ご紹介

ActiveSDKドキュメントの目的はGPaymentsの3-Dセキュア2 3DS SDKを既存の3DSリクエスターアプリ、すなわち加盟店のモバイルアプリ、に統合するために必要な知識とリソースをiOSまたはAndroidの開発者に提供することです。そうすることにより、カード会員に対してシームレスにEコマースの認証エクスペリエンスを提供できるようになります。

本ドキュメントはActiveSDKの紹介を通して統合プロセスをガイドし、トラブルシューティングの手順を提供します。

ActiveSDKの紹介

GPayments ActiveSDKへようこそ！

ActiveSDKはGPayments Pty Ltdにより提供されている3-Dセキュア2ソリューションです。

EMVCoスペックに準拠し、ActiveServerと並行して開発されたActiveSDKは、3-Dセキュア認証のネイティブモバイルサポートを提供しており、AndroidとiOSの両方で利用が可能です。当社のActiveSDKの利用でアプリ開発者が簡単にあらゆる3DS Serverに接続して3-Dセキュア認証を完了することができます。

モバイルコマースは世界中で益々普及してきています。モバイルデバイスを介した支払額は2019年には1兆米ドルに達すると見込まれており、モバイルウォレットは2020年までにデビットカードやクレジットカードよりも一般的になると予想されています。一方、モバイルアプリを介して製品を購入し、多数のオンラインサービスにログインするために多くのデバイスが使用されていることから、モバイル詐欺のリスクもこれまでになく高まっています。

当社のActiveSDK（AndroidとiOS）は、リスクベース認証によるモバイルデバイスとACS間での豊富なデータ交換を可能にし、取引を許可するか、チャレンジするかの即時決定を可能にしま

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

す。またActiveSDKはネイティブ認証画面も提供します。これにより、認証時のインターフェイスが加盟店のモバイルアプリケーションとシームレスに統一され、その他のアプリ内購入プロセスと一貫性のあるルック・アンド・フィールを提供します。ActiveSDKはデフォルトでActiveServerと統合されており、貴社のアプリケーションに迅速に統合するためのサンプルコード付きデモアプリも含まれています。

ActiveSDKは **iOS** や **Android** アプリと統合することにより、ユーザーに対してモバイル決済認証のネイティブエクスペリエンスを提供することができます。

当社の３DSサーバーソリューション（ActiveServer）についてのドキュメントは下記のリンクよりご覧いただけます。

- ActiveServerドキュメント日本語版

対応バージョン

下記のテーブルはActiveSDKが対応する最低バージョンを示します：

OS	最低バージョン
iOS	iOS 9
Android	4.4 (API Level 19)

> **Note**
>
> Windows 10 Mobileは対応されていません。

対応するEMVCo 3-Dセキュアのプロトコル・バージョン

ActiveSDK (iOSとAndroid両方)は下記のEMVCo 3-Dセキュアプロトコルバージョン番号をサポートしています。

プロトコル・バージョン	ステータス
2.1.0	対応

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

プロトコル・バージョン	ステータス
2.2.0	対応

知るべきコンセプト

ActiveSDKはEMVCo 3-Dセキュアモバイル SDKスペックに準拠したものになっています。 3-Dセキュア認証のコンセプトに関する以下の文書を本書とともに理解していただくと開発がより容易で、かつGPayments ActiveSDKのケーパビリティをフル活用することができます。

- EMV® 3-D Secure Protocol and Core Functions Specification.pdf （訳：EMV® 3-Dセキュアプロトコルとコア機能スペック.pdf）
- EMV® 3-D Secure—SDK Specification.pdf （訳：EMV® 3-DセキュアSDKスペック.pdf）
- 用語集

統合

統合セクションには、ActiveSDKをAndroidおよびiOSアプリへと統合する方法に関する詳細なチュートリアルが含まれています。すべての3DS SDK認証プロセスとそのワークフローについて説明しています。

- 統合セクション

3DSリクエスター

ActiveSDKを統合するには、加盟店サイトはバックエンドに3DSリクエスターを実装する必要があります。このセクションでは、Active SDKを3DSリクエスターデモコードと統合するプロセスについて説明します。このデモでは、3DSリクエスターが接続した3DSサーバーは 弊社のActiveServerです。

- 3DSリクエスターセクション

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

APIレファレンス

APIレファレンスセクションには、ActiveSDK(iOSとAndroidの両方)のすべてのAPIがリストされています。

さらに、アプリ・フローのAndroidおよび iOS用の3DSサーバー (ActiveServer) APIリファレンスは次の場所で入手できます。

- ActiveServer 認証APIレファレンス - `/api/v2/auth/app`
- ActiveServer 認証APIレファレンス - `/api/v2/auth/app/result`

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

統合ガイド

このセクションでは、ActiveSDKを使ったAndroidおよびiOSアプリの詳細な開発チュートリアルをご紹介します。すべてのSDKでの3DS認証プロセスおよびそのワークフローを網羅しております。

セキュリティガイドおよびベストプラクティス

3DS SDKを使った開発を進める前に、セキュリティ面の影響や機微情報を保護する方法を検討いただくことが大切です。EMVCo 3DセキュアSDK用仕様の指定に準拠して、SDK側でも多くのセキュリティ事項を考慮しておりますが、その他にも実装時に考慮すべき点はございます。

3DSサーバとの通信

3DS サーバとの通信は 3DS SDKではスコープ外となっておりますが、3DS SDK から生成されたデータおよびその他の方法で送信されたデータは、決済システムのセキュリティ基準に従って適切に保護される必要があります。EMVCo Protocol and Core Functions SpecificationのAnnex Dには、以下のようなガイダンスが記載されています。これらの規格は、3DS SDKと3DSサーバ間の通信に適しています。

TLSは1.2以上を利用し、鍵長は下記に準拠すること。

- **RSA:** 2048ビット以上。
- **ECC:** 256ビット以上。

下記のいずれかの暗号スイートを利用すること。

- **TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256**
- **TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256**

楕円曲線暗号P-256 を使用し、暗号スイート拡張子に表示すること。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

カード会員の機微情報の保護

認証に用いられるデータの多くは3DS SDKのスコープ範囲外であり、認証パラメータの外で3DSサーバーに送信されます。こうした機微情報にはカード番号やカード会員の詳細な連絡先も含まれます。

機微情報の保護には、上記データの通信の安全性担保のほか、デバイス自体の保護も含まれます。そのためには、データをローカルに保存しない、あるいは保存する場合は適切に暗号化する、などの対策が考えられます。

