۔ AWSLambda1	 ションエクスプローラー ・ つ ・ つ ・ つ ・ つ ・ つ ・ の 	- 4 × • =• trl+:)
101 101 101 101 101 101 101 101 101 10	1+1) パッケージ (4) Amazon Lambda APIGat Amazon Lambda NET Core su Amazon Lambda NET Core su Amazon Lambda NET Core su DioDocs.Excelja 作成素: DioDocs.Excelja 作成素: DioDocs.Excelja 作成素: のを発展にサポートして必定め、 パッケージ (27)	P O D	2.7.0 2.2.0 2.4.1 ET Standard ET Standard ET Standard ET Standard		シャー: ExcelExport パッケージ ソース:	WSLambdaT : nugetorg • (\$	ソリューション エクスパローラー の検索 (C マ りコーション エクスパローラー マ Dys-Up1-2y2 T Legel Boot WASL T D	11-3
				r				
コンテナー エラ	ー一覧 出力 データツール操作							
C:¥Users¥kuni	¥Desktop¥ExcelExportAWSLam	bda1¥ExcelExportAWSLambda1¥ExcelExportAWSLambda1.csproj	を復元しました (2.07 秒)。				↑ ソース管理に追加 🔺	町 リボジトリの選択 🔺 📮

Amazon API Gateway を使うコードを追加

Lambda 関数が Amazon API Gateway から HTTP リクエストを受け取り、Lambda 関数から API Gateway へ HTTP レスポンスを返すために、以下のように FunctionHandler の引数と戻り値に APIGatewayProxyRequest と APIGatewayProxyResponse を設定します。

public APIGatewayProxyResponse FunctionHandler(APIGatewayProxyRequest input, ILambdaContext context)

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 FunctionHandler を以下のように更新します。

public APIGatewayProxyResponse FunctionHandler(APIGatewayProxyRequest input, ILambdaContext context)

```
{
```

APIGatewayProxyResponse response;

string ? queryString;

input.QueryStringParameters.TryGetValue("name", out queryString);

string Message = string.IsNullOrEmpty(queryString)

- ? "Hello, World!!"
- : \$"Hello, {queryString}!!";

//Workbook.SetLicenseKey("製品版またはトライアル版のライセンスキーを設定");

```
Workbook workbook = new Workbook();
```

```
workbook.Worksheets[0].Range["A1"].Value = Message;
var base64String = "";
using (var ms = new MemoryStream())
{
   workbook.Save(ms, SaveFileFormat.Xlsx);
   base64String = Convert.ToBase64String(ms.ToArray());
}
response = new APIGatewayProxyResponse
{
   StatusCode = (int)HttpStatusCode.OK,
   Body = base64String,
   IsBase64Encoded = true,
   Headers = new Dictionary<string, string> {
       {"Content-Type", "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet"},
       {"Content-Disposition", "attachment; filename=Result.xlsx"},
   }
};
return response;
```

DioDocs for Excel で作成した Excel ファイルを MemoryStream に保存し、これを一旦 base64 エンコードして います。これを文字列 base64String として APIGatewayProxyResponse の Body に設定して Amazon API Gateway に渡すようにしています。

作成した Excel ファイルを base64 にエンコードする理由ですが、Lambda 関数と Amazon API Gateway を連 携させる「AWS Lambda プロキシ統合」を利用する際の決まり事になっています。

<u>AWS Lambda プロキシ統合</u>から**バイナリメディアを返すには、Lambda 関数からのレスポンスを base64 でエン コードします**。また、API のバイナリメディアタイプを設定する必要があります。

https://docs.aws.amazon.com/ja_jp/apigateway/latest/developerguide/lambda-proxy-binary-media.html

デバッグ実行で確認

}

作成した Lambda 関数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。[F5] キーをクリックする と Mock Lambda Test Tool が起動します。

AWS .NET 8.0 Mock Lambda Ter	t +	- o ×
\leftrightarrow \rightarrow C (i) localhost:5050		🖈 🖸 1 Ø 🖬 🥹 i
 ► Test Function ☐ Executable Assembly ☐ Dead Letter Queue ☐ Documentation 	AWS .NET 8.0 Mock Lambda Test Tool Run .NET Lambda function code inside this tool. IDEs can attach their debuggers to t If you are developing .NET Lambda function using custom runtimes or C# top level Executable Assembly page should be used to test the function. Config File aws-lambda-tools-defaults.json AWS Credential Profile default Example Requests:	his tool and step through the Lambda code. statements that use the Amazon.Lambda.RuntimeSupport NuGet package the Function ExcelExportAWSLambda1:ExcelExportAWSLambda1.Function::FunctionHar v AWS Region ap-northeast-1
	Tip: If a Lambda function using the default serializer, Amazon Lambda Serialization.J LAMBDA_NET_SERIALIZER_DEBUG set to true the JSON input for the Lambda fu this tool to step through the code.	son, is deployed with the environment variable nction will be written to CloudWatch Logs. The captured JSON can then be used in

Example Requests に「API Gateway AWS Proxy」を設定して [Exceute Function] をクリックします。Response の body に base64 にエンコードされた文字列が格納されていれば OK です。

🗙 📄 AWS .NET 8.0 Mock Lambda Te: 🗙		
\leftrightarrow \rightarrow C (localhost:5050		★ 🎦 Ø 🛛 🥹 :
	ano tambaa toolo astaatogosti	Endersnydd i'r deanbad mendersnydd i'r deanbad m andionn anolon ra
Test Function	AWS Credential Profile	AWS Region
Executable Assembly	default	ap-northeast-1
E Dead Letter Queue	Example Requests: API Gateway AWS Proxy V	
Documentation	Function Input:	
	{ "body": "{\"test\":\"body\"}", "resource": "/{proxy+}", "path": "path/"(resource", "httpMethod": "POST", "queryStringParameters": { "foo": "bar" }, "pathParameters": { "proxy": "pathIto/resource" Trip: If a Lambda function using the default serializer, Amazon Lambda.Serialization.J LAMBDA_NET_SERIALIZER_DEBUG set to true the JSON input for the Lambda fur this tool to step through the code. Execute Function Save Request	son, is deployed with the environment variable nction will be written to CloudWatch Logs. The captured JSON can then be used in
	Response:	Log Output:
	The area below shows the result returned by your function execution.	The area below shows the logging calls in your code.
	{"statusCode":200,"headers":["Content-Type":"application/vnd.openxmlformats- officedocument.spreadsheetml.sheet","Content-Disposition":"attachment; filename=Result.xlsx"},"body":"UESDB0AAAAIABeolaFgfZBjttAAAAQBAAAIJbH9VLNJ1bH0E:	
	AWS .NET 8.0 Mock Lambda Test Tool (0.15.1) is an open source project. For issues	or feedback visit the AWS Lambda for .NET Core GitHub repository.

AWS ヘデプロイ

作成した Lambda 関数アプリケーションを AWS 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「ExcelExportAWSLambda1」プロジェクトを右クリックして「Publish to AWS Lambda」を選択します。



「Function Name」に DioDocsExcelExport を入力して [Next] をクリックします。

🔋 Upload to AWS La	nbda	-		
aws	Jpload Lambda Function hter the details about the function you want to upload.			
AWS Credentials:	Profile:default Tradicity Region: Asia Pacific (Tokyo) Tradicity Asia Pacific (Tokyo) Tradici			
Package Type:	Zip			
Lambda Runtime:	.NET 8			
Architecture:	• x86 O ARM			
Function Name:	Create new function DioDocsExcelExport Re-deploy to existing			
	▼			1
Handler:	ExcelExportAWSLambda1::ExcelExportAWSLambda1.Function::FunctionHandler For .NET runtimes, the Lambda handler format is: <assembly>::<type>::<method></method></type></assembly>]
Description:]
Configuration:	Release		Ŧ	
✓ Save settings to a	aws-lambda-tools-defaults.json for future deployments.			
	Close Back Nex	ct	Upload	.::)

