ユニバーサルリンク/アプリリンクを使って
QRコードでゲストログインできるようにする

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

実装した機能の概要

- 以下のissueで実装
  - QRコードでユーザーを招待できるようにする
- 概要
  - ログインしているユーザーが招待用のQRコードを生成
  - QRコードを読み取ると、招待したユーザーがゲストログインできる
- モチベーション
  - アプリの使用にはアカウント作成が必要
  - アプリ的に家族と共有して使用したいユースケースが多い
  - その際にアカウント作成を求めるのはハードルが高いので、QRコードでゲストログインできるようにしたい
  - QRコードの読み取りはアプリ内、アプリ外どちらでも可能にしたい

実装に必要な技術

- backend/アプリのゲストログインの実装
- QRコード作成/スキャン
- QRコードをスキャンした時にアプリに指定の動作をさせる

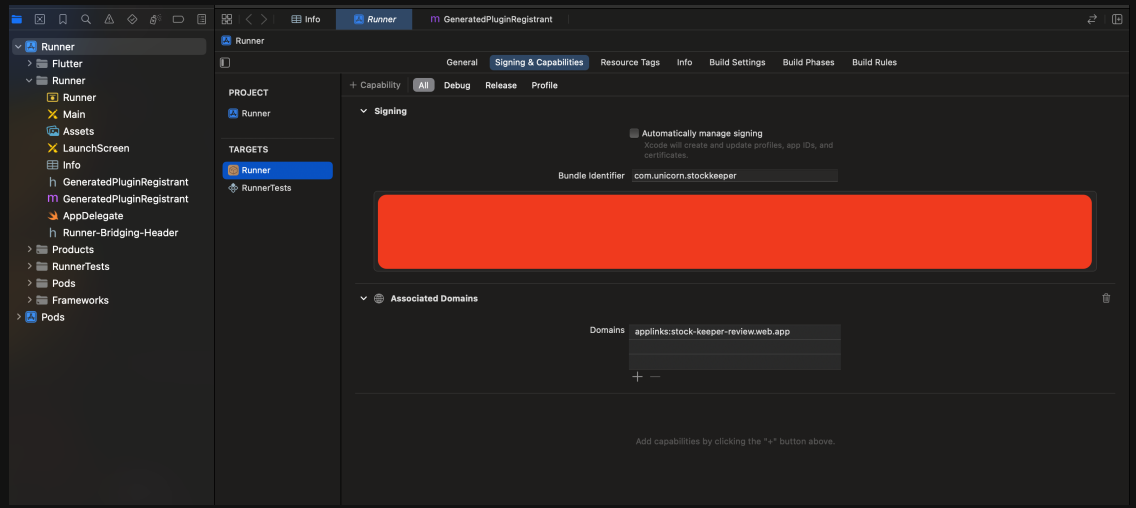
実装に必要な技術

- backend/アプリのゲストログインの実装
  - ゲストユーザーログインのトークンを発行してHeaderに設定して認証するミドルウェアを実装
- QRコード作成/スキャン
  - QRコード作成: qr_flutter
  - QRコードスキャン: mobile_scanner
- QRコードをスキャンした時にアプリに指定の動作をさせる
  - iOS: ユニバーサルリンク
  - Android: アプリリンク
  - 上記のリンクからアプリを起動するためのURLスキーム設定: go_router

ユニバーサルリンクとは

- Webサイトのリンクをクリックした時にiOSのアプリを起動させる仕組み
- Appleデバイス専用のディープリンクのためのプロトコル
- アプリを起動時にパラメータを付与することができるので、それを利用して特定の画面に遷移させることができる
- iOS 9以降で利用可能

ユニバーサルリンクの実装①



- XCodeでアプリを開いて、CapabilitiesのAssociated Domainsを追加
- Associated Domainsに`applinks:****.com`を追加

ユニバーサルリンクの実装②

- 先ほどのapplinksに設定をしたドメインの`.well-known/apple-app-site-association`で以下のjsonを設定
- ```
  {
    "applinks": {
      "apps": [],
      "details": [
        {
          "appID": "7KWTGL2ZDY.com.unicorn.stockkeeper",
          "paths": ["/"]
        }
      ]
    }
  }
  ```
- `appID`は`Apple Developerのteam id`と`bundle id`を組み合わせたものを指定
- これでiOS端末から`https://stock-keeper-review.web.app`を開いた時に、アプリがインストール済みの場合にアプリが起動できる
  ```

アプリリンクとは

- 先ほど説明したユニバーサルリンクのAndroid版
- Android 6.0以降で対応

アプリリンクの実装①

AndroidManifest.xmlに以下を追加