サービス用インスタンスのクリーンアップ

3Dセキュアが完了した後はSDKを欠かさずクリーンアップしてください。下記の `cleanup` メソッドがご使用いただけます。

Android iOS

```
ThreeDS2Service.INSTANCE.cleanup(applicationContext);
```

ダウンロードとアップデート

3DS SDKへのアップデートはGPaymentsのレポジトリからご利用いただけます。ビルドは定期的に更新され、また重要なセキュリティ更新が行われた場合にも更新されます。ビルド更新時は、SDKを用いて開発しているお客様にメールで通知いたします。

デプロイされたアプリケーションに直接変更が反映されるわけではないので、お客様側で最新のビルドを取得し、アプリケーションに統合することが必要になります。アップデートの際には、すでに進行中のトランザクションの整合性を確保にご注意ください。

セキュリティの更新

`Security Checks` セクションには、SDKの初期化時に行われるセキュリティチェックの詳細と、ランタイム時に定期的に行われるチェックの内容が記載されています。アプリケーションがこれらの問題を検知し、適切に対応できるよう、このセクションの記載に従って実装していただく必要がございます。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

3DS SDKの流れ

ActiveSDKを使用することで、モバイルアプリケーションで3Dセキュア2の認証をサポートすることができます。本SDKは、EMVCoが公開している3DS V2 (2.1.0, 2.2.0) の仕様に準拠して構築されています。EMVの3Dセキュア仕様、および3DS V2全般に関する詳細は、EMVCoのウェブサイトをご確認ください。

3Dセキュア認証における3DS SDKの呼び出しシークエンスの全体像については、認証シーケンスページをご参照ください。

はじめに

SDKのバイナリはGPaymentsのリポジトリから入手可能です。リポジトリの情報は登録手続き時にご提供いたします。

インストール時に含まれる3DS SDKには2つのバージョンがあります。1つは開発バージョン、もう1つは本番バージョンです。本番バージョンには強力なセキュリティ制約を設けているため、シミュレーターやエミュレーターやお使いの開発環境のデバッガーを利用した上での実行などの一般的な開発プロセスでは実行できません。

PCI 3DS SDKは、3DS SDKに対して、開発を厳しく制限するような要件を設けています。例えば、エミュレータ上での実行やデバッグは、本番用3DS SDKでは禁止されています。そのため、GPayments 3DS SDK には開発バージョンと本番バージョンがあり、主な違いは実行されるセキュリティチェックとその結果の動作に関連するものです。詳細は、セキュリティチェックのセクションをご参照ください。

Android

以下のコードスニペットでは、お客様のAndroidアプリケーションにGPaymentsリポジトリを追加する方法をご紹介しております。

> **注意**
>
> 現在Android SDKはAndroidXを使ってビルドされています。プロジェクトでAndroidXまたはそれより古いAndroidサポートライブラリをお使いであることをお確かめください。Android Studio 3.2以上の場合、メニューバーの *Refactor -> Migrate to AndroidX* を選択して既存プロジェクトをAndroidXに移行していただけます。

1. 下記のようにメイン `build.gradle` に `repositories` 属性を追加してください。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Gradle

```
repositories {
    maven {
        url "https://repository.gpayments.com/repository/mvn-sdk"
        credentials {
            username = "put your username here"
            password = "put your password here"
        }
    }
}
```

2. アプリケーションの `build.gradle` ファイル内の `dependencies` に以下の行を追加してください。

Production SDK Development SDK

```
dependencies {
   ...
   // Production SDK
   implementation 'com.gpayments.3ds2:activesdk-android:<version>@aar'
   ...
}
```

各リリースのバージョン番号については、当ドキュメントの リリースノート ページ内の **SDKビルドバージョン番号** をご参照ください。

iOS

iOS用SDKのバイナリを入手するには、GPaymentsのリポジトリサーバー経由とCocoaPods経由の2つの方法があります。 リポジトリサーバから取得する方法と、CocoaPods経由で取得する方法があります。

GPaymentsリポジトリからダウンロードする

GPaymentsリポジトリからフレームワークバイナリをダウンロードし、以下の手順を実施してください。 :

1. お使いのプロジェクト用のディレクトリにフレームワークのディレクトリを配置してください
2. Xcode上で、お使いのプロジェクトをクリックしてください。その後にリンクさせたい対象をクリックしてください。最後にBuild Phasesをクリックしてください。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

3. Link Binary With Librariesセクションを展開してください。
4. +をクリックしてください。
5. その後にAdd Otherをクリックし、Add Filesを選択してください。
6. フレームワークディレクトリを設定してください。
7. Openをクリックしてください。

CocoaPods経由でインストールする

CocoaPodsを既にインストールされている場合は、次のコマンドを実行してください。：

```
$ sudo gem install cocoapods
```

その後、 プロジェクトのルートディレクトリに移動して `$ pod init` コマンドを実行してください。

ルートフォルダには生成されたPodfileが入っています。ここでは、プロジェクトに追加したいポッドをすべてリストアップすることができます。現時点利用可能なPodファイルは下記の通りです。：

```
ActiveSDKDev
ActiveSDKProd
ActiveSDKSwift4.2Dev
ActiveSDKSwift4.2Prod
```

各Podは、Swiftがサポートしているバージョン用のXCFrameworkとしてアップロードされます。XCFramework は、すべてのターゲットプラットフォームとアーキテクチャに対して単一の依存関係を持つので、以前のバージョンのように、個々のデバイスやシミュレーターといったプラットフォームを個別に指定する必要はございません。

ファイル名にSwiftのバージョンが指定されていない限り、すべてのPodファイルはSwift 5でコンパイルされています。**ActiveSDKSwift4.2Dev**と**ActiveSDKSwift4.2Prod**はSwift 4でコンパイルされています。

ActiveSDKのPodを使うには、以下の例のようにお使いのPodファイルを初期化する必要があります。：

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

```
source 'https://github.com/gpayments/activesdk-ios.git'

target 'ExampleProjectName' do
  pod 'ActiveSDKDevSimulator', '~> 2.2.7889'
end
```

各リリースでどのバージョン番号を使うべきかは、リリースノートの**SDK build version numbers**をご確認ください。

> **注意**
>
> GitHubにCocoaPodが設定されているので、CocoaPodリポジトリにアクセスする際にはお好きな方法（HTTPSまたはSSH）をご利用ください。SSHをご利用の際は、podfileの最初の行を `source 'git@github.com:gpayments/activesdk-ios.git'` に書き換えてください。
>
> SSHを使う前に、お使いのGitHubアカウントにSSHキーを紐づける必要があります。もしキーを取得されていない場合は、GitHubのドキュメントをご参照の上設定してください。