「Role Name」に New role based on AWS managed policy: AWSLambda_FullAccess を設定して [Upload] をクリックします。

ij Upload to AWS Lambda				-		×
Advanced Function	on Details gs for your funct	ion.				
Permissions						
Select an IAM role to provide AWS credentials to o	ur Lambda function	n allowing access to AWS Ser	vices like S3.			
Role Name: New role based on AWS managed p	olicy: AWSLambda_	FullAccess				-
Execution Memory (MB): 256 Timeout (Secs): 30 (1 - 900)	Debugging and E DLQ Resource:	rror Handling (no dead letter queue) tracing (AWS X-Ray)	1 More.			•
VPC If your function accesses resources in a VPC, select the list of subnets and security group IDs (these	Environment KMS Key: Variable	(default) aws/lambda				•
VPC Subnets:						
					,	Add
		Close	Back	Next	Upload	

成功すると以下の画面が表示されます。

	8) デパッグ(D) テスト(S) 分析(N) ツール(T) 拡張機能(X) ウィンドウ(W) ヘJ ◆ Mock Lambda Test Tool ◆ ▷ グ ~ 節 罰 。	ルプ(H) ク 検索 ・ ExcelExportAWSLambda1	× □ - ♥ 🐌 ≶ ≌
Constantion: DioDoesSacellesport	Last Update Status: Soccessful Last Modified: 2020/03/07 944128	Code Size: 46,253,679 bytes	 > yyu-yyu 70,70-5- ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
Log output			
□ #備充了			↑ ソース管理に追加 🔺 🔟 リボジトリの選択 🔺 🚨

AWS のコンソールで AWS Lambda の「関数」を選択するとデプロイした Lambda 関数「DioDocsExcelExport」 が表示されます。



トリガーの追加

デプロイした Lambda 関数「DioDocsExcelExport」をクリックして以下の画面から [トリガーを追加] をクリ ックします。

 DioDocsExcelExport - Lambda × + 		- o ×
← → C 😄 ap-northeast-1.console.aws.amazo	n.com/lambda/home?region=ap-northeast-1#/functions/DioDocsExcelExport?tab=coc	de 🛧 🖸 Ø 💐 🖬 🥹 :
dWS 詳サービス Q 検索 Namba 図 S3 図 Amazon CodeCatalyst 房 AWS Ay	[Alt+S] pp Runner 🛛 🔣 AWS Amplify 🛛 📆 Amazon CodeWhisperer	込
■ Lambda > 関数 > DioDocsExcelExport		٥
DioDocsExcelExport		スロットリング ゴ ARN をコビー アクション ▼
▼ 関数の概要 情報		Application Composer にエクスボート ダウンロード ▼
ダイアグラム テンプレート + トリガーを追加	DioDocsExcelExport ② Layers (0) 十 送信先を	説明 - 最終更新 2 分前 閲数の ARN 『 arrawstambda:ap-northeast-1:311330428996:function:Di oDocsExcelExport 閲数 URL 情報 -
コード テスト モニタリング	設定 エイリアス バージョン	
コードソース 情報		アップロード元 ▼
① Lambda 関数「DioDocsExcelExport.	」のデブロイバッケージが大きすぎて、インラインコード編集を有効にできません。カ	ただし、関数を呼び出すことはできます。
)、CloudShell フィードバック		© 2024, Amazon Web Services, Inc. またはその間連会社。 ブライパシー 用語 Cookie の設定 🗸

「API Gateway」を選択し、さらに「Create a new API」を選択します。作成する API タイプは「REST API」を 選択して、セキュリティは「Open」を選択します。この状態で [追加] をクリックします。

– –	トリガーの追加 - Lambda × +									×
	C ap-northeast-1.console.aws.amazon.com/lambda/	ome?region=ap-northeast-1#/add/re	lation?focus=aws%2Flaml	bda⌖=arn%3Aaws%3Ala	mbda%3Aa	p-northeast-1%3A	*	บิ ø	⊒ / 🔲 🖞	🐵 i
aws	Ⅲ サービス Q 検索	[Alt+S]		G	0 ¢	0 0	東京 ▼	ukgraphics @	3113-3042-89	96 🔻 🕯
📐 Lamb	la 🔞 53 🧕 Amazon CodeCatalyst 🧾 AWS App Runner 📕 A	WS Amplify 🛛 🐻 Amazon CodeWhisperer								
=				_						(
	トリガーの設定 情報									9
										Ť
	API Gateway aws api application-services backend HTTP RES	r serverless	•							
	Add an API to your Lambda function to create an HTTP en types of RESTful APIs: HTTP APIs and REST APIs. Learn mo	Ipoint that invokes your function. AP	Gateway supports two							
	Intent Use an existing api or have us create one for you.									
	Create a new API									
	 Use existing API 									
	API type									
	HTTP API Build low-latency and cost-effective REST APIs with built- in features such as OIDC and OAuth2, and native CORS support.	 REST API Develop a REST API where you g over the request and response al management capabilities. 	ain complete control ong with API							
	Security Configure the security mechanism for your API endpoint.									
	Open	•								
	Additional settings									
	Lambda will add the necessary permissions for Amazon AF Learn more 🖸 about the Lambda permissions model.	l Gateway to invoke your Lambda fur	action from this trigger.							
			キャンセル 追加							
E Cloud	hell フィードパック			© 2024, Amazon We	b Services, Inc	1. またはその関連会社	t . プ ラ	ライバシー 用	語 Cookie	の設定

以下のようにトリガーに API Gateway が追加されます。

👻 🕺 DioDocsExcelExport - Lambda 🛛 🗙	
← → ♂ 😁 ap-northeast-1.c	onsole aws.amazon.com/lambda/home?region=ap-northeast-1#/functions/DioDocsExcelExport?tab=configure 🖈 🗅 🥥 🗊 🖬 🧕
aws III サービス Q 検索	[Alt+S]
🔀 Lambda 🛛 🖸 S3 🔯 Amazon CodeCa	stalyst 📝 AWS App Runner 🔀 AWS Amplify 🥫 Amazon CodeWhisperer
■ 関数の概要 情報	Application Composer にエクスポート ダウンロード ▼ 0
ダイアグラム デ	ジプレート DioDocsExcelExport 会 Layers (0) 中送信先を追加 開放のARN つ arruavslambda:ap-northeast-1:311330428996:function:Di oDocsExcelExport 開放 URL 情報 -
コード テ スト	モニタリング 設定 エイリアス バージョン
	トリガー (1) mm C エラーを修正 編集 所除 トリガーを追加
F02-	Q トリガーを検索 〈 1 〉
アクセス権限	► トリガー
送信先	API Gateway: DioDocsExcelExport-API
関数 URL	arraws:execute-apicap-northeast-1311330428996.opjeiav231/*//DioDocsExceExport API endnoint: https://opieiav231.execute-apicap-northeast-1_amazonaws.com/default/DioDocsExceExport
環境変数	
タヴ	
CloudShell フィードバック	© 2024, Amazon Web Services, Inc. またはその関連会社。 プライバシー 用語 Cookie の設定

API Gateway のバイナリメディアタイプを設定

AWS のコンソールで作成した API「DioDocsExcelExport-API」の「API の設定」から「メディアタイプの管理」 をクリックします。



「バイナリメディアタイプを追加」をクリックして「*/*」を追加します。追加後に[変更を保存]をクリックし ます。

•	Diol	DocsE:	xcelEx	oort -	Lambda	×		API G	ateway	- パイナ	リメディフ	94 >	K H	+																							-	o	×	
		Э	••	ap-n	orthea	it-1.co	onsole	.aws.a	mazor	.com/	apigat	eway/n	main/a	apis/o	pjeiav		api-de	tail/me	dia-ty	oes-ed	it?api	opjeia	av2318	®io	on=ap-	-north	east-1							☆	ប		I] 😔) :	
aws		. . .	ービス	•	2、検?	ą.											Į,	Alt+S]										۵.	¢	Q	>	0	東京		ukgra	phics (ត្ថ 3113	-3042-8	996 🔻	
<u> N</u> Lan	nbda	•	S3	🖸 An	nazon (odeCa	talyst	1	WS Ap	p Runn	er 🚺	AWS /	Amplif	iy 💽	💈 Ama	izon Ci	odeWh	isperer																						
=	<u>A</u>	PI Gar	^{tewa}	。 -リ	<u>арі</u> ў) <u>Dic</u> ディ	って	ExcelE クタ・	_{(port-}	^{API (0}	_{pjeiav} 管理	2 <u>31)</u>) 里	> <u>AP</u>	1の設		・ バー	イナリ	メディ	アタイ	プを管	理																		6)
		パー APLO Type	イナ のため 1 ヘッ	リメ にバー ダーと	ディ チリサ Accept	アタ ポート HTTP	イプ を設定 ヘッダ	するに ーをチ	は、AP エック	1 ガゾ(-1 して、!	(ナリデ リクエス	ータと ト本文	して扱: の処理	うメデ 方法を	イアタ 決定し	イプる /ます。	:入力L	ます。	API Gati	way (J	Conte	ıt-																		
		۲۲-۱ ۲/۲	イナリ 、 バイ:	メデ. ・リメ	(アタ ディ)	イプ クタイ	ブを述	当力口]									i	削除]																				
																	:	キャン	セル		変更る	保存																		
R. Co	udShel		71-	× <i>1</i> (۵	0																				© 20)24. Am	nazon V	Veb Se	rvices. I	nc. また	1474	の関連会	·21.	7=	イバシー	_	用語	Cookie	の設定	

