

2020年8月27日

株式会社インプレスR&D

<https://nextpublishing.jp/>

サーバレス環境で、保護されたPDFを簡単作成！
『AWS／サーバレスでつくる！安心安全PDF』発行
技術の泉シリーズ、8月の新刊

インプレスグループで電子出版事業を手がける株式会社インプレス R&D は、『AWS／サーバレスでつくる！安心安全PDF』（著者：木村 俊彦）を発行いたします。

最新の知見を発信する『技術の泉シリーズ』は、「技術書典」や「技術書同人誌博覧会」をはじめとした各種即売会や、勉強会・LT 会などで頒布された技術同人誌を底本とした商業書籍を刊行し、技術同人誌の普及と発展に貢献することを目指します。

『AWS／サーバレスでつくる！安心安全PDF』

<https://nextpublishing.jp/isbn/9784844378945>



著者：木村 俊彦

小売希望価格：電子書籍版 2000 円（税別）／印刷書籍版 2500 円（税別）

電子書籍版フォーマット：EPUB3／Kindle Format8

印刷書籍版仕様：B5 判／カラー／本文 166 ページ

ISBN：978-4- 8443-7894-5

発行：インプレス R&D

<< 発行主旨・内容紹介 >>

電子書籍の普及や技術同人誌の盛り上がりにより、PDF を以前より手にする機会はもちろん、配布する機会も多くなりました。その中で、再配布や加工を防止している PDF ファイルはまだまだ少数です。そこで本書では AWS のサーバレス環境を利用して、PDF ファイルを動的に保護する手段について解説しています。さらに、構築した環境を組み合わせ、最終的に API 化する部分まで紹介します。

（本書は、次世代出版メソッド「NextPublishing」を使用し、出版されています。）

ウォーターマークやカスタムプロパティの追加などの方法を紹介



図 3.12: ウォーターマークの追加

前書き

第4章 文字列入りページを追加する

次に、文字列入りページの追加を実装します。新規にページを作成しますが、こちらも難しくありません。日本語も好きなフォントも使えます。

4.1 概要

本章の処理では、メールアドレスなどの一意な文字列や、メッセージを埋め込んだページを作成し、そのPDFと元となるPDFを連結させます。全ページに埋め込むウォーターマークに比べると、各ページの見やすさはそのまま、識別できるものを仕込むことができます。ただ、こちらもツールを使うとそのページのみ削除できてしまうため、暗号化と組み合わせると効果的です。

今回は、pdf-sign-go という名前でもLambda関数を作成しています。設定や使用するツールはこれまでと同一です。

ライブラリーとして pdfcpu に加え、gopdf を利用しています。Go 言語で利用できる PDF 作成のライブラリーは多くありますが、その中で数少ない日本語 (マルチバイト) に対応しているライブラリーです。

また、文字を埋め込むために別途フォントが必要で、好みの日本語フォント (TrueType) をあらかじめ準備してください。今回は「たぬき油性マジック」を準備し、main.go と同じディレクトリーに TanukiMagic.ttf という名前でも保存しています。

4.2 コード

今回デプロイするコードは次のものです。第2章で作成した sign ディレクトリーの中に、main.go のファイル名で作成します。

```

リスト 4.1: main.go
package main

import (
    "context"
)

1 https://github.com/va1en7in/pdfcpu
2 利用するフォントのライセンスもあらかじめ確認してから利用してください
3 https://tanukimagic.com/tanuki-permanent-marker/
                    
```

38 | 第3章 ウォーターマークを追加する
第4章 文字列入りページを追加する | 39

API 化の方法を解説




図 7.55: 変更結果確認

CORS の有効化

- ✓ OPTIONS メソッドを有効化する
- ✓ 追加メソッドも追加する必要がある。ステータスコードも合わせて OPTIONS メソッドに追加する
- ✓ Allow 設定も OPTIONS メソッドに追加する
- ✓ Allow-Origin-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin メソッドもステータスコードも OPTIONS メソッドに追加する
- ✓ Access-Control-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin 設定もステータスコードも OPTIONS メソッドに追加する
- ✓ Access-Control-Allow-Origin メソッドもステータスコードも POST メソッドに追加する
- ✓ Access-Control-Allow-Origin 設定もステータスコードも OPTIONS メソッドに追加する
- ✓ Access-Control-Allow-Origin 設定もステータスコードも OPTIONS メソッドに追加する

同じ作業を /generate/status に対しても行い、両方に OPTIONS メソッドが設定できれば完了です。

図 7.56: 両方へ適用

最後に API を公開します。アクションから「API のデプロイ」を選択します。

図 7.57: デプロイ準備

デプロイ設定画面になります。「デプロイされるステージ」で「新しいステージ」を選択し、ステージ名は「v1」と入力して「デプロイ」を選択します。




図 7.58: デプロイ設定

「API の呼び出し」という表示の横にエンドポイントが表示されましたので、これで API が公開されました。

図 7.59: デプロイ完了

試しに curl を用いて実行します。

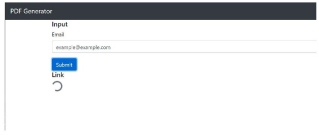
```

$ curl -X POST -H 'Content-Type:application/json' -d '{"name":"curlRun",
"email":"curl@example.com"}' https://82x7jense5.execute-api.ap-northeast-1.amazonaws.com/v1/generate
{"executionArn":"arn:aws:states:ap-northeast-1:1234567890:execution:PDFEditStateMachine:curlRun","startDate":1.579773511957E9}
$ curl -X POST -H 'Content-Type:application/json' -d '{"executionArn":"arn:aws:states:ap-northeast-1:1234567890:execution:PDFEditStateMachine:curlRun"}' https://lcfk3e3usku.execute-api.ap-northeast-1.amazonaws.com/v1/generate/status
{"output":{"email":"curl@example.com"},"path":"public/1579773516_644c386334854e477ee99946bbab71.pdf()","status":"SUCCESS"}
                    
```