以下のコマンドを実行する前に、コマンドライン上でGPaymentsからプライベートリポジトリへのアクセス権を付与されたGitのアカウントに必ずログインしてください。

アクセスにはmacOSのキーチェーンや Git Credential Manager Core をお使いいただけます。あるいはGit の設定が正しいアカウントに設定されているかどうか確認していただくだけでもかまいません。プライベートリポジトリへのアクセス権が与えられていないアカウントを使用している場合、 `No `Podfile' found in the project directory.` エラーがスローされます。

> **重要**
>
> プライベートリポジトリへのアクセス権限を持つGitアカウントがない場合は、ActiveSDKプロバイダにご連絡ください。

最後に、 `$ pod install` をルートフォルダ内で実行すると、podがダウンロードされ、お使いのプロジェクトに統合されます。

SDKの初期化

3DS Requestorアプリケーションの実装によりますが、 `ThreeDS2Service` インスタンスは、3DS Requestorアプリのスタートアップ時（バックグラウンドタスクとして）、またはトランザクションが開始されたときに生成されます。サービスの状態は、 `cleanup` メソッドが呼ばれるま

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

で維持されます。 `ThreeDS2Service` のインスタンスを生成するには、以下のように `initialize` メソッドをコールする。:

Android iOS

```
ThreeDS2Service.INSTANCE.initialize(applicationContext,
    configParameters,
    locale,
    uiCustomization,
    new MyClientEventListener(),
    new MySecurityEventListener());
```

初期化メソッドで必要になるパラメータは以下のとおりです。:

Android用パラメータ

パラメータ名	型	説明
applicationContext	Context	AndroidアプリケーションのContextのインスタンスです。例えば `MainActivity.this` のように、現在のアクティビティのインスタンスにすることができます。
configParameters	ConfigParameters	初期化時に使用する設定情報。詳しくは、Config Parameters の項を参照してください。
locale	String	アプリのユーザーインターフェースのロケールを表す文字列（例："en-US"）。将来的に使用します。
uiCustomization	UiCustomization	3DS SDKのUI要素（フォント、色など）をカスタマイズするためのクラスです。詳しくは、UIカスタマイズの項を参照してください。
clientEventListener	ClientEventListener	SDKのイベントを受け取るリスナー（ロギング用など）です。 詳細は、Client Event Listener の項を参照してください。
securityEventListener	SecurityEventListener	SDKの初期化時および実行時に発生するセキュリティ警告イベントを受信するためのリスナーです。詳細は、セキュリティ イベント リスナー のセクションを参照してください。

iOS用パラメータ

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

パラメータ名	型	説明
configParameters	ConfigParameters	初期化時に使用する設定情報です。詳しくは、Config Parameters の項を参照してください。
locale	String	A string that represents the locale for the app's user interface (e.g. "en-US"). For future use.
uiCustomization	UiCustomization	3DS SDKのUI要素（フォント、色など）をカスタマイズするためのクラスです。詳しくは、UIカスタマイズの項を参照してください。
clientEventListener	ClientEventListener	SDKのイベントを受け取るリスナー（ロギング用など）です。 詳細は、Client Event Listener の項を参照してください。
securityEventListener	SecurityEventListener	SDKの初期化時および実行時に発生するセキュリティ警告イベントを受信するためのリスナーです。詳細は、セキュリティ イベント リスナー のセクションを参照してください。

Config Parameters

`ConfigParameters` クラスは、3DS SDK が初期化のために必要とする詳細や、追加のオプション設定を提供するために使用されます。 `ConfigParameters.Builder` クラスは、これらのパラメータを整理するためのクラスです。次のように使用することができます。:

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Android iOS

```
//adding the license
String runtimeLicense = ".....";


List<ConfigParameters.DirectoryServerInfo> directoryServerInfoList = new
ArrayList<>();
directoryServerInfoList.add(new
ConfigParameters.DirectoryServerInfo(DSInfo.DS_ID, DSInfo.DS_CERT,
DSInfo.DS_CA_CERT));

List<String> deviceParameterBlacklist = new ArrayList<>();
    deviceParameterBlacklist.add("A009");
    deviceParameterBlacklist.add("A010");

List<String> maliciousApps = new ArrayList<>();
    maliciousApps.add("de.robv.android.xposed");

List<String> trustedAppStores = new ArrayList<>();
    trustedAppStores.add("com.xiaomi.market");

String appsignature = ParseAppSignature(HTTP.Get("https://www.example.com/
signature"));

List<String> clientConfigs = new ArrayList<>();
    clientConfigs.add("logLevel=3");
    clientConfigs.add("MaskSensitive=false");

ConfigParameters configParameters = new
ConfigParameters.Builder(directoryServerInfoList, runtimeLicense)
    .clientConfig(clientConfigs)
    .build();
```

Note

DSInfoクラスの実装については下部 でご確認いただけます。

パラメータは下記のグループに分かれます。:

Android用パラメータ

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

パラメータ名	型	説明
DirectoryServerInfo	List	サポートされているディレクトリサーバーの暗号化キーと証明書を指定すれます。詳細は、ディレクトリサーバー情報のセクションをご確認ください
RuntimeLicense	String	3DS SDKライセンスです。
DeviceParameterBlacklist	List	デバイスからの取得対象外とするデバイスパラメータのリスト。デフォル可能な限り多くのデバイスパラメータを取得します。この設定グループを定のパラメータを取得しないようにSDKに指示することができます。このメータを追加するには、識別子を指定します。パラメータの完全なリストら直接入手できる `EMV® 3-D Secure SDK Device Information` をご確認く
MaliciousApps	List	悪質と認識されるアプリの一覧です。このリスト内のアプリがインストー合、セキュリティ警告（SW02）が発生します。デフォルトでは、以下のアあるとみなされます： `de.robv.android.xposed` , `de.robv.android.xpose` `com.saurik.substrate` .
TrustedAppStores	List	信頼できるアプリストアからインストールされていない場合、セキュリテが発生します。デフォルトでは、Google Playストア（ `com.android.vendi` 頼されています。このパラメータを使用して、信頼できるアプリストアをことができます。
AppSignature	List	この値がアプリの署名と一致しない場合、セキュリティ警告（SW02）が発のプロパティには、アプリの署名に使用された証明書のSHA256フィンガー定する必要があります。この値が指定されていない場合、SDKはその署名ます。セキュリティ上の理由から、この値をアプリにハードコードしてはわりに、フィンガープリントをサーバーに保存し、実行時に取得し、ここす。
ClientConfig	List	SDKの追加設定に使用されるコンフィギュレーション設定です。利用可能な一覧は、Client Config セクションで確認することができます。