「*/*」を設定する理由ですが、 Lambda 関数と Amazon API Gateway を連携させる「AWS Lambda プロキシ 統合」を利用する際の決まり事になっています。

この統合例でウェブブラウザを使用して API を呼び出すには、API のバイナリメディアタイプを */* に設定しま す。API Gateway は、クライアントからの最初の Accept ヘッダーを使用して、レスポンスがバイナリメディアを 返すかどうかを判断します。ブラウザからのリクエストなど、Accept ヘッダー値の順序を制御できない場合に、バ イナリメディアを返すには、API のバイナリメディアタイプを */* (すべてのコンテンツタイプ) に設定します。

https://docs.aws.amazon.com/ja_jp/apigateway/latest/developerguide/lambda-proxy-binary-media.html

この設定ですが、気を利かせたつもりで application/vnd.openxmlformats-

<mark>officedocument.spreadsheetml.sheet</mark> など、固有のバイナリメディアタイプを設定してしまうと base64 エ ンコードされたただの文字列が出力されてしまうので注意が必要です。

デプロイしたアプリケーションを確認

AWS のコンソールで作成した API「DioDocsExcelExport-API」の「リソース」から「API のデプロイ」を選択 します。



デプロイされるステージは「default」を選択して [デプロイ] をクリックします。

💌 💹 DioDocsExcelExport - Lambda 🛛 🗙	M API Gateway - リソース	× +		- o ×
← → C 😁 ap-northeast-1.consc	ole.aws.amazon.com/apigatewa	ay/main/apis/opjeiav231/resources?api=opjeiav231®ion=ap-northeast-1		🖈 🆸 🛛 🖉 ፤
👑 🗰 9-63 - Q. 113		[A10+5]		R.R. +
		Deploy API × API がデプロイされるステージを選択します。例えば、API のテスト版を beta という名前のステージにデプロイできます。 ステージ Zテージ default ▼ デプロイメントの説明		
		キャンセル デブロイ	xy+FL2152	
🔁 Gourácal) - Za – PA (v Ž				

API「DioDocsExcelExport-API」の「ステージ」から「default - / - /DioDocsExcelExport - GET」を選択しま す。



「URLの呼び出し」に表示されている APIの URL をコピーして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=DioDocsForExcel」を追加します。



この API を実行するとクエリパラメータで渡した文字列「DioDocsForExcel」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。

 ● 新しいタブ × + 					×
				(@ 77F	:
		最近のダウンロード履歴	\otimes		
		 ダウンロードしたファイルは、このデバイスを るすべてのユーザーが確認できます 	使用す		
	ゲスト モードでブラウジング中	Result.xlsx S.8 K8・完了 C			
	このウィンドウで開いたページはブラウザの履歴に記録されませ ん。また、聞いているゲスト ウィンドウをすべて閉じると	すべてのダウンロード履歴	ß		
	Cookie などのデータはパソコンから消去されます。ただし、ダウンロードしたファイルは保持されます。				

红 自動保存	(TR)	5-9-	v ⊽ Res	ult.xlsx • この	PC に保存済	<i></i> ж ∨	P	検索									-	0 -	ø >	×
ファイル ホーム	挿入 ペ	ージ レイアウト	数式 テ	-9 校閲	表示 自	動化 開発	ヘルプ											-4<%	13 共有	~
A1	• : × ~	fx Hello	, DioDocs	ForExcel!!																\sim
A	В	С	D	E	F	G	н	1.1	J	К	L	М	N	0	Р	Q	R	S	Т	
1 Hello, Dio	DocsForEx	cel!!																		-L
2																				-L
4																				÷Ŀ.
5																				н
6																				н
7																				н
8																				
9																				1
10																				÷Ŀ
12																				÷L.
13																				н
14																				ъ
15																				1
16																				1
17																				-L
18																				÷
20																				÷L.
21																				÷.
22																				ъ
23																				н
24																				
25																				1
26																				1
28																				1
29																				
30																				1
31																				U
32																				
< >	Sheet1	Evaluati	ion Warnin	g H	F							: .		_		_			-	•
準備完了 🐻	家 アクセシビ	リティ: 検討がる	必要です													=			- + 10	0%

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 「DioDocs.Pdf.ja」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 FunctionHandler を以下のように更新します。PDF ファイルを出力するのでそれに合わせて APIGatewayProxyResponse の Header の内容も変更しています。

```
new PointF(72, 72));
var base64String = "";
using (var ms = new MemoryStream())
{
   doc.Save(ms, false);
   base64String = Convert.ToBase64String(ms.ToArray());
}
response = new APIGatewayProxyResponse
{
   StatusCode = (int)HttpStatusCode.OK,
   Body = base64String,
   IsBase64Encoded = true,
   Headers = new Dictionary<string, string> {
       {"Content-Type", "application/pdf"},
       {"Content-Disposition", "attachment; filename=Result.pdf"},
   }
};
return response;
```

さいごに

}

動作を確認できる AWS Lambda アプリケーションのサンプルはこちらです。 https://github.com/MESCIUSJP/ExcelExportAWSLambda1 https://github.com/MESCIUSJP/PDFExportAWSLambda1

AWS Lambda 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 AWS Lambda で「<u>DioDocs(ディオドック)</u>」を使用した C#(.NET 8)の Lambda 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 AWS Lambda アプリケーションで <u>Amazon API Gateway</u>から HTTP リクエストを受け取る Lambda 関数を作成します。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 Amazon API Gateway に渡して 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** <u>Amazon S3</u> へ出力します。 Amazon S3 への保存には AWS SDK for .NET を使用します。

AWS Lambda アプリケーションを作成

以下のドキュメントを参考に AWS Lambda アプリケーションを作成していきます。 基本 AWS Lambda プロジェクト - AWS Toolkit for Visual Studio

Visual Studio 2022 でプロジェクトテンプレート「AWS Lambda Project (.NET Core – C#)」を選択して [次へ] をクリックします。



プロジェクト名に「ExcelExportAWSLambda2」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			
AWS Lambda Project (.NET Core - C#) C# AWS クラウド Serverless			
プロジェクト名(J)			
ExcelExportAWSLambda2			
場所(L)			
C:¥Users¥kuni¥Desktop •			
ソリューション名(M) 🕕			
ExcelExportAWSLambda2			
プロジェクト は "C:¥Users¥kuni¥Desktop¥ExcelExportAWSLambda2¥ExcelExportAWSLambda2¥" で作成され ます			
戻る(B)	ſſ	F成(C)	

AWS Lambda Project のテンプレートを選択します。「Empty Function」を選択して [Finish] をクリックします。



「ExcelExportAWSLambda2」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 Amazon API Gateway のイベントを処理するための パッケージ「Amazon.Lambda.APIGatewayEvents」と Amazon S3 を使うためのパッケージ「AWSSDK.S3」、そ して DioDocs for Excel のパッケージ「GrapeCity.DioDocs.Excel.ja」をインストールします。



Amazon S3 にバケットを作成

Amazon S3 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-file-export]を、

Visual Studioの「AWS Explorer」から作成します。



Amazon API Gateway を使うコードを追加

Lambda 関数が Amazon API Gateway から HTTP リクエストを受け取り、Lambda 関数から Amazon API Gateway へ HTTP レスポンスを返すために、以下のように FunctionHandler の引数と戻り値に APIGatewayProxyRequest と APIGatewayProxyResponse を設定します。

