**WebviewでのApp Invokeハンドリング**

# 目的

 加盟店様Native Webviewアプリケーションにおいて、PayPayのWeb Cashierページ (App Invoke)を適切に取り扱えるようにするための設定方法を記載する。

※決済選択画面が決済代行会社様の画面でない実装を想定したものです。

# 現状の課題

 加盟店様Native Webviewアプリケーションが同じウインドウ内でPayPayのWeb Cashier URLをロードするとき、リダイレクト処理を適切にユーザに要求できないことが課題である。

# ソリューション

 加盟店様Native Webviewアプリケーションに対し、以下のスニペットを実装する。iOS, Androidはそれぞれ以下を実装する必要があります。

iOS: 「1」番を実装する。

Android: 「2」番を実装する。「2」番の実装が難しい場合は、「1」番を実装する。

1. iOS/Android共通:

Javascriptにより、外部ブラウザをオープンし、PayPayアプリを起動する。

 window.open("<web\_cashier URL>", "\_blank");

リクエストを実施する処理については、決済代行会社の仕様をご確認ください。

加盟店様から決済代行会社への遷移等を実施されるタイミングにて、外部ブラウザから開くようにご対応ください。

※ 本スニペットで開かれるブラウザはユーザのデフォルトブラウザとなります。

1. Android:

「Intent」を使用し、PayPayへのリダイレクトを実現する。

val codeWeb: WebView = WebView(this)

 val settings = codeWeb.settings

 settings.domStorageEnabled = true

 settings.javaScriptEnabled = true

 codeWeb.loadUrl("<merchant web URL>")

 codeWeb.webViewClient = object : WebViewClient() {

 override fun shouldOverrideUrlLoading(

 view: WebView?,

 url: String?

 ): Boolean {

 Log.i("shouldOverrideUrlLoading", "shouldOverrideUrlLoading: $url")

 val intent: Intent

 if (url != null) {

 if (url.contains("<web-cashier host>")) {

 // Current value of web cashier host:

 // <https://www.paypay.ne.jp/app/cashier>

 intent = Intent(Intent.ACTION\_VIEW)

 intent.data = Uri.parse(url)

 startActivity(intent)

 return true

 }

 }

 return false

 }

 }

#

# 問い合わせ先

 opa-salesengineer@paypay-corp.co.jp