`ConfigParameters` オブジェクトをビルドした後、 `addParams` メソッドを使用して追加のパラメータを追加することができます。このメソッドには、グループ、パラメータ名、パラメータ値を指定します。現時点では、SDKに関連する特定のグループは存在しないため、グループにはnullを設定することができます。例えば、以下のようになります。:

```
configParameters.addParam(null, "ParameterName", "ParameterValue");
```

追加可能なパラメータは以下のとおりです。:

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

名前	説明
ShowWhiteBoxInProcessingScreen	処理アイコンとディレクトリサーバーの画像の後ろにホワイトボックスを表示します。このパラメータにはブーリアン文字列値（"True "または "False"）のどちらかを設定できます。
UseDefaultTrustedAppStoreList	この設定は、デフォルトの信頼できるアプリストアのリスト (例: com.android.vending) を使用するかどうかを制御します。デフォルトではTrueですが、Falseに設定すると、初期化時に `TrustedAppStores` リストで明示的に指定したストアのみが信頼されるようになります。このパラメータにはブーリアン文字列値（"True "または "False"）のどちらかを設定できます。
ProgressBarColor	この設定は、 `getProgressView` メソッドが返すプログレスダイアログに表示されるプログレスバーの色をコントロールします。16進数のカラーコード値 (例: "#b884f2") を設定してください。

iOS用パラメータ

パラメータ名	型	説明
DirectoryServerInfo	List	対応するさまざまなディレクトリサーバーの暗号化キーと証明書を指定するために使用します。詳細は、ディレクトリサーバー情報に記載されています。
RuntimeLicense	String	3DS SDKライセンスです。
DeviceParameterBlacklist	List	デバイスからの取得対象外とするデバイスパラメータのリスト。デフォルトでは、SDKは可能な限り多くのデバイスパラメータを取得します。この設定グループを使用すると、特定のパラメータを取得しないようにSDKに指示することができます。このリストにパラメータを追加するには、識別子を指定します。パラメータの完全なリストは、EMVCoから直接入手できる `EMV® 3-D Secure SDK Device Information` をご確認ください。
ClientConfig	List	SDKの追加設定に使用されるコンフィギュレーション設定です。利用可能なオプションの一覧は、Client Config セクションで確認することができます。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

パラメータ名	型	説明
AppBundleID	String	アプリケーションのバンドル識別子を指定します。これは、アプリケーションのビルド時に指定されたバンドル識別子のID設定と一致する必要があります。この値が実行時にアプリケーションのバンドル ID と一致しない場合、セキュリティ警告が発生します。この値が指定されていない場合、SDKはバンドルIDが無効であると見なします。セキュリティ上の理由から、この値のアプリへのハードコードは避けてください。代わりに、バンドルIDはサーバーに保存され、実行時に取得され、ここで設定してください。

`ConfigParameters` オブジェクトをビルドした後、 `addParams` メソッドを使用して追加のパラメータを追加することができます。このメソッドには、グループ、パラメータ名、パラメータ値を指定します。現時点では、SDKに関連する特定のグループは存在しないため、グループにはnullを設定することができます。例えば、以下のようになります。:

```
try configParameters.addParam(group: nil, paramName: "ParameterName", paramValue: "ParameterValue")
```

追加可能なパラメータは以下のとおりです。:

名前	説明
ShowWhiteBoxInProcessingScreen	処理アイコンとディレクトリサーバーの画像の後ろにホワイトボックスを表示するか否かを制御します。このパラメータにはブーリアン文字列値（"True "または "False"）のどちらかを設定できます。
FireChallengeStatusFirst	このパラメータで、completed、cancelled、protocolError Challenge Status Receiverのイベントがそれぞれいつ発生するかを設定することができます。デフォルト（"False"）では、チャレンジUIが閉じた後にこれらのイベントが発生します。True" に設定すると、チャレンジUIが閉じる前にこれらのイベントが発生します。

UIカスタマイズオプション

`UICustomization` クラスは、3DS SDKのUI要素をカスタマイズするための機能を公開します。これにより、3DS Requestorアプリケーションは、ネイティブのチャレンジページの外観を調整できます。ボタン、ラベル、テキストボックス、ツールバーは、以下の様々なクラスとサブクラスでカスタマイズできます。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

UIのカスタマイズ

UI要素のカスタマイズ用のメインクラスです。

Android iOS

```
public class UiCustomization {
    enum ButtonType (VERIFY, CONTINUE, NEXT, CANCEL, RESEND)
    enum LabelType (INFO_HEADER, INFO_TEXT, INFO_LABEL, WHITELIST, WHY_INFO,
WHY_INFO_TEXT, EXPANDABLE_INFO, EXPANDABLE_INFO_TEXT, SELECTION_LIST)

    void setButtonCustomization(ButtonCustomization buttonCustomization,
ButtonType buttonType)
    void setToolbarCustomization(ToolbarCustomization toolbarCustomization)
    void setLabelCustomization(LabelCustomization labelCustomization)
    void setTextBoxCustomization(TextBoxCustomization textboxCustomization)
    void setBackground(String hexColorCode)
    void setInformationZoneIconPosition(int position) // 0: right, 1: left, 2:
hide

    ButtonCustomization getButtonCustomization(ButtonType buttonType)
    ToolbarCustomization getToolbarCustomization()
    LabelCustomization getLabelCustomization()
    TextBoxCustomization getTextBoxCustomization()
    String getBackground(String hexColorCode)
    int getInformationZoneIconPosition(int position)
}
```

ボタンのカスタマイズ

チャレンジプロセス中の各種ボタンのカスタマイズは `ButtonCustomization` クラスで行います。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Android iOS

```
public class ButtonCustomization {
    void setBackgroundColor(String hexColorCode)
    void setCornerRadius(int cornerRadius)
    void setTextFontName(String fontName)
    void setTextFontSize(int fontSize)
    void setTextColor(String hexColorCode)
    void setHeight(int pixels)
    void setPadding(int start, int top, int end, int bottom) // applicable only
for ButtonType of CANCEL

    String getBackgroundColor()
    int getCornerRadius()
    String getTextFontName()
    int getTextFontSize()
    String getTextColor()
    int getHeight()
    int[] getPadding() // applicable only for ButtonType of CANCEL
}
```