DioDocs for Excel を使うコードを追加

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

{

APIGatewayProxyResponse response;

```
try
{
    // クエリ文字列を取得
    string? queryString;
    input.QueryStringParameters.TryGetValue("name", out queryString);
```

```
// ワークシートに追加するテキスト
```

string Message = string.IsNullOrEmpty(queryString)

- ? "Hello, World!!"
- : \$"Hello, {queryString}!!";

//Workbook.SetLicenseKey("製品版またはトライアル版のライセンスキーを設定");

```
Workbook workbook = new Workbook();
workbook.Worksheets[0].Range["A1"].Value = Message;
```

```
using (var ms = new MemoryStream())
```

```
{
```

{

```
workbook.Save(ms, SaveFileFormat.Xlsx);
```

```
// S3にアップロード
```

```
AmazonS3Client client = new AmazonS3Client(RegionEndpoint.APNortheast1);
```

```
var request = new PutObjectRequest
```

```
BucketName = "diodocs-file-export",
```

```
Key = "Result.xlsx",
InputStream = ms
```

};

```
await client.PutObjectAsync(request);
```

```
}
```

```
catch (Exception e)
```

}

```
{
    response = new APIGatewayProxyResponse
    {
        StatusCode = (int)HttpStatusCode.InternalServerError,
        Body = e.Message,
        Headers = new Dictionary<string, string> {
            { "Content-Type", "text/plain" }
        }
    };
}
return response;
}
```

DioDocs for Excel で作成した Excel ファイルを MemoryStream に保存し、これを AmazonS3Client クラスの PutObjectAsync メソッドを使用して Amazon S3 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 Lambda 関数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。[F5] キーをクリックする と Mock Lambda Test Tool が起動します。

AWS .NET 8.0 Mock Lambda Ter											
→ C	6	\$8 ★ D Ø ₹ D €									
Test Function Executable Assembly Dead Letter Queue	AWS .NET 8.0 Mock Lambda Test Tool Run .NET Lambda function code inside this tool. IDEs can attach their debuggers to this tool and step through the Lambda code. If you are developing .NET Lambda function using custom runtimes or C# top level statements that use the Amazon.Lambda.RuntimeSupport NuGet package the Executable Assembly page should be used to test the function.										
Documentation	Config File	Function									
	aws-lambda-tools-defaults.json	ExcelExportAWSLambda2::ExcelExportAWSLambda2.Function::FunctionHar									
	AWS Credential Profile	AWS Region									
	default	~ ap-northeast-1									
	Example Requests: select a request	v									
	Function Input:										
	JSON document as input to Lambda Function. Plain strings must	be wrapped in quotes.									
	Tip: If a Lambda function using the default serializer, Amazon.Lam LAMBDA_NET_SERIALIZER_DEBUG set to true the JSON input this tool to step through the code.	bda.Serialization.Json, is deployed with the environment variable for the Lambda function will be written to CloudWatch Logs. The captured JSON can then be used in									

Example Requests に「API Gateway AWS Proxy」を設定して [Exceute Function] をクリックします。Response で以下のように body に「ファイルが保存されました。」が表示されていれば OK です。

AWS .NET 6.0 Mock Lambda Te: X	+	- o ×
\leftrightarrow \rightarrow C (i) localhost:5050		■ ☆ む ■ □ 🧐 :
 Test Function Executable Assembly Dead Letter Queue Documentation 	Config File aws-lambda-tools-defaults.json AWS Credential Profile default Example Requests: [API Gateway AWS Proxy Function Input: { "body: "{\test":\"body\"}", "resource"; "{(proxy+)", "path*: "ipath/to/resource", "intplMethod": POST", "queryStingParameters": { "foo": "bar" }, "pathParameters": { "foor: "bar" }, "proxy: "path/to/resource", Tip: If a Lambda function using the default serializer, Amazon.Lambda.Serialization. LAMBDA_PTE_SERIALIZER_DEBUG set to true the JSON input for the Lambda function to step through the code. Execute Function Swe Request	Function ExcelExportAWSLambda2::ExcelExportAWSLambda2.Function::FunctionHar AWS Region ap-northeast-1 Json, is deployed with the environment variable unction will be written to CloudWatch Logs. The captured JSON can then be used in
	Response: The area below shows the result returned by your function execution.	Log Output: The area below shows the logging calls in your code.
	{"statusCode":200,"headers":{"Content-Type":"text/plain; charset-utf-&"},"body":"ファ イルが保存されました。","isBase64Encoded":false)	

AWS Explorer から「diodocs-file-export」をクリックして、DioDocs for Excel で作成した Excel ファイル 「Result.xlsx」がアップロードされているか確認できます。なお、AWS ヘデプロイした後にも確認するのでアッ プロードした Excel ファイルはここで一旦削除しておきます。

00 ファイル(F) 編集(E) 表示(V) Git(G) プロジェク ※ ⑧ ▼ ◎ 鋼 ▼ 醫 雪 ♡ ▼ ◯ ▼ Debug	 (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 Any CPU Mock Lambda Test Tool ○ 	٤(X) ウィンドウ(W) ヘルブ(H) │ ♀ 検索 •	ExcelExportAWSLambda2	() (× ت – جم ک
AWS Explorer - 4 ×	S3 Bucket: diodocs-file-export 🛥 🗙 Function.cs		- ¢	ソリューション エクスプローラー	- 4 × §
Credentials: Profile:default * *8 & ×	🖹 Upload File 🛛 🚱 Upload Folder 🎣 Create Folder 🛛 🖓 Refresh			Jo•≒⊟© 💊• 🖋 🛋	fту-л
Region: Asia Pacific (Tokyo) T	😚 diodocs-file-export		Fetched 1 Items (Cancel)	ソリューション エクスプローラー の検索 (Ctrl+:)	- م
	Filter:			マリューション 'ExcelExportAWSLambda	a2' (1/1 のプロジェクト)
Amazon CloudWatch	Name Size La	st Modified Date Storage Class		 ExcelExportAWSLambda2 	
Amazon CodeArtifact	★			P Properties	
Amazon DynamoDB	Result.xlsx 6,234 bytes 202	24/03/07 15:54:03 STANDARD		aws-lambda-tools-defaults is	on
G ^I Amazon EC2				C# Function.cs	
Amazon Elastic Container Service				Readme.md	
Go Amazon RDS					
Amazon S3					
Amaran SimpleDR					
Amazon SNS					
Amazon SQS					
Amazon VPC					
Generation					
AWS Elastic Beanstalk					
AWS Identity and Access Management					
AWS Lambda				ソリューション エクスプローラー Git 変更 通知	
				ブロパティ	
					•
				8 9 P	
	100 % *				
	⊖Clear		A		
	Title Status	Progress			
ツールボックス エンドポイント エクスプローラー AWS Explorer					
コンテナー エラー一覧 出力 データ ツール操作					
□ 半備完了				挿入 个 ソース管理に追加 🔺 🗉	リボジトリの選択 🔺 🚨

AWS ヘデプロイ

作成した Lambda 関数 アプリケーションを AWS 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「ExcelExportAWSLambda2」プロジェクトを右クリックして「Publish to AWS Lambda」を選択します。



「Function Name」に DioDocsExcelExportToS3 を入力して [Next] をクリックします。

🗊 Upload to AWS La	mbda	-		×
aws	Jpload Lambda Function nter the details about the function you want to upload.			
AWS Credentials:	Profile:default Region: Asia Pacific (Tokyo)			
Package Type:	Zip			
Lambda Runtime:	.NET 8			
Architecture:	● x86 ○ ARM			
Function Name:	 Create new function DioDocsExcelExportToS3 Re-deploy to existing 			
	▼			
Handler:	$\label{eq:excel} Excel {\tt Excel Export AWS Lambda 2. Function :: Function Handler} \\$			
Description:	For .NET runtimes, the Lambda handler format is: <assembly>::<type>::<method></method></type></assembly>			
Configuration:	Release Framework: net8.0			•
Save settings to	aws-lambda-tools-defaults.json for future deployments.			
	Close Back Ne	xt	Upload	

