CI/CD Flutter

2020-09-13

Github ActionsでiOSアプリのCI/CDをやってみた

FlutterでのiOSアプリ開発

GithubActions x Deploygateで

CI/CD環境を整える！

FlutterでiOS/Androidアプリを開発していた際に、GithubActionsを使ってiOSのCI/CD環境を構築した際のまとめです。

INDEX

この記事 is 何？

Yahoo!さん主催のハッカソン「HackU」に向けてチーム開発している際に、デザイナーのメンバーにもアプリを手元の実機で確認してもらえる環境が作りたい！

できれば[redacted]みたいなカッコよさげなことしてみたい！

と思ってCI/CD環境を整えた際に得た知見をまとめた記事です。

Android用のCI/CD環境を整えたお話はこちらにまとめました。

兎にも角にも完成品のyamlファイルをひとまず次に載せておきます。

完成品

```
name  iOS CI/CD

on
  push
    branches  [ master ]
  pull_request
    branches  [ master ]

jobs
  build
    runs-on  macos-latest

    steps
      - uses  actions/checkout@v2

      - name  Select Xcode version 11.7
        run  sudo xcode-select -s '/Applications/Xcode_11.7.app/Contents/Developer'

      - name  Show Xcode version
        run  xcodebuild -version

      - name  setup cache
        uses  actions/cache@v1
        with
          path  /Users/runner/hostedtoolcache/flutter
          key  ${{runner.OS}}-flutter-install-cache

      - name  install flutter
        uses  subosito/flutter-action@v1
        with
          flutter-version  '1.20.2'
          channel  'stable'

      - name  flutter dependencies install
        run  flutter pub get

      # 証明書関連のファイルを生成します
      - name  Import Provisioning Profile
        run  |
          mkdir -p ~/Library/MobileDevice/Provisioning\ Profiles
          touch ~/Library/MobileDevice/Provisioning\ Profiles/decoded.mobileprovisi
          echo -n '${{ secrets.PROVISIONING_PROFILE }}' | base64 -d -o ~/Library/Mo

      # 署名をします
      - name  Import Code-Signing Certificates
        uses  Apple-Actions/import-codesign-certs@v1
        with
          p12-file-base64  ${{ secrets.CERTIFICATES_P12 }}
          p12-password  ${{ secrets.CERTIFICATE_PASSWORD }}

      - run  flutter build ios

      # 頑張ってipaを出力します
      - name  XCode Build Archive
        uses  yukiarrr/ios-build-action@v1.1.1
        with
          project-path  ios/Runner.xcodeproj
          p12-base64  ${{ secrets.CERTIFICATES_P12 }}
          certificate-password  ${{ secrets.CERTIFICATE_PASSWORD }}
          mobileprovision-base64  ${{ secrets.PROVISIONING_PROFILE }}
          code-signing-identity  ${{ secrets.CODE_SIGNING_IDENTITY }}
          team-id  ${{ secrets.TEAM_ID }}
          workspace-path  ios/Runner.xcworkspace
          output-path  app-release.ipa
          export-method  ad-hoc

      # ipaをdeploygate経由で配布します
      - name  Distribute iOS app
        run  |
          curl \
            -H "Authorization: token ${{secrets.DEPLOYGATE_API_KEY}}" \
            -F "file=@/Users/runner/work/tapiten_app/tapiten_app/app-release.ipa" \
            -F "message=git:$GIT_HASH" \
            -F "distribution_name=$GIT_BRANCH" \
            -F "release_note=new ios build" \
            -F "distribution_key=${{secrets.IOS_DISTRIBUTION_HASH}}" \
            "https://deploygate.com/api/users/${{secrets.DEPLOYGATE_USER}}/apps"
```

使用したもの

- publicなリポジトリ
- deploygate

publicなリポジトリにした理由は特に隠さなければいけない情報がリポジトリに含まれていないことと、GithubActionsの使用制限がpublicリポジトリの場合は無いらしいから、という理由です。

yamlファイルの説明

ビルドが走るタイミングの設定

```
on
  push
    branches  [ master ]
  pull_request
    branches  [ master ]
```

この部分で、[redacted]と、[redacted]にワークフローが実行されるように設定しています。

XCodeのバージョン設定

```
- name  Select Xcode version 11.7
  run  sudo xcode-select -s '/Applications/Xcode_11.7.app/Contents/Developer'
```

iOSビルド時にはたまにXCodeのバージョンの差異によってビルドが失敗したりすることもあるので、バージョンを指定しておきます。

Flutterのインストールをキャッシュしておく

```
- name  setup cache
  uses  actions/cache@v1
  with
    path  /Users/runner/hostedtoolcache/flutter
    key  ${{runner.OS}}-flutter-install-cache
```