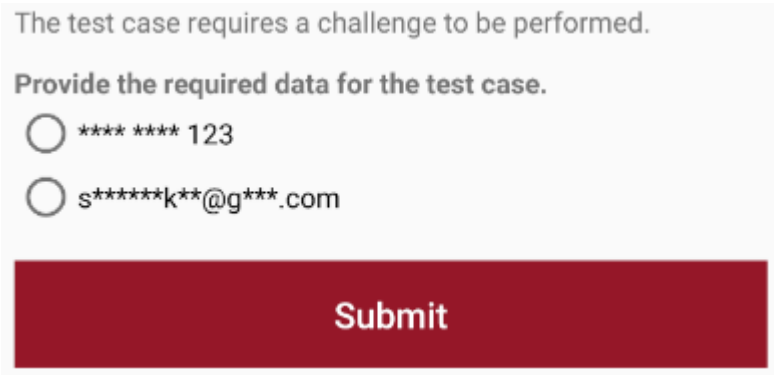
例として、下記のコードでは赤地に白のフォントで表示されます。:

Android iOS

```
UiCustomization uiCustomization = new UiCustomization();

ButtonCustomization customButton = new ButtonCustomization();
customButton.setTextColor("#FFFFFF"); // White
customButton.setBackgroundColor("#951728"); // Dark red

uiCustomization.setButtonCustomization(customButton,
UiCustomization.ButtonType.CANCEL);
uiCustomization.setButtonCustomization(customButton,
UiCustomization.ButtonType.CONTINUE);
uiCustomization.setButtonCustomization(customButton,
UiCustomization.ButtonType.NEXT);
uiCustomization.setButtonCustomization(customButton,
UiCustomization.ButtonType.RESEND);
uiCustomization.setButtonCustomization(customButton,
UiCustomization.ButtonType.SUBMIT);
```

チャレンジ画面の `Submit` ボタンは以下のようになります。:

Copyright ©2019 GPayments Pty Ltd. All rights reserved.



ラベルのカスタマイズ

チャレンジプロセス中の各種ラベルは `LabelCustomization` クラスでカスタマイズできます。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Android iOS

```
public class LabelCustomization {
    void setHeadingTextAlignment(int textAlignment)
    void setHeadingTextColor(String hexColorCode)
    void setHeadingTextFontName(String fontName)
    void setHeadingTextFontSize(int fontSize)
    void setInputTextColor(String hexColorCode)
    void setInputTextFontName(String fontName)
    void setInputTextFontSize(int fontSize)
    void setTextColor(String hexColorCode)
    void setTextColor(LabelType labelType, String hexColorCode)
    void setTextFontName(String fontName)
    void setTextFontName(LabelType labelType, String fontName)
    void setTextFontSize(int fontSize)
    void setTextFontSize(LabelType, labelType, int fontSize)
    void setBackgroundColor(LabelType labelType, String hexColorCode)
    void setPadding(LabelType labelType, int start, int top, int end, int
bottom)

    int getHeadingTextAlignment()
    String getHeadingTextColor()
    String getHeadingTextFontName()
    int getHeadingTextFontSize()
    String getInputLabelTextColor()
    String getInputLabelTextFontName()
    int getInputLabelTextFontSize()
    String getTextColor()
    String getTextColor(LabelType labelType)
    String getTextFontName()
    String getTextFontName(LabelType labelType)
    int getTextFontSize()
    int getTextFontSize(LabelType labelType)
    String getBackgroundColor(LabelType labelType)
    int[] getPadding(LabelType labelType)
}
```

テキストボックスのカスタマイズ

チャレンジプロセス中の各種テキストボックスは `TextBoxCustomization` クラスでカスタマイズできます。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Android iOS

```
public class TextBoxCustomization {
    void setBorderColor(String hexColorCode)
    void setBorderWidth(int borderWidth)
    void setCornerRadius(int cornerRadius)
    void setTextColor(String hexColorCode)
    void setTextFontName(String fontName)
    void setTextFontSize(int fontSize)

    String getBorderColor()
    int getBorderWidth()
    int getCornerRadius()
    String getTextColor()
    String getTextFontName()
    int getTextFontSize()
}
```

ツールバーのカスタマイズ

チャレンジプロセス中の各種ツールバーは `ToolbarCustomization` クラスでカスタマイズできます。

Android iOS

```
public class ToolbarCustomization {
    void setBackgroundColor(String hexColorCode)
    void setButtonText(String buttonText)
    void setHeaderText(String headerText)
    void setTextColor(String hexColorCode)
    void setTextFontName(String fontName)
    void setTextFontSize(int fontSize)

    String getBackgroundColor()
    String getButtonText()
    String getHeaderText()
    String getTextColor()
    String getTextFontName()
    int getTextFontSize()
}
```

アイコンのカスタマイズ

iOSのみ

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

上記の UI Customization クラスによる UI の外観のカスタマイズと同時に、iOSではSDK が使用する画像もカスタマイズ可能です。画像は7種類あり、以下の名称のBundle of imagesでカスタマイズ可能です。:

- **CollapsedLabelIndicator** セクションが折り畳み可能であることを示すアイコン。
- **ExpandedLabelIndicator** セクションが展開可能であることを示すアイコン。
- **MultiSelection** 複数選択オプション用のアイコン。
- **MultiSelectionSelected** 選択済複数選択オプション用のアイコン。
- **SingleSelection** 単一選択オプション用のアイコン。
- **SingleSelectionSelected** 選択済単一選択オプション用のアイコン。
- **WarningIndicator:** （ACSで示されている通り）チャレンジ警告テキストを表示するためのアイコン。

バンドルに含んだ画像は下記のようにThreeDS2Serviceコンストラクタに設定します。:

```
let threeDS2Service = ThreeDS2Service(bundle: Bundle.main)
```

ダークモード

3DS SDKは、システムレベルで構成された設定に基づき、ダークモードをサポートしております。コード上での変更は必要ありません。ダークモードの画像もそれぞれ設定することができます。

クライアントイベントリスナー

`ClientEventListener` クラスは、SDKの動作中に特定の時点で発生するいくつかのイベントを公開しています。以下は、利用可能なイベントのリストです。:

データパケットイン

このイベントはサーバからデータパケットを受信した際に発火します。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Android iOS

```
public class MyClientEventListener implements ClientEventListener {
    ...
    public void fireDataPacketIn(byte[] dataPacket) {
        Log.i("ClientDataPacketIn", new String(dataPacket));
    }
    ...
}
```