「Role Name」に New role based on AWS managed policy: AWSLambdaExecute を設定して [Upload] を クリックします。

ij Upload to AWS Lambda						_		×
Advanced Function	on Details gs for your funct	tion.						
Permissions								
Select an IAM role to provide AWS credentials to o	ur Lambda function	n allowing access to	o AWS Service	s like S3				
Role Name: New role based on AWS managed p	olicy: AWSLambda	Execute						-
Execution Memory (MB): 512 Timeout (Secs): 30 (1 - 900)	Debugging and DLQ Resource:	Error Handling <no dead="" letter<br="">e tracing (AWS X-R</no>	queue> ay) <u>Learn Mc</u>	ore.				Ŧ
VPC	KMS Kev:	(default) aws/lan	nbda					-
If your function accesses resources in a VPC, select the list of subnets and security group IDs (these must belong to the same VPC).	Variable		Value					
VPC Subnets:								
Security Groups:								
								Add
			Close		Back	Next	Uploa	ad

成功すると以下の画面が表示されます。

00 771ル(F) ### ∭ ⊕ • ⊚ ∰ • ⊑	集(E) 表示(V) Git(G) プロジェクト(P) ビルド(B) 5 多 🔄 🗐 🏷 × 🤆 × │ Debug ▼ │ Any CPU	デバッグ(D) テスト(S) 分析(N) ツール(T • Mock Lambda Test Tool • D ()) 拡張機能(X) ウィンドウ(W) ヘルプ(H) ダー 即 罰 -		× □ - ♥ Ø ≶ ≥
Pendior: Die. xd Appl: Change Function: Die State: Active Runtime: dotte Test Function: Configuration AvyS x: Ray Logs	elExpontOS • × s	Last Update Status: Successful Last Modified: 2024/03/07 160	D1:51 Response ISON Pretty Print	Code Size: 48,103.925 bytes	ソリューション ビススパローラ・ ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
<i>דקעב</i> 19	Log output				
4 #備完了					个 ソース管理に追加 🔺 🗐 リボジトリの選択 🔺 📮

AWS のコンソールで AWS Lambda の「関数」を選択するとデプロイした Lambda 関数「DioDocsExcelExportToS3」 が表示されます。



トリガーの追加

デプロイした Lambda 関数「DioDocsExcelExportToS3」をクリックして以下の画面から [トリガーを追加] を クリックします。

DioDocsExcelExportToS3 - Lam +		- o ×
← → C 😅 ap-northeast-1.console.aws.amaze	on.com/lambda/home?region=ap-northeast-1#/functions/DioDocsExcelExport1	ToS3?tab=code 🗴 🖸 🗊 🖬 🥹 🗄
aWS 詳サービス Q 検索 国 Lambda 図 S3 Ø Amazon CodeCatalyst 🗾 AWS A	[Alt+S] pp Runner 🛛 AWS Amplify 🥃 Amazon CodeWhisperer	② ◆ ⑦ ④ 東京 ▼ ukgraphics @ 3113-3042-8996 ▼
■ Lambda > 関数 > DioDocsExcelExportTo	153	٥
DioDocsExcelExportTo	53	スロットリング □ ARN をコピー アクション ▼
▼ 関数の概要 (##		Application Composer にエクスポート ダウンロード ▼
<u>タイアグラム</u> テンプレート + トリガーを追加	DioDocsExcelExportToS3	説明 - 最終更新 10 分前 閲数の ARN ① arraws/ambda:ap-northeast-1:311330428996:function:Di oDocsExcelExportToS3 閲数 URL 情報 -
コード テスト モニタリング	設定 エイリアス パージョン	
コードソース 情報		アップロード元 ▼
 Lambda 関数「DioDocsExcelExport 	ToS3」のデブロイバッケージが大きすぎて、インラインコード編集を有効にT	できません。ただし、関数を呼び出すことはできます。
)、CloudShell フィードバック		© 2024, Amazon Web Services, Inc. またはその周速会社。 プライバシー 用語 Cookie の設定

「API Gateway」を選択し、さらに「Create a new API」を選択します。作成する API タイプは「REST API」を 選択して、セキュリティは「Open」を選択します。この状態で [追加] をクリックします。

				.
Ľ 🏻	トリガーの追加 - Lambda × +			- 0 X
$\leftrightarrow \rightarrow$	C ap-northeast-1.console.aws.amazon.com/lambda/	home?region=ap-northeast-1#/add/relation?focus=aws%2Flambda	⌖=arn%3Aaws%3Alambda%3Aap-northeast-1%3A31	133 🛧 🖸 🗏 🖬 🥹 🗄
aws		[Alt+S]	5 4 0 Ø #	京 🔻 ukgraphics @ 3113-3042-8996 🔻 🕯
<u> I</u> Lamb	da 🖻 S3 💆 Amazon CodeCatalyst 🧾 AWS App Runner 🔣 /	AWS Amplify 🛛 🐻 Amazon CodeWhisperer		
=				0
	トリガーの設定 情報			0
	API Gateway aws api application-services backend HTTP RES	▼		
	Add an API to your Lambda function to create an HTTP en types of RESTful APIs: HTTP APIs and REST APIs. Learn mo	dpoint that invokes your function. API Gateway supports two ore 🖸		
	Intent Use an existing api or have us create one for you.			
	Create a new API			
	O Use existing API			
	API type			
	 HTTP API Build low-latency and cost-effective REST APIs with built- in features such as OIDC and OAuth2, and native CORS support. 	 REST API Develop a REST API where you gain complete control over the request and response along with API management capabilities. 		
	Security Configure the security mechanism for your API endpoint.			
	Open	•		
	Additional settings			
	Lambda will add the necessary permissions for Amazon Al Learn more 🖸 about the Lambda permissions model.	PI Gateway to invoke your Lambda function from this trigger.		
		キャンセル 追加		
E Cloud	ihell フィードバック		© 2024, Amazon Web Services, Inc. またはその間運会社。	プライバシー 用語 Cookieの設定

以下のようにトリガーに API Gateway が追加されます。

DioDocsExcelExportToS3 - Lami ×		
C = ap-northeast-1.consol	e.aws.amazon.com/lambda/home?region=ap-northeast-1#/functions/DioDocsExcell	ExportToS3?tab=configure 🖈 🖸 💐 🖬 🧐
説 サービス Q 検索	[Alt+S]	区 👃 🕜 🞯 東京 🕶 ukgraphics @ 3113-3042-8996
oda 🧧 S3 👩 Amazon CodeCatalysi	📴 AWS App Runner 🛛 🔣 AWS Amplify 🛛 🤯 Amazon CodeWhisperer	
▼ 関数の概要 情報		Application Composer にエクスポート ダウンロード 🔻
タイアグラム テンプ	L→ト DioDocsExcelExportToS3 Stayers (0)	説明 - 温終更新 13 分前 開窓の ARN ● arr:aws1ambda:ap-northeast-1:311330428996:function:Di oDoc5ExceExportToS3 図数 URL 情報 -
コード テスト モニ 一般設定	タリング 設定 エイリアス バージョン	
トリガー		
アクセス権限	Q、トリカーを検索	< 1 >
送信先	 トリガー 	
関數 URL	API Gateway: DioDocsExcelExportToS3-API amaws:execute-apicap-northeast-1:311330428996kij56d3ki1/*/*/D	NoDocsExcelExportToS3
環境変数	► 詳細	se-ramazonaws.com/uerauty.piopocsexcerexport1053
IShell フィードバック		© 2024, Amazon Web Services, Inc. またはその関連会社。 プライバシー 用語 Cookie の話

デプロイしたアプリケーションを確認

AWS のコンソールで作成した API「DioDocsExcelExportToS3-API」の「リソース」から「API のデプロイ」を 選択します。



デプロイされるステージは「default」を選択して[デプロイ]をクリックします。

→ C 😄 ap-northeast-1.consol	e.aws.amazon.com/apigatewa	ay/main/apis/kj56d3ki1i/resources?api=kj56d3ki1i®ion=ap-northeast-1		☆	10 ₹	0 🍕
5 ∰ 9-cx q + 5		Tan+51	8 4 0	 9.4 • 	iil griphics go 311	5-10-12
		Deploy API X				
		ADIがデブロイネわるフテージを選択します。 励うげ ADIのテフト版を beta という名				
		前のステージにデブロイできます。				
		7 = 24				
		A5-5				
		default				
		デプロイメントの説明				
		1				
		キャンセル デプロイ				
			- Storesta			

API「DioDocsExcelExportToS3-API」の「ステージ」から「GET」 を選択します。



「URL の呼び出し」に表示されている API の URL をコピーして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=DioDocsForExcel」を追加します。



