

Firebase App Checkを実装したので紹介



Press Space for next page →



自己紹介

- 📄 飯野陽平 ([wheatandcat](#))
- 🏢 会社: [合同会社UNICORN](#) 代表社員
- 📖 Blog: <https://www.wheatandcat.me/>
- 🛠️ 今までに作ったもの
 - [memoir](#)
 - [OOMAKA](#)
 - [MarkyLinky](#)

Firebase App Check

- Firebase App Check
- Firebase App Checkは、アプリが正当なものであることを確認して、不正なアクセスを防ぐためのFirebaseが提供しているセキュリティ機能
- 各プラットフォームで保護を行うことができる
 - iOS
 - DeviceCheck
 - Android
 - Play Integrity
 - Web
 - reCAPTCHA v3

DeviceCheck

- DeviceCheck
- Appleが提供するフレームワークで、Appleのサーバーと通信して端末情報を取得して不正なアクセスを防ぐのに使用されている
- 各iOS端末でのアプリをインストール、アンインストールの情報をApple側で管理しており、フレームワークを使用して、その情報を元にトークンを生成、認証の仕組みを行うことができる
- 正式にアプリをインストールした端末からしかトークンが発行できないので、クローリングやbotからのアクセスを防ぐことができる

Play Integrity

- Play Integrity
- Googleが提供するフレームワークで、Googleのサーバーと通信して端末情報を取得して不正なアクセスを防ぐのに使用されている
- 基本的にはDeviceCheckのAndroid版
- Play Integrityを使用することで、現在アクセスしているアプリがPlay Storeからインストールされたものか判別できる
- AndroidはiOSよりも簡単に野良アプリをインストール可能なので、そういったアプリからのアクセスを防ぐことができる

Firebase App Checkの機能

- 以下のリソースに対して保護を行うことができる
 - 各Firebaseのリソース
 - Storage
 - Functions
 - Firestore
 - Authentication
 - etc
 - ※モニタリングの情報をDEMOで表示
 - Firebase以外のリソースの場合は実装すれば保護が可能
 - カスタム バックエンドから App Check トークンを検証する

Firebase App Checkのデバッグトークン①



- App Checkを実装するとストアからインストールされたアプリしかアクセスできなくなるので、そのままだと開発時にアクセスできなくなってしまう
- そのため、開発時はデバッグトークンを使用してアクセスできるようにする

Firebase App Checkのデバッグトークン②

```
await FirebaseAppCheck.instance.activate(  
  androidProvider:  
    kReleaseMode ? AndroidProvider.playIntegrity : AndroidProvider.debug,  
  appleProvider:  
    kReleaseMode ? AppleProvider.deviceCheck : AppleProvider.debug,  
);
```

- デバッグトークンはProviderを**debug**にすることで有効になる
- アプリを起動するとデバッグトークンが発行されて、コンソールに設定するとアクセスが可能になる
- ※デバッグトークンの表示は以下を参考
 - FlutterでFirebase App Checkに対応 (+Firebase App Distributionで配信時の対応)

App Checkの実装①

- 実装環境
 - アプリ: Flutter
 - バックエンド: NestJS
- 事前準備
 - DeviceCheckの導入
 - 参考: [【Flutter】App Check を導入して Firebase を守ろう！\(DeviceCheck\)](#)
 - Play Integrityの導入
 - 参考: [【Flutter】App Check を導入して Firebase を守ろう！\(Play Integrity\)](#)

App Checkの実装②

- PR
 - Firebase App Check実装
- backendの実装
 - 対象コード
- アプリの実装
 - 対象コード1
 - 対象コード2

App Checkの実装③

App Checkを実装した状態でトークンなしでAPIにアクセスするとエラーが発生してアクセスできない

Query	Pre-request	Post-request	Auth	Result	Response headers
<div><div>▶ (Send query Me)</div><pre>1 query Me { 2 me { 3 id 4 uid 5 } 6 }</pre></div>				<div>200 OK ⌚ 311ms 🗑 Clear</div> <pre>1 { 2 "data": { 3 "me": null 4 }, 5 "errors": [6 { 7 "message": "App Check token not found", 8 "locations": [9 { 10 "line": 2, 11 "column": 3 12 } 13], 14 "path": [15 "me" 16] 17 } 18] 19 }</pre>	

まとめ

- Firebase App Checkを導入することで不正アクセスを防ぐことができる
- 開発時はデバッグトークンを使用してアクセスできる
- デバッグトークンはアプリを再インストールする毎に変更されるので、その度にコンソールに設定する必要があるので複数人開発では運用を考える必要がある
- [Firebase App Distribution](#)を使って配信した場合に、Androidだと**Play Integrity**でエラーにされてしまうので注意（今回の開発だとAndroid側は本番以外はデバッグトークンを使用して認証させた）

ご清聴ありがとうございました 🎉