

پرداخت درون برنامه‌ای کیپویی

رویه کلی پرداخت (غیر فنی)

۱. کتابخانه می‌بایست قبل از فراخوانی دستور پرداخت/انتقال وجه، با استفاده از اطلاعات مورد نیاز، پیکربندی شود. چنانچه در زمان فراخوانی دستور پرداخت/انتقال وجه، الزامات مورد نیاز کتابخانه برآورده نشده باشد، خطای مناسبی به برنامه‌نویس تحویل خواهد شد.
۲. نرم‌افزار (بنا به شرایط مورد نظر برنامه‌نویس) درخواست پرداخت/انتقال وجه مبلغ مورد نظر را به کتابخانه می‌دهد.
۳. کتابخانه با استفاده از اطلاعاتی که در مرحله پیکربندی دریافت نموده است، درخواست پرداخت/انتقال وجه را شبیه‌سازی کرده و کاربر را از نرم‌افزار، به مرورگر سافاری منتقل می‌کند؛ تا روند پرداخت/انتقال وجه توسط کاربر ادامه یابد.
۴. کاربر، در مرورگر، پس از مشاهده اطلاعات پرداخت (مانند مبلغ، دریافت‌کننده و ...) اقدام به تصمیم‌گیری جهت ادامه پرداخت، یا انصراف از پرداخت خواهد کرد.
۵. در صورتی که کاربر اقدام به پرداخت نماید، (پس از انجام امور مربوطه) در نهایت صفحه‌ای به کاربر نمایش داده خواهد شد که علاوه بر پیام‌های مناسب (براساس موفقیت یا بروز خطا)، شامل دکمه‌ای برای بازگشت به نرم‌افزار نیز خواهد بود.
۶. با زدن دکمه بازگشت به نرم‌افزار، اطلاعات حاصل از این تراکنش نیز به نرم‌افزار تحویل داده خواهد شد؛ و برنامه‌نویس براساس اطلاعات دریافتی، اقدامات لازم و مورد نظر خود را ادامه خواهد داد.
۷. بعدش، برنامه‌نویس، نرم‌افزار، کاربر، و خود ما، همگی خوش و خوشحال به زندگی‌مون ادامه خواهیم داد.

روند نصب و پیکربندی

برای نصب و استفاده از این ابزار، رویه مرسوم در برنامه‌نویسی iOS یا همان استفاده از Cocoapod (یکی از ابزارهای مدیریت وابستگی‌های پروژه) را مورد استفاده قرار دادیم.

۱. برنامه‌نویسان می‌توانند با استفاده از اضافه‌کردن عبارت 'KIPoiAPKit' pod به فایل Podfile پروژه، و نصب وابستگی‌ها با استفاده از دستور pod install، این کتابخانه را به پروژه اضافه نمایند.

۲. روند انتقال نتیجه تراکنش‌ها به نرم‌افزارها، با استفاده از Deep Linking انجام خواهد شد.

برای اینکار، شناسه نرم‌افزار یا همان Bundle Identifier بعنوان URLScheme مورد استفاده قرار خواهد گرفت؛ تا بدین صورت از غیرتکراری بودن URLScheme اطمینان حاصل شود. برای پیکربندی Deep Linking کد زیر را به فایل info.plist اضافه می‌کنیم:

```
<key>CFBundleURLTypes</key>
<array>
  <dict>
    <key>CFBundleURLName</key>
    <string>{{APP_BUNDLE_ID}}</string>
    <key>CFBundleURLSchemes</key>
    <array>
      <string>{{APP_BUNDLE_ID}}</string>
    </array>
  </dict>
</array>
```

۳. مرحله بعدی پیکربندی، پیکربندی کتابخانه درون نرم‌افزار خواهد بود:

۳.۱. برای استفاده از اطلاعات منتقل‌شده از مرورگر به نرم‌افزار، می‌بایست از متد Check(_ url:) استفاده نماییم. این متد را در قسمت مربوطه در فایل AppDelegate قرار خواهیم داد:

```

application(_ application: UIApplication, open url: URL, sourceApplication: String?,
annotation: Any) -> Bool {
    return KipolAP.Check(url)
}

```

چنانچه ساختار URL دریافتی این متد، صحیح (براساس ساختار و متغیرهای مورد انتظار

کتابخانه) باشد، براساس اطلاعات دریافتی، یکی از دو حالت زیر اتفاق خواهد افتاد:

۱.۱.۳. چنانچه روند پرداخت بدون بروز خطا به اتمام رسیده باشد، و توکن پرداخت تولید شده

باشد، متد زیر فراخوانی خواهد شد:

```

KipolAP.delegate?.kipoPayment(paymentToken:)

```

۱.۲.۳. چنانچه در روند پرداخت خطایی بروز نماید، متد زیر به همراه پیام خطای رخ داده،

فراخوانی خواهد شد:

```

KipolAP.delegate?.kipoPayment(errorMessage:)

```

۲.۳. قدم بعدی، استفاده از متد Setup(merchantKey:) خواهد بود. ورودی این متد، از نوع

String بوده، و برنامه نویس می بایست (پیش از فراخوانی درخواست پرداخت/انتقال وجه) شماره

همراهی را که با آن در سامانه کیپو ثبت نموده است را بعنوان پارامتر این متد تحویل دهد.

فراخوانی درخواست پرداخت/انتقال وجه

۱. برای فراخوانی درخواست پرداخت، برنامه نویس ابتدا می بایست مشخصه را مقداردهی نماید.

این کار مشابه مقداردهی delegate و یا dataSource المان UITableView انجام خواهد شد.

همانگونه که برای استفاده از UITableView، المانی بعنوان delegate در نظر گرفته می شود، و

متدهای اجباری UITableViewDelegate پیاده سازی می شوند، در اینجا هم باید مشخصه

delegate مربوط به KipolAP مقداردهی شده و متدهای آن نیز پیاده سازی شوند؛ با این کار

پروتکل KipolAPDelegate پیاده سازی خواهد شد. این پروتکل دارای سه متد اجباریست:

- متد (kipoCannotPerform(error:)) : زمانیکه انجام فراخوانی پرداخت امکان پذیر نباشد، به همراه خطای مربوطه، در اختیار برنامه نویس خواهد بود.
 - متد (kipoPayment(paymentToken:)) : پس از انجام موفقیت آمیز پرداخت، به همراه توکن پرداخت، در اختیار برنامه نویس خواهد بود.
 - متد (kipoPayment(errorMessage:)) : در صورت بروز خطا در روند پرداخت، به همراه پیام خطای رخ داده، در اختیار برنامه نویس خواهد بود.
- پیگربندی delegate نیز با استفاده از متد Setup(delegate:) صورت می پذیرد.
۲. فراخوانی درخواست پرداخت/انتقال وجه نیز با استفاده از متد Pay(amount: Int) انجام می شود که تنها یک ورودی، از نوع عددی (Int) دارد، که مشخص کننده مبلغ پرداختی به ریال خواهد بود.