この API を実行するとクエリパラメータで渡した文字列「DioDocsForExcel」が追加された Excel ファイル 「Result.xlsx」が Amazon S3 のバケット「diodocs-file-export」に出力されます。





红 自動保存	(17)	5-6-	⊽ Res	ult.xlsx • この	PC に保存済	<i></i> ж ∨	Q	検索									-	0 -	0	×
ファイル ホーム	、挿入 ペ	ージ レイアウト	数式 テ	データ 校閲	表示 自	動化 開発	ヘルプ												ピ 共有	~
A1	• : × ~	fx Hello	, DioDocs	ForExcel!!																\sim
A	В	С	D	E	F	G	Н	1	J	К	L	М	N	0	Р	Q	R	S	Т	
1 Hello, Die	DocsForEx	cel!!																		-1
2																				-1
3																				-1
4																				-1
6																				1
7																				1
8																				1
9																				
10																				
11																				-1
12																				1
13																				1
15																				1
16																				1
17																				1
18																				
19																				
20																				-1
21																				1
22																				1
24																				1
25																				
26																				
27																				
28																				
29																				
30																				-1
32																				
32																				Ŧ
$\langle \rangle$	Sheet1	Evaluati	on Warnin	g -	-							-	_	_	_			_	_	Þ
準備完了 🐻	蒙 アクセシビ	リティ: 検討が必	要です													=			+ 10	00%

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

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

```
public async Task<APIGatewayProxyResponse> FunctionHandler(APIGatewayProxyRequest input, ILambdaContext
context)
{
    APIGatewayProxyResponse response;
    try
    {
        // クェリ文字列を取得
        string? queryString;
        input.QueryStringParameters.TryGetValue("name", out queryString);
        // ドキュメントに追加するテキスト
        string Message = string.IsNullOrEmpty(queryString)
        // ドキロし, {queryString}!!";
        //GcPdfDocument.SetLicenseKey("製品版またはトライアル版のライセンスキーを設定");
    }
}
```

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

}

{

}

```
g.DrawString(Message,
       new TextFormat() { Font = StandardFonts.Helvetica, FontSize = 12 },
       new PointF(72, 72));
   using (var ms = new MemoryStream())
   {
       doc.Save(ms, false);
       // S3にアップロード
       AmazonS3Client client = new AmazonS3Client(RegionEndpoint.APNortheast1);
       var request = new PutObjectRequest
       {
           BucketName = "diodocs-export",
           Key = "Result.pdf",
           InputStream = ms
       };
       await client.PutObjectAsync(request);
   }
   response = new APIGatewayProxyResponse
   {
       StatusCode = (int)HttpStatusCode.OK,
       Body = "ファイルが保存されました。",
       Headers = new Dictionary<string, string> {
           { "Content-Type", "text/plain; charset=utf-8" }
       }
   };
catch (Exception e)
    response = new APIGatewayProxyResponse
   {
       StatusCode = (int)HttpStatusCode.InternalServerError,
       Body = e.Message,
       Headers = new Dictionary<string, string> {
           { "Content-Type", "text/plain" }
       }
   };
```

さいごに

}

動作を確認できる AWS Lambda アプリケーションのサンプルはこちらです。 <u>https://github.com/MESCIUSJP/ExcelExportAWSLambda2</u> https://github.com/MESCIUSJP/PDFExportAWSLambda2

AWS Lambda 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 AWS Lambda で「<u>DioDocs(ディオドック)</u>」を使用した C#(.NET 8)の Lambda 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 AWS Lambda 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 ● オフ) E 9	~ CI ~ ~	;		Result.xlsx	-		Q
ファイル ホーム	挿入	ページ レ1	イアウト 数	式 データ	校閲	表示 開発	そ ヘルプ	
いた 貼り付け 日 ~	MSP明朝 BI <u>U</u>	• ⊞ •	~ 11	→ A [^] A [*]	= = <u>=</u>	≫~ ~ E = = [設 折り返してき 豆 セルを結合	全体を表示 して中央捕
クリップボード 15		フォント		_ل دا		百百	置	
A1		•	X 🗸	fx Zh	,にちは、世	±界!		
A	В	С	D	E	F	G	Н	I
1 こんにちは t	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)

.NET 8 のランタイムが含まれる Lambda 関数の実行環境 OS は、「<u>Amazon Linux 2023</u>」になっています。 Windows OS とは違って Amazon Linux 2023 には日本語フォントは含まれていませんので、デフォルトの状態 で PDF 出力を実行すると文字化けが発生する、または文字列そのものが表示されない現象が発生します。

参考:「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 AWS Lambda で DioDocs を使用する場合は、Lambda レイヤーを使用して日本語フォントを追加する必要があります。

参考:「レイヤーの使用方法」を見る

まずローカル環境の適当な場所に「.fonts」フォルダを作成し、そこにフォントファイルを格納後、「.fonts」フ ォルダを ZIP ファイルへ圧縮します。本記事では「IPAex フォント」を「.fonts」フォルダにコピーしています。



AWS のコンソールで AWS Lambda の「レイヤー」を選択し、[レイヤーの作成] をクリックします。

S III サービス Q サービス。	特徴、ブログ、およひ	ドキュメントなどを検索	[Alt+S]		D & Ø **	ukgraphics @ 3113-3042-8
ambda 🔞 S3 🙀 API Gateway						~
WS Lambda X	Lambda > L	1447-				
ソシュポード	レイヤー	(0)			C	レイヤーの作成
プリゲーション 数	Q 904	属性によるフィルター、また	はキーワードによる検索			< 1 >
の他のリソース	名前	バージョン	互換性のあるランタイム	互換性のあるアーキテクチャ	パージョン	2 ARN
- ド署名設定 イヤー			表示	するデータがありません。		

レイヤーの名前に「DioDocs-Japanese-Fonts」を、説明に「DioDocs で使う日本語フォント」を設定して[ア ップロード]をクリックします。

レイヤーの作成 - Lambda × +	
C 🕼 ap-northeast-1.console.aws.amazon.com/lambda/home?region=ap-northeast-1#/create/layer	☆ ひ: ◎ ₹ 4 🖬
発生しました。再ロードしてください。	
Lambda > レイヤー > レイヤーの作成	
レイヤーの作成	
レイヤー設定	
名前	
DioDocs-Japanese-Fonts	
説明 - オブション	
DioDocsで使う日本語フォント	
◎ .zip ファイルをアップロード	
○ Amazon S3 からファイルをアップロードする	
日 アップロード	
10 MBより大きいファイルの場合は、Amazon S3 を使用したアップロードを検討してください。	
互換性のあるアーキテクチャ・オブション 情報 レイヤーと互換性のある金合ヤットアーキテクチャを演得します。	
x86,64	
Li armo4	
互換性のあるランタイム・オブション 情報 最大 15 個のランタイムを選択します。	
⋽>タイム	
ライセンス・オブション 傍職	
6 2024	Amazon Web Services, Inc. またはその関連会社。 プライバシー 用語 Cook