データパケット全体（すべてのフレーミングとエラー検出文字を含む）が `dataPacket` パラメータに含まれます。このパラメータは、より詳細なトラブルシューティングのために検査されたり、追加のレスポンスプロパティを抽出するために使われる可能性があります。

データパケットアウト

このイベントは、各データパケットが送信される直前に発生します。

Android iOS

```
public class MyClientEventListener implements ClientEventListener {
    ...
    public void fireDataPacketOut(byte[] dataPacket) {
        Log.i("ClientDataPacketOut", new String(dataPacket));
    }
    ...
}
```

データパケット全体（すべてのフレーミングとエラー検出文字を含む）が `dataPacket` パラメータに含まれます。このパラメータは、高度なトラブルシューティングのために利用される可能性があります。

ログ

各ログメッセージが出力されるごとに1度発火します。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Android iOS

```
public class MyClientEventListener implements ClientEventListener {
    ...
    public void fireLog(int logLevel, String message, String logType) {
        Log.i("ClientLog", logType + " - " + message);
    }
    ...
}
```

ログの記録は、Log イベントで処理されます。これは、メッセージが作成されたり、応答が解析されたりしたときに、エラーメッセージも含めて、いつでも発生します。

Log イベントが発生すると、 `message` イベントパラメータを介して、該当するメッセージを利用できます。その他のパラメータは `logType` と `logLevel` です。

logType` パラメータは、ログエントリのタイプを示します。指定できる値は以下の通りです。:

- Info
- RequestHeaders
- ResponseHeaders
- RequestBody
- ResponseBody
- ProxyRequest
- ProxyResponse
- FirewallRequest
- FirewallResponse
- CReq
- CRes
- Erro
- EphemeralKey
- DeviceParams
- SW

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

`logType` が `SW` の場合、セキュリティ警告を表します。この値が存在する場合、 `message` パラメータには、より具体的な警告の原因を示すコードが含まれます。以下は、警告の `message` 値とその内容です。:

Android用セキュリティ警告

SWメッセージ値	内容
01	改ざん; 信頼できるストアからインストールされたものではない
02	改ざん; Appストアのチェック時の内部処理エラー
03	改ざん; 誤ったappSignature
04	改ざん; 不審なアプリケーションがインストールされている
05	ルート; 不審なファイルが存在する
06	ルート; 不審なapkが存在する
07	ルート; ルート権限
08	ルート; ルートタグ
09	ルート; フックされている
10	デバッグ; デバッガーが設定されている

iOS用セキュリティ警告

SWメッセージ値	内容
01	改ざん; Ad-HocまたはXCodeからインストールされている
02	改ざん; 辞書情報が無効
03	改ざん; Info.plistがアプリのビルド後に変更されている
04	改ざん; バンドルIDの不一致
05, XX	ジェイルブレイク; デバイス内にジェイルブレイクの疑いがあるファイルが存在する

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

SWメッセージ値	内容
06, XX	ジェイルブレイク; ジェイルブレイクの疑いがあるライブラリが読み込まれた
07, XX	ジェイルブレイク; システム辞書への書き込み権限が有効化された
08	ジェイルブレイク; アプリがフォークする権限を持っている
09, XX	ジェイルブレイク; アプリがジェイルブレイクURLスキームを実行できる
10	デバッグ; p_traceフラグが検出された
11	デバッグ; 内部処理エラー: sysctlが検出できない
12	デバッグ; 内部処理エラー: getpidが検出できない
13	デバッグ; 無効なPID
14, XX	デバッグ; リバースエンジニアリングツールがデバイスから検出された
15, XX	デバッグ; 不審なTCPポートが検出された

LogLevelコンフィギュレーション設定は、Logイベントを通して発生するログの詳細を指定するために使用されます。イベントの `logLevel` パラメータは現在のメッセージが属しているログレベルを示します。指定できる値は以下の通りです。:

- 0 - なし
- 1 - Info
- 2 - Verbose
- 3 - Debug

デフォルトでは、infoレベルのログが利用可能です。クライアント設定 で指定される `LogLevel` 設定を採用するか、より詳細またはより少ないログを出力するようにSDKに指示できます。

このイベントで発生したメッセージは、記録保管やトラブルシューティングのために、すべてファイルに出力することをお勧めします。

SSLサーバ認証

サーバーがクライアントに証明書を提示した後に発火します。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Android iOS

```
public class MyClientEventListener implements ClientEventListener {
    ...
    public void fireSSLServerAuthentication(byte[] certEncoded, String
certSubject, String certIssuer, String status, boolean[] accept) {
        //...
    }
    ...
}
```

このイベントは、TLS 接続を確立するときに発火し、サーバーの証明書に関する情報を提供します。 `accept` パラメータは、証明書がデフォルトで信頼されるかどうかを示します。

`accept` が False の場合、 `status` には検証に失敗した理由が表示されます (それ以外の場合、 `status` には "OK" という文字列が表示されます)。続行することになった場合は、 `accept` パラメータを True に設定すると、証明書を上書きして許可できます。

SSLステータス

安全な接続構築の進行状況を示します。

Android iOS

```
public class MyClientEventListener implements ClientEventListener {
    ...
    public void fireSSLStatus(String message) {
        Log.i("ClientSSLStatus", message);
    }
    ...
}
```

このイベントは、情報提供とロギングのみを目的として発火します。接続の進行状況を追跡するために使用されます。

セキュリティイベントリスナー

`SecurityEventListener` クラスは、SDK の初期化時とランタイムの両方で発生するセキュリティ問題を通知する方法を以下にご紹介します。これは `alarm` イベントを通して行われます。

アラーム

セキュリティ上の問題が検知された際に発火します。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Android iOS

```
public class MySecurityEventListener implements SecurityEventListener {
    ...
    public void alarm(Severity severity, SecurityEvent event) {
        TransactionManager.getInstance().showToast("[Security Alarm] Severity: "
+ severity + ", Event: " + event);
    }
    ...
}
```

セキュリティ上の問題が検出された場合、Alarm イベントが発生します。エラーの深刻度は `severity` イベントパラメータで示されます。設定可能な値は以下の通りです。:

- LOW
- MEDIUM
- HIGH

`event` パラメータはどの問題を検出したかを示します。設定され得る値は以下のとおりです。:

- ROOT
- TAMPERED
- INSTALLED_FROM_UNTRUSTED_STORE
- HOOK
- EMULATOR
- DEBUGGING
- DEBUG_ENABLED

コンポーネントが実行するセキュリティチェックの詳細な情報と、検出された問題の処理方法については、以下のセキュリティチェックのセクションをご参照ください。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