106 | 第7章 API 化する
第7章 API 化する | 107

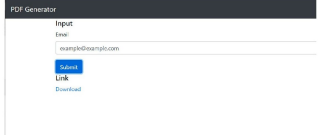
Web サイトへの組み込みを紹介

図8.2: ロード中



生成が完了すると、「Download」のリンクが生成されます。クリックするとダウンロードが始まり、生成されたPDFを取得することができます。

図8.3: 生成完了



もしボタンを押しても反応がなかったり、なかなかリンクが生成されない場合は処理に問題がある可能性があります。その場合は、デベロッパーツール（開発者ツールという名称の場合もあります）を利用することで検証可能です。どこかにエラーがある場合は、「Console（コンソール）」タブに赤枠でエラーメッセージが表示されます。APIとの通信でのエラーは、API自体にエラーが発生している場合は、「Network（ネットワーク）」に通信履歴が表示されます。そのため、ツールを開いたまま再度処理を実行し、リクエストやレスポンスに問題がないか確認します。

図8.4: Consoleタブ

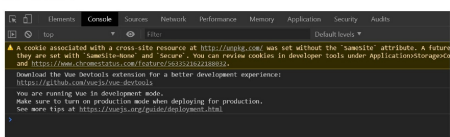
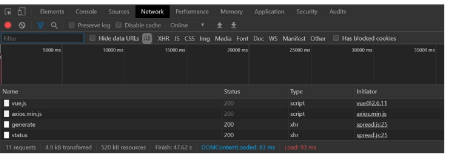



図8.5: Networkタブ



8.4 S3で公開してみる

もし今回のサイトを公開したい場合、サーバサイドの技術は特に使っていませんので、サーバは問いません。今回はS3のStatic website hostingを利用して、公開する方法を紹介します。S3のバケットをWebサイトとして公開できる機能で、バケットのポリシーやアクセス設定を行うだけで公開できる手軽な機能です。ただし、カスタドメインが設定できなかったり、HTTPSに対応していないため、本格的に公開する際はPDF生成と同様に、CloudFrontも利用することをオススメします。まずはバケットを作成します。このとき、「パブリックアクセスをすべてブロック」のチェックを外し、「現在の設定により～」のチェックボックスをオンにします。これは、Webサイトとしてホスティングするには、バケットを公開する必要があるためです。

図8.6: パブリックアクセスを許可



作成後、バケットの詳細画面に移動します。最初にStatic website hostingを有効にします。「プ

<<目次>>

- 第1章 基本的な準備をする
- 第2章 開発周りの準備をする
- 第3章 ウォーターマークを追加する
- 第4章 文字列入りページを追加する
- 第5章 カスタムプロパティを追加する
- 第6章 暗号化する
- 第7章 API化する
- 第8章 Webサイトに組み込む
- 付録A Serverless Frameworkを利用してLambdaを管理する
- 付録B CloudFormationを利用してAWSリソースを管理する

<<著者紹介>>

木村 俊彦
宮城県出身・在住。学生時代からプログラミングに勤しみ、そのままシステム開発会社へ就職。その後Web制作会社などを経て、現在はサーバサイド(PHP)のアプリケーション構築がメインで、インフラ(AWS)も扱う。クラウド(AWS/GCP)とコンテナ(Docker/Kubernetes)とIoTに興味がある。PHPカンファレンス仙台コアスタッフ、技術同人誌サークル「杜の都の開発室」主宰。

<<販売ストア>>

電子書籍:

Amazon Kindle ストア、楽天 kobo イーブックストア、Apple Books、紀伊國屋書店 Kinoppy、Google Play Store、
honto 電子書籍ストア、Sony Reader Store、BookLive!、BOOK☆WALKER

印刷書籍:

Amazon.co.jp、三省堂書店オンデマンド、honto ネットストア、楽天ブックス

※ 各ストアでの販売は準備が整いしだい開始されます。

※ 全国の一般書店からもご注文いただけます。

【インプレス R&D】 <https://nextpublishing.jp/>

株式会社インプレスR&D(本社:東京都千代田区、代表取締役社長:井芹昌信)は、デジタルファーストの次世代型電子出版プラットフォーム「NextPublishing」を運営する企業です。また自らも、NextPublishing を使った「インターネット白書」の出版など IT 関連メディア事業を展開しています。

※NextPublishing は、インプレス R&D が開発した電子出版プラットフォーム(またはメソッド)の名称です。電子書籍と印刷書籍の同時制作、プリント・オンデマンド(POD)による品切れ解消などの伝統的出版の課題を解決しています。これにより、伝統的出版では経済的に困難な多品種少部数の出版を可能にし、優秀な個人や組織が持つ多様な知の流通を目指しています。

【インプレスグループ】 <https://www.impressholdings.com/>

株式会社インプレスホールディングス(本社:東京都千代田区、代表取締役:松本大輔、証券コード:東証1部9479)を持株会社とするメディアグループ。「IT」「音楽」「デザイン」「山岳・自然」「モバイルサービス」「学術・理工学」「旅・鉄道」を主要テーマに専門性の高いメディア&サービスおよびソリューション事業を展開しています。さらに、コンテンツビジネスのプラットフォーム開発・運営も手がけています。

【お問い合わせ先】

株式会社インプレス R&D NextPublishing センター

TEL 03-6837-4820

電子メール: np-info@impress.co.jp