先ほどローカル環境で作成した ZIP ファイル「.fonts.zip」を選択して[作成] ボタンをクリックします。

① ① ①				- 0
レイヤー設定 名前 BioDocs-Japanese-Fonts 2期・オブション DioDocsで使う日本語フォント ③ コウ フィイルをアップロード ④ ハazon 53 からファイルをアップロードする ① アップロード ▲ Amazon 53 まゆうD・オブション 10HB よび入意いファイルの増付は、Amazon 53 を使用したアップロードを触げしてください、 ■ かけの よび入ホゾファイルの増付は、Amazon 53 を使用したアップロードを触げしてください、 ■ かけの よび入ホゾファイルの増付は、Amazon 53 を使用したアップロードを触げしてください、 ■ かけの よび入ホイプション 10% ■ オmodel ■ Sta Seport - ホプジョン 10% ● インイル - 変更 10% ● オmodel ● アップロード ● オージアクチャー・オブション 10% ● オmodel ● オmodel ● オmodel ● フィイル - 変更 ● フィイル - 変更 ● フィイル ● フィイル - 変更 ● フィイル ● フィイム ● フィイル ● フィイム ● フィイム ● オレン化	C ap-northeast-1.console.aws.amazon.com/lambda/home?region=ap-northeast	-1#/create/layer	☆ <u>む</u> @	₹ ± 🛛
レイヤー設定 定 回のたのにより調加に合いていた ビロシロンでは、日本語コスット ロ・カフランレ ロ・カフライルをアップロード ヘ mazon 53 がらファイルをアップロードする ① アップロード クップロード ● パップロード ● パップ・ ● パッグ・ ● パッグ	生しました。再ロードしてください。			
<form> ch Diboos-games-rons Cboos-departed-rons for yorg-n-ro for yorg-n-ro for yorg-n-ro-ro-ro-ro-ro-ro-ro-ro-ro-ro-ro-ro-ro-</form>	レイヤー設定			
am 「DioDocs/apanes-Fonts BioDocs/GEGERABJ7メト ● よりフイルをアップロード ● 小和con 53 からファイルをアップロードする ● アップロード 「Amaon 53 からファイルをアップロードもあり ● アップロード ● パップロード ● パッグロード ● パッグロード ● パック・オプション 10% ● パック ● 日始色のあるワンクイム・オプション 10% ● コウタイム ● コウタイム ● コウタイム ● コウタイム ● コーパら ● ローパら ●	な前			
歴期- 北フション DioDouxで使う日本語フォント ● ホロ フィ パルをアップロード ● 小田 なの 53 から ファ イルを アップロード する 原 アップロード 「加たご加 10 49 よりたき いプア イルの増信は、Amazon 53 を使用した アップロード き機制して くだきい、 10 49 よりたき い ファ イルの増信は、Amazon 53 を使用した アップロード き機制して くだきい、 10 49 よりたき い ファ イルク増信 は、Amazon 53 を使用した アップロード き機制して くだきい、 11 49 よりたき い ファ イホク ブチャ オブション 10 11 12 49 にのある きた ウィ オブション 10 11 13 大 町 小 15 ファ イム 15 ファ イム スプション 10 11 15 大 町 公 15 ファ イム 16 インス・オブション 10 11 17 大 コブション 10 11 18 大 田 小 19 ノンドム 17 ノンド	DioDocs-Japanese-Fonts			
DioDosで使き日本語フォント DioDosでした ① ロアフロード ① アップロード 「カップロード 「カップロード 「カップロード 「カップロード 「カップロード 「かたない 「たちいう 「カップロード 「かれないの53 からファイルの増点は、Amazon 53 を使用したアップロードを検討してだされ、 「たちのあるかたマッチアクチャ・オブション 10% してしたした日間的ためるの特化マットアーキテクチャ も満知します。 こ、さらん4 □ 即的た こ フタイム 【 こ フタイム 【 ご フタイム 【 ご フタイム 【 ご フタイム 【 ご フタイム 【 こ フタイム 【 ・ (*) ・ (*) ・ 1 ・ (*) ・ (*) ・ (*) ・ (*) ・ (*) ・ (*) ・ (*) ・ (*) ・ (*) ・ (*) ・ (*) ・ (*) </td <td>説明 - オブション</td> <td></td> <td></td> <td></td>	説明 - オブション			
 ● : ip ファイルをアップロード ● : ip ファイルをアップロードする ● : ip ファブロード ● : ip ファブロード ● : ip ひっていたをアップロードを検討してください。 ● : ip ひったうブヤ・オブション (16) ● : ip ひったうジャイムや選択します。 ● : ip 00 ついろうシックイム・オブション (16) Bx 15: 40 つ ワンタイムを選択します。 ● : ip 00 ついろう (16) ● : ip 00 ついろう (16)	DioDocsで使う日本語フォント			
● アップロード 「かちご戸」 × 5.5 MB × 10 Ma 20 大きいファイルの増信は、Amazen S3 を使用したアップロードを検討してください、 三酸性のあるアーキアクチャ・オブション 常飾 しパー・こ回性ののよる寺たシャイム・オブション 常飾 日本的たるラブシタイム・オブション 常飾 日本的たるラブシタイム・オブション 常飾 ランタイム マ C ライセンス・オブション 常飾 レパー・この生ののようたい・オブション 常飾 レパー・この生ののようたい・オブション 常飾 レパー・コンション 常飾 レパー・コンション 常飾	● zip ファイルをアップロード ○ Amazon S3 からファイルをアップロードする			
Instruction × 30.56 м/b * 30.66 m/b * 30.66 m/b * 30.66 m/b * 30.67 m/b<	דילט איז			
ID MB より大きいファイルの場合は、Amazon S3 を使用したアップロードを検討してください。 互換性のあるアーキテクチャ・オブション 領制 レイャーと正路地である命やセットアーキテクチャを選択します。 □ S6,64 □ arm64 互換性のあるランタイム・オブション 領制 尽大 15 知のランタイムを提択します。 □ アンタイム ○ アイレンス・オブション 領制 ● <td>.fonts.zip 9.76 MB</td> <td>×</td> <td></td> <td></td>	.fonts.zip 9.76 MB	×		
ランタイム ▼ C ラインス-オブション 協調 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	 互換性のあるアーキテクチャ・オブション 情報 レイヤーと互換性のある命をセットアーキテクチャを選択します。 □ x86_64 arm64 互換性のあるランタイム・オブション 情報 最大 15 回のランタイムを認択します。 			
ライセンス・オブション (1)開 	ランタイム	• C		
キャンセル 作成	ライセンス・オブション 16種			
キャンセル 作成		le le		
		キャンセル 作成		

Lambda レイヤーが作成されます。「バージョン ARN」は、デプロイした Lambda 関数から Lambda レイヤー を追加する際に使用するのでコピーしておきます。

mbda > レイヤー > DioDocs-Japanese-Fo	onts	
ioDocs-Japanese-Fonts	5	削除 ダウンロード バージョンの作
⊘ レイヤー DioDocs-Japanese-Fonts のパー:	ジョン2が正常に作成されました。	:
バージョンの詳細		
バージョン 2	バージョン ARN の amawslambda:ap-northeast-1:311330428996:layer:DioDocs- apanese-Fonts:2	説明 J DioDocsで使う日本語フォント
作成済み 51 秒前 互換性のあるアーキテクチャ -	ライセンス -	互換性のあるランタイム -
バージョン このバージョンを使用する世	司 政	
すべてのバージョン (1)		< 1
バージョン バージョン ARN		說明

AWS コンソールから Lambda の [関数] をクリックし、デプロイした関数名をクリックします。表示された詳細画面の下部にある [レイヤーの追加] をクリックします。

ap-northeast-1.console.aws.amazon.com	n/lambda/home?region=ap-northeast-1#/functions/DioDocsUseJpnFont?tab=code	☆ ひ ◎ ₹ ₺
ました。再ロードしてください。		
コードソース 情報		アップロード元 🔻
🚯 Lambda 関数「DioDocsUseJpnFont」のデ	デブロイバッケージが大きすぎて、インラインコード編集を有効にできません。ただし、	、関数を呼び出すことはできます。
コートのノロハティ 情報		
バッケージサイズ	SHA256 八ッシュ	最終更新
42.8 MB	kiiFK+4urFiA00v0yK+YPgZiYT69XoBBPin1UJ2ha6E=	2024年3月11日 16:59 JST
ランタイム設定 情報		編集 ランタイム管理設定を編集
ランタイム	ハンドラ 情報	アーキテクチャ 情報
.NET 8 (C#/F#/PowerShell)	DioDocsUseJpnFont::DioDocsUseJpnFont.Function::FunctionHa er	ndl x86_64
▶ ランダイム官理設定		
レイヤー 情報		編集 レイヤーの追加
マージ順序 名前	レイヤーバージョン 互換性のあるランタイ.	ム 互換性のあるアーキテクチャ バージョン ARN
	表示するデータがありません。	
	2.1.7.87 7.1.05.56.2.10	

「レイヤーを追加」画面で「ARN を指定」を選択し、「ARN を指定」欄に先ほどコピーした Lambda レイヤーの ARN を入力して、[追加] をクリックします。