この部分でFlutterのインストール先のファイルをキャッシュ対象にしています。

毎回インストールしていると遅くなってしまうので、[redacted]です。

Flutterのインストール

```
- name  install flutter
  uses  subosito/flutter-action@v1
  with
    flutter-version  '1.20.2'
    channel  'stable'
```

Flutterもローカル環境と同じバージョンをインストールします。

Flutterのライブラリ関連のインストール

```
- name  flutter dependencies install
  run  flutter pub get
```

Flutterのライブラリ関連のインストールを行います。

Provisioning Profileのデコード

```
# 証明書関連のファイルを生成します
- name  Import Provisioning Profile
  run  |
    mkdir -p ~/Library/MobileDevice/Provisioning\ Profiles
    touch ~/Library/MobileDevice/Provisioning\ Profiles/decoded.mobileprovision
    echo -n '${{ secrets.PROVISIONING_PROFILE }}' | base64 -d -o ~/Library/MobileDe
```

Githubのsecretsに設定しているbase64でエンコードしたprovisioning_profileを取り出して、デコードします。

[redacted]注意してください。

最後にも記載していますが、改行コードが含まれているだけでデコード後の値が変わってしまいエラーになってしまいます。

Certificateの設定

```
# 署名をします
- name  Import Code-Signing Certificates
  uses  Apple-Actions/import-codesign-certs@v1
  with
    p12-file-base64  ${{ secrets.CERTIFICATES_P12 }}
    p12-password  ${{ secrets.CERTIFICATE_PASSWORD }}
```

base64でエンコードしたp12ファイルの中身と設定したパスワードからcodesignをします。

ビルドとアーカイブ

```
- run  flutter build ios

# 頑張ってipaを出力します
- name  XCode Build Archive
  uses  yukiarrr/ios-build-action@v1.1.1
  with
    project-path  ios/Runner.xcodeproj
    p12-base64  ${{ secrets.CERTIFICATES_P12 }}
    certificate-password  ${{ secrets.CERTIFICATE_PASSWORD }}
    mobileprovision-base64  ${{ secrets.PROVISIONING_PROFILE }}
    code-signing-identity  ${{ secrets.CODE_SIGNING_IDENTITY }}
    team-id  ${{ secrets.TEAM_ID }}
    workspace-path  ios/Runner.xcworkspace
    output-path  app-release.ipa
    export-method  ad-hoc
```

ipa出力のためのアーカイブにはiOS build actionを利用させて頂きました。

使用しているSecretsの詳細

名前	内容
PROVISIONING_PROFILE	mobileprovisionをbase64でエンコードしたもの
CERTIFICATES_P12	p12ファイルをbase64でエンコードしたもの
CERTIFICATE_PASSWORD	p12ファイルに設定したパスワード
CODE_SIGNING_IDENTITY	code sign identity
DEPLOYGATE_API_KEY	Deploygateのアカウント設定>APIKeyに記載されている値です
DEPLOYGATE_DIST_PAGE_HASH	Deploygateのアプリ配布ページURLに記載されているハッシュ値です
DEPLOYGATE_USER	Deploygateのアカウントに登録されたユーザ名です

詰まったこと

base64のデコード

base64にエンコードしたprovisioning profileのデコードがどうやらうまく行っていない…

というところで数時間を費やしたのですが、原因は[redacted]ことでした。

```
base64 sample.mobileprovision | pbcopy
```

でコピーアンドペーストしていたのですが、最後の改行文字ももちろん文字として認識されるのでデコードした際に違う値となってしまっていたのが原因でした。

言われてみればなんてことない些細なミスなのですが、気づくのに相当時間を費やしてしまったので載せておきます。

マミーめも〜 (id:flat-M_M) 191日前

関連記事

2020-09-13
XCodeのビルドを高速化する並列ビルドの設定方法
GithubActionsを使ったiOSのCI/CD環境構築の際にiOSのビルドが...

2020-09-13
Github ActionsでAndroidアプリのCI/CDをやってみた
FlutterでiOS/Androidアプリを開発していた際に、GithubActions...

コメントを書く

XCodeのビルドを高速化する並列ビルドの設...
Github ActionsでAndroidアプリのCI/CDを...