ディレクトリサーバ情報

`DirectoryServerInfo` パラメータでは、各ディレクトリサーバ (DS) の証明書の詳細を定義できます。 `Transaction` を作成する際に、DS_ID を使用してそのトランザクションで使用する証明書を指定します。各 `DirectoryServerInfo` は3つの情報で構成されています。:

- **DS_ID**: トランザクションを特定のDSと照合するためにSDK内部で使用される識別子。RIDとも呼ばれる。
- **DS_CERT**: DSから提供される暗号化証明書。DeviceInfoを暗号化し、Challengeセキュリティに使用するキーを生成するために使用される。
- **DS_CA_CERT**: DS_CERTを発行するために使用されたルートCA証明書。DSからの応答（AResパケット）の署名を検証するために使用され、中間証明書も含めることができる。

`DS_CERT` および `DS_CA_CERT` 証明書はPEMでエンコードされています。 `DSInfo` クラスを実装する際、PEMでエンコードされた証明書の内容はスタティックな変数の値として使用することができます。GPayments TestLabs証明書を用いた例は以下のとおりです。:

```
public class DSInfo {

    public static final String DS_ID = "DS_RSA_TESTLAB_PROD";

    public static final String DS_CERT = "-----BEGIN CERTIFICATE-----\n" +

"MIIDyzCCArOgAwIBAgIUWcqKLMtnejuVbszZM00DLPZRqO4wDQYJKoZIhvcNAQEL\n" +

"BQAwdzELMAkGA1UEBhMCQVUxGDAWBgNVBAgMD05ldyBTb3V0aCBXYWxlczEPMA0G\n" +

"A1UEBwwGU3lkbmV5MRIwEAYDVQQKDAlHUGF5bWVudHMxEjAQBgNVBAsMCUFjdGl2\n" +

"ZVNESzEVMBMGA1UEAwwMQWN0aXZlU0RLLkNBMB4XDTIxMDcxNTA4MDMwOVoXDTIz\n" +

"MTAxODA4MDMwOVowcTELMAkGA1UEBhMCQVUxGDAWBgNVBAgMD05ldyBTb3V0aCBX\n" +

"YWxlczEPMA0GA1UEBwwGU3lkbmV5MRIwEAYDVQQKDAlBY3RpdmVTREsxEjAQBgNV\n" +

"BAsMCUFjdGl2ZVNESzEPMA0GA1UEAwwGU0RLIHYxMIIBIjANBgkqhkiG9w0BAQEF\n" +

"AAOCAQ8AMIIBCgKCAQEAzxcnpOaV9lGE4XWiafXtnYynlEMqrN65cqlovSd73nL4\n" +
                "WaAgl36lBVHb2/
vYEjIQLcf8VNme8bb9EvQmNulyva3ifBbQKRZT96XRVAzpj5qb\n" +

"47rpfskQ4gemR3XZZbqH2zMBPPTdXEKwkYbXPVE9riHkciLcbDO686zjsLLpTcmN\n" +
                "mqnFcJbSDggZfntKsTDro/
deenKKJd8q9yDlqO8KaZkXD1M2AWBF+aOEW5QINUg4\n" +
```

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

```
                "C4n89Q30Wva2c1q4S9DBczBHcSQntl9xIcOKrp3//
OJBP0Bl00S6CgxzpSnel1+J\n" +
                "en6cuDjbEm2WsOJrt5Nsjc+8LlYwIsuII3qd3Nuf/
QIDAQABo1UwUzAfBgNVHSME\n" +
                "GDAWgBTXgOf2gz2c4gvLTz6WbTL4/
QgKajAJBgNVHRMEAjAAMAsGA1UdDwQEAwIE\n" +

"8DAYBgNVHREEETAPgg1hY3RpdmVzZGsuY29tMA0GCSqGSIb3DQEBCwUAA4IBAQCx\n" +
                "8q1JOdZub+Obv/GABj/
eh4+3fSBdIuDQrCsNRC+uGWX7rfzWRQ0Px9qcK4cGbdVM\n" +

"SVek6zpF7m8LrX7AQBSXk6CgFWrjJqkwo8OzXddE4FTXdbBYKAIb5slhIXMVzPhu\n" +
                "3DnJ/QnXc2QfwrAaXSe/
vJ0eUNMm37dbJlGsLrJHFBWkZ9LphYg4Ji2TKYLIx5ek\n" +
                "3vxfj3iIFgCc/
RdAHKX38sG2+cASCKxonkK8SKgcqo5rKt+b4gny9yqSMEqfvT6j\n" +
                "4CyitN6ggL3nPnecc2MFpCSIYA+ZMf0UeLREOcVSg4o8rDB/
T8ZmGbDaz43E7AiB\n" +
                "sMcXGlVMEVh1isB/3SrW\n" +
                "-----END CERTIFICATE-----\n";

    public static final String DS_CA_CERT = "-----BEGIN CERTIFICATE-----\n" +
                "MIIDzzCCAregAwIBAgIUcfdJYkaK8cGVC8JX2RfP/
gWKHc4wDQYJKoZIhvcNAQEL\n" +

"BQAwdzELMAkGA1UEBhMCQVUxGDAWBgNVBAgMD05ldyBTb3V0aCBXYWxlczEPMA0G\n" +

"A1UEBwwGU3lkbmV5MRIwEAYDVQQKDAlHUGF5bWVudHMxEjAQBgNVBAsMCUFjdGl2\n" +

"ZVNESzEVMBMGA1UEAwwMQWN0aXZlU0RLLkNBMB4XDTIxMDcxNTA3NTc0NFoXDTI2\n" +

"MDcxNDA3NTc0NFowdzELMAkGA1UEBhMCQVUxGDAWBgNVBAgMD05ldyBTb3V0aCBX\n" +

"YWxlczEPMA0GA1UEBwwGU3lkbmV5MRIwEAYDVQQKDAlHUGF5bWVudHMxEjAQBgNV\n" +

"BAsMCUFjdGl2ZVNESzEVMBMGA1UEAwwMQWN0aXZlU0RLLkNBMIIBIjANBgkqhkiG\n" +
                "9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxG68SctQctVCM1L/
CIoFOr6W2TYbrx5Wj12Q\n" +
                "PTy/MWF0XZK8780iwI7xAjHeeyEMFPv/BOL6dzVtQYA/
3pNCdi5uhwqBhfvxeN0V\n" +
                "jo7A8zxFLH6/JkF4PUyffn10d3rvqKddOW/0/
gv1A32earnOVfhJympe1jfqgQbt\n" +

"Za8ryOxLVujP3OSyDL9hobTevd1ffYpnFyCHC4mqJVAELEoLEFWEp4ixlRh9yvUh\n" +

"j2icv4vHO6gE3MDWGjNoMwix3hsv6p1gQbzsUAWspg3UkjIALO3GnCY4U6Ml11jN\n" +

"cnp9cIP33madxY93vb0oKmJHCHsdBdVGLgcXFD2+gVAroN05pwIDAQABo1MwUTAd\n" +

"BgNVHQ4EFgQU14Dn9oM9nOILy08+lm0y+P0ICmowHwYDVR0jBBgwFoAU14Dn9oM9\n" +
                "nOILy08+lm0y+P0ICmowDwYDVR0TAQH/BAUwAwEB/
```

