**App Invoke Handling in Webview**

# Objective:

This document describes a setting method to properly handle PayPay’s Web Cashier page (App Invoke) on merchant’s Native Webview application. \* Assuming an implementation where the payment selection screen is not a collection agency’s screen.

# Current Issue:

When a merchant’s Native Webview application loads PayPay’s Web Cashier URL in the same window, redirect processing cannot be appropriately requested to user.

# Solution:

For merchant’s Native Webview application, implement a snippet as follows on iOS and Android respectively.

iOS: Implement #1.

Android: Implement #2. If it is difficult, implement #1.

1. Common to iOS and Android

Use JavaScript to open an external browser and start up PayPay App.

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

See collection agency’s specifications for the process to conduct a request.

Conduct setting to open from an external browser at the time of transition from merchant to collection agency.

\* The browser to be opened with this snippet is user’s default browser.

1. Android:

Use “Intent” to realize a redirect to PayPay.

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

}

}

# 

# Contact:

opa-salesengineer@paypay-corp.co.jp