▼ N レイヤーを追加 - Lambda × +				-	σ×
← → C 😅 ap-northeast-1.console.aws.amazon.com/lambda/	nome?region=ap-northeast-1#/add/layer?function=DioDocsUseJpnF	ont 🖈	1) Ø		J 🥹 :
エラーが発生しました。再ロードしてください。					·
					0
関数のランタイム設定					
ランタイム .NET 8 (C#/F#/PowerShell)	アーキテクチャ ×86_64				
レイヤーを選択					
レイヤーソース 情報 ランタイムおよび命作セットアーキテクチャと互換性のあるレイヤー を指定します。新しいレイヤーを作成することもできます。	から選択するか、レイヤーバージョンの Amazon リソースネーム (ARN)				
 AWS レイヤー AWS が提供するレイヤーのリスト から、レイヤーを選択します。 カイヤーを選択します。 	イヤー トまたは組織によっ レイヤーのリストか を選択します。 ・ ARN を指定 ます。				
ARN を指定 Amazon リソースネーム (ARN) を入力して、レイヤーを指定します。					
arn:aws:lambda:ap-northeast-1:311330428996:layer:Dio	Docs-Japanese-Fonts:2 検証				
説明 DioDocsで使う日本語フォント					
互換性のあるランタイム -					
互換性のあるアーキテクチャ -					
	キャンセル 追加				
		© 2024, Amazon Web Services, Inc. またはその関連会社。	プライバシ-	· 用語 Co	ookie の設定

Lambda 関数に Lambda レイヤーが追加されます。

3 ap-northeast-1.console.aws.amazon.co	m/lambda/home?regi	on=ap-northeast-1#/functions/DioDocsUseJpnFont?tab=code		☆ ひ @ ₹ ₺
ました。再ロードしてください。				
コードソース 情報				アップロード元 🔻
🚯 Lambda 関数「DioDocsUseJpnFont」の	デプロイバッケージが	大きすぎて、インラインコード編集を有効にできません。ただし、『	関数を呼び出すことはできます。	
コードのプロパティ 🗰				
バッケージサイズ		SHA256 八ッシュ	最終更新	
42.8 MB		kiiFK+4urFiA00v0yK+YPgZiYT69XoBBPin1UJ2ha6E=	2024年3月11日 17:03 JST	
ランタイム設定 情報			編集	ランタイム管理設定を編集
ランタイム		ハンドラ 情報	アーキテクチャ 情報	
		DioDocsUseJpnFont::DioDocsUseJpnFont.Function::FunctionHand	l x86_64	
.NET 8 (C#/F#/PowerShell)				
.NET 8 (C#/F#/PowerShell)		CI		
.NET 8 (CH/F#/PowerShell) ▶ ランタイム管理設定		τ.		
.NET 8 (C#/F#/PowerShell) ▶ ランタイム管理設定		G		
 NET 8 (C#/F#/PowerShell) ランタイム管理設定 レイヤー 情報 		G	[福集
 NET 8 (C#/F#/PowerShell) > ランタイム管理設定 レイヤー 情報 マージ順序 名前 	レイヤーバージョン	G B 換性のあるランタイム 互換性のあるアーキテクチャ	〔 J〔ージョン ARN	編集
NET 8 (C#/F#/PowerShell) ランタイム管理設定 レイヤー 情報 マージ順序 名前 1 DioDocs-Japanese-Fonts	レイヤーバージョン 2	 回換性のあるランタイム 互換性のあるアーキテクチャ ・ ・	ノ(ージョン ARN am:aws:lambda:ap-northeast-1:31	編集 レイヤーの追加 1330428996:layer.DioDocs-Japane

Visual Studio から Lambda 関数をデプロイする際に、以下の画面で [Add…] をクリックして環境変数を追加し ます。変数名は HOME で値は/opt を設定します。/opt は Lambda レイヤーが展開されるディレクトリです。こ の設定により Lambda 関数を実行するパス(ホームディレクトリ)が/opt になるので、Lambda レイヤーに格 納した.fonts 配下に含まれる日本語フォント「IPAex ゴシック」が利用できます。

🔋 Upload to AWS Lambda			_		×
Advanced Function	on Details gs for your func	tion.			
Permissions					
Select an IAM role to provide AWS credentials to o	our Lambda function	n allowing access to AWS Services like S3.			
Role Name: Existing role: lambda_exec_DioDocsU	JseJpnFont				~
xecution Memory (MB): 512 Timeout (Secs): 30 (1 - 900) PC If your function accesses resources in a VPC, select the list of subnets and security group IDs (these	Debugging and DLQ Resource: Enable active Environment KMS Key: Variable	Error Handling <pre></pre> <pr< td=""><td></td><td></td><td>•</td></pr<>			•
must belong to the same VPC). VPC Subnets: Security Groups:	HOME	/opt			Add
		Close Back N	ext	Upload	ł

この Lambda レイヤーで日本語フォントを追加した Lambda 関数で、以下のコードのように「IPAex ゴシック」 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 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 APIGatewayProxyResponse FunctionHandler(APIGatewayProxyRequest input, ILambdaContext context)
{
    APIGatewayProxyResponse response;
    string? queryString;
    input.QueryStringParameters.TryGetValue("name", out queryString);
    string Message = string.IsNullOrEmpty(queryString)
        ? "こんにちは、世界!"
        : $"こんにちは、世界!"
        : $"こんにちは、{queryString}!";
    Workbook workbook = new Workbook();
    workbook.Worksheets[0].Range["A1"].Font.Name = "IPAex ゴシック";
    workbook.Worksheets[0].Range["A1"].Value = Message;
    var base64String = "";
```

```
using (var ms = new MemoryStream())
{
    workbook.Save(ms, SaveFileFormat.Pdf);
    base64String = Convert.ToBase64String(ms.ToArray());
}
response = new APIGatewayProxyResponse
{
    StatusCode = (int)HttpStatusCode.OK,
    Body = base64String,
    IsBase64Encoded = true,
    Headers = new Dictionary<string, string> {
        {"Content-Type", "application/pdf"},
        {"Content-Disposition", "attachment; filename=Result.pdf"},
    }
};
```

```
return response;
```

```
}
```

Result.pdf - Adob ファイル (F) 編集 (E)	e Acrobat Reader DC (32-bit) 表示(V) 署名(S) ウィンドウ(W)) ヘルプ(H)			- c	3	×
ホーム ツール	Result.pdf	×		?	4	ログイ	>
B 🔶 🖥			(↑) (↓ 1 / 1) (→ (↓ 159% · □) (↓ 159% ·	ć	2		6
,	こんにちは、	世界!	xxm20107f7 x 構成: Turkippe (C0) エンコーディング::dentby+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、「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)」と同じ手順で Lambda レイヤーを使用して 日本語フォントを追加します。

以下のコードのように「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 APIGatewayProxyResponse FunctionHandler(APIGatewayProxyRequest input, ILambdaContext context)
{
   APIGatewayProxyResponse response;
   string? queryString;
   input.QueryStringParameters.TryGetValue("name", out queryString);
   string Message = string.IsNullOrEmpty(queryString)
   ? "こんにちは、世界!"
   : $"こんにちは、{queryString}!";
   GcPdfDocument doc = new GcPdfDocument();
   GcPdfGraphics g = doc.NewPage().Graphics;
   g.DrawString(Message,
       new TextFormat() { FontName = "IPAex ゴシック", FontSize = 12 },
       new PointF(72, 72));
   var base64String = "";
   using (var ms = new MemoryStream())
   {
       doc.Save(ms, false);
       base64String = Convert.ToBase64String(ms.ToArray());
   }
   response = new APIGatewayProxyResponse
   {
       StatusCode = (int)HttpStatusCode.OK,
       Body = base64String,
       IsBase64Encoded = true,
       Headers = new Dictionary<string, string> {
           { "Content-Type", "application/pdf" },
           { "Content-Disposition", "attachment; filename=Result.pdf"},
       }
   };
```

<pre>return response; }</pre>			
 Resultpdf - Adobe Acrobat Reader DC (32-bit) 7イル(Γ) 編集(E) 表示(Y) 署名(S) 9インドウ(W) ヘルルプ(H) ホーム ツール Resultpdf × Ŷ ြ ♀ ♀ ♀ ♀ ♀ ♀ ♀ ♀ ♀ ♀ ♀ ♀	1 / 1 🖡 🖑 \ominus 🕂 159% 🔹 🙀 🐼 🛃 🙀 🐺		- □ × ⑦ ▲ □ʲ/1> & ☑
こんにちは、世界!	 ×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 -この文書で使用しているフォント ■ 承 Meiryo (理め込みサブセット) 種類: TrueType (CD) エンコーディング: Identity+H 	×	^

本記事では AWS Lambda 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。

OK キャンセル

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