```
<intent-filter android:autoVerify="true">
  <action android:name="android.intent.action.VIEW" />
  <category android:name="android.intent.category.DEFAULT" />
  <category android:name="android.intent.category.BROWSABLE" />
  <data android:scheme="http" android:host="stock-keeper-review.web.app" />
  <data android:scheme="https" />
</intent-filter>
```

`android:host` にアプリに遷移させたいドメインを指定

アプリリンクの実装①

AndroidManifest.xmlに以下を追加

```
<intent-filter android:autoVerify="true">
  <action android:name="android.intent.action.VIEW" />
  <category android:name="android.intent.category.DEFAULT" />
  <category android:name="android.intent.category.BROWSABLE" />
  <data android:scheme="http" android:host="stock-keeper-review.web.app" />
  <data android:scheme="https" />
</intent-filter>
```

`android:host` にアプリに遷移させたいドメインを指定

アプリリンクの実装②

- 先ほどの`android:host`に設定をしたドメインの`.well-known/assetlinks.json`で以下のjsonを設定
- ```
  [
    {
      "relation": ["delegate_permission/common.handle_all_urls"],
      "target": {
        "namespace": "android_app",
        "package_name": "com.unicorn.stockkeeper",
        "sha256_cert_fingerprints": [
          "B7:FA:3A:2D:DA:29:22:6B:FF:02:40:55:69:E8:6D:8D:54:40:89:CE:69:8C:E6:E3:7C:FF:1B:9B:8E:82:EF:A8"
        ]
      }
    }
  ]
  ```
- `sha256_cert_fingerprints`は`アプリの署名証明書`のSHA-256ハッシュ値を指定
- これでAndroid端末から`https://stock-keeper-review.web.app`を開いた時に、アプリがインストール済みの場合にアプリが起動できる
  ```

go_routerでURLスキーマを定義

以下のようにgo_routerでURLスキーマを定義

```
final goRouter = GoRouter(
  initialLocation: '/',
  routes: [
    GoRoute(
        path: '/',
        name: "home",
        pageBuilder: (context, state) {
          return MaterialPage(key: state.pageKey, child: const AuthWrapper());
        },
        routes: [
          GoRoute(
            path: "guest/login/:code",
            name: "guest_login",
            pageBuilder: (context, state) {
              final code = state.pathParameters['code']!;
              return BottomSheetPage(builder: (_) => ShareBottomSheet(code: code));
            },
          ),
```

`https://stock-keeper-review.web.app/guest/login/xxxxxx`のURLでゲストログインが可能

go_routerでURLスキーマを定義

以下のようにgo_routerでURLスキーマを定義

```
final goRouter = GoRouter(
  initialLocation: '/',
  routes: [
    GoRoute(
        path: '/',
        name: "home",
        pageBuilder: (context, state) {
          return MaterialPage(key: state.pageKey, child: const AuthWrapper());
        },
        routes: [
          GoRoute(
            path: "guest/login/:code",
            name: "guest_login",
            pageBuilder: (context, state) {
              final code = state.pathParameters['code']!;
              return BottomSheetPage(builder: (_) ⇒ ShareBottomSheet(code: code));
            },
          ),
```

`https://stock-keeper-review.web.app/guest/login/xxxxxx`のURLでゲストログインが可能

go_routerでURLスキーマを定義

以下のようにgo_routerでURLスキーマを定義

```
final goRouter = GoRouter(
  initialLocation: '/',
  routes: [
    GoRoute(
        path: '/',
        name: "home",
        pageBuilder: (context, state) {
          return MaterialPage(key: state.pageKey, child: const AuthWrapper());
        },
        routes: [
          GoRoute(
            path: "guest/login/:code",
            name: "guest_login",
            pageBuilder: (context, state) {
              final code = state.pathParameters['code']!;
              return BottomSheetPage(builder: (_) => ShareBottomSheet(code: code));
            },
          ),
```

`https://stock-keeper-review.web.app/guest/login/xxxxxx`のURLでゲストログインが可能

動作検証

- QRコードに招待URLを設定
- ※PRにアプリ内でQRコードを読み取るデモと、アプリ外でQRコードを読み取るデモの動画を貼っている

まとめ

- ユニバーサルリンク/アプリリンクを使ってQRコードでゲストログインを作ることができた
- 設定箇所さえ分かれば、実装はそれほど難しくない
- ゲストの認証がHeaderに設定されたトークンのみ行っていてセキュリティ的には問題がある
- なので、この後Firebase App Checkを導入して端末以外からのアクセスを制限してカバーして機能完成予定

ご清聴ありがとうございました🎉