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

```
zANBgkqhkiG9w0BAQsFAAOC\n" +

"AQEAXtGL3CI9K2QugLv0HAPL2Jj0k9JMiV8RpQbMUVPmLQm9Tk48o3RKxZpCGizN\n" +
                "VKt3Lk5ryNAWkzypYfparZow8s/fRA0dXmZtq/
1Fjnj8ZJRmnYmKpIssdi+0NPr1\n" +
                "7V+2Y8g7wpinQ9w0tyMc7CVcqjlD3p+
+djiocjQ70HA6FdepxHF9Uj0Kzu71RecF\n" +
                "a7bqX9hdoS4dg9AkZzOQ6JF7NBcW225vhHJkQUekojW/
vloJPjnR7wDTaLrQVqq3\n" +
                "gIDwo2TgF/W1rIG9bf0hNx5V3fuOk1+Pv/8SE1tMnSj/KwHEeRKIWYrVn/
+sHKGQ\n" +
                "95DBV5KDrrNt5lr8CK5XhisHoQ==\n" +
                "-----END CERTIFICATE-----\n";

}
```

設定済ディレクトリサーバのDS_IDの利用

上記の `DirectoryServerInfo` パラメータ追加以外にも、SDKにはいくつかの国際ブランドの商用ディレクトリサーバの `DirectoryServerInfo` があらかじめ設定されています。サポートされている国際ブランドは以下の通りです。:

Card scheme	DS_ID
Visa	A000000003
Mastercard	A000000004
Amex	A000000025
Discover	A000000152
JCB	A000000065
UnionPay	A000000333

上記の設定済みのDS_DSを使用するには、トランザクションの作成のセクションでDS_IDを指定してください。この設定済みのDS_IDを使用する場合、 `DirectoryServerInfo` パラメータに証明書を設定する必要ありません。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Important

これらの事前設定された証明書を最新の状態に保つために注意しておりますが、変更が発生した場合に備えて、お客様側でも国際ブランドの更新を監視し、認識し続けることを推奨します。証明書の更新が行われた場合、SDKの新しいバージョンが利用可能になります。このアップデートは弊社のサイトから入手できますが、お客様はSDKのどのバージョンでも（DirectoryServerInfoListパラメータを使用して）手動で詳細を更新または追加できるため、絶対に必要なわけではありません。

クライアント設定

SDKの動作を変更する追加のオプション設定が可能です。これらはSETTINGとVALUEのペアで設定され、 `ConfigParameters` を構築する際に `clientConfig` として設定されます。

ログレベル

コンポーネント内でのログレベルはこの設定で指定されます。設定され得る値は以下のとおりです。:

0 (None)	すべてのイベントに対してログを出力しない
1 (Info - default)	Infoレベルのイベントのみログを出力する
2 (Verbose)	詳細なデータをログとして出力する
3 (Debug)	デバッグ用のデータをログとして出力する

デフォルトでは1 (Info)が設定されています。

機密データのマスキング

この設定は、 `Log` イベントにおいて機密データをマスクするかどうかを制御します。 `True` (デフォルト) に設定すると、カード番号の値は `*********` という値に置き換えられます。

備考: この設定に関わらず `DataPacketOut` ではマスクされていない生データが含まれます。この設定は `Log` イベントのみに対して有効です。

デフォルト値は `True` です。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

セキュリティチェック

SDKは初期化中にセキュリティチェックを行い、デバイス情報を収集します。3DS SDKの初期化時に発生する警告は、 `getWarnings` メソッドを使用して取得することができ、 `Warning` オブジェクトのリストを返します。これらの警告は、アプリがチェックアウト処理を続行するかどうかを決定するために確認できます。:

Android iOS

```
/* check warnings */
List<Warning> warnings = ThreeDS2Service.INSTANCE.getWarnings();
if (warnings.size() > 0) {
    // process warnings
    // abort the checkout if necessary
}
```

`Warning` オブジェクトには下記のフィールドが含まれています。:

名前	型	説明
id	String	警告の識別子
message	String	警告メッセージ
severity	Severity (enum)	警告レベル(LOW, MEDIUM, HIGH)

下記の警告はEMVCo仕様で定義されています。:

セキュリティ警告ID	説明	重大度
SW01	デバイスにジェイルブレイクが確認された	HIGH
SW02	SDKの整合性が改ざんされている	HIGH
SW03	アプリの実行にエミュレーターが使用されている	HIGH
SW04	アプリにデバッガが添付されている	MEDIUM
SW05	OSまたはOSのバージョンがサポートされていない	HIGH

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

初期化時と実行時の両方でセキュリティ問題を通知するには、セキュリティイベントリスナーも使用することを推奨します。

初期化処理中のセキュリティチェック

下記のセキュリティチェックはSDKの初期化中に実施されます。:

- ルート検出：デバイスがルート化されているかどうかの検出
- SDK改ざん検知：アプリ署名者の指紋認証、フック検知、悪意のあるアプリの検知
- アプリが信頼できるアプリストアからインストールされていることの確認
- エミュレーター検知
- デバッガ適用の検知
- OSバージョンの検知

これらの初期化チェックの結果、上記の `getWarnings` で警告を確認できます。このメッセージには、発生した問題の説明が含まれます。

ランタイム上で実施されるセキュリティチェック

下記のセキュリティチェックはランタイム上で実施されます。:

Android用セキュリティチェック

- エミュレーターの検知 (重大度：高): `Transaction.getAuthenticationRequestParameters()` 実施中にチェック
- デバッガーの検知 (重大度：中): `Transaction.doChallenge()` 実施中にチェック
- ルートの検知 (重大度：高): `Transaction.doChallenge()` 実施中にチェック
- フックの検知 (重大度：高): `ThreeDS2Service.createTransaction()` 実施中またはチャレンジプロセス内でSubmit/Resend/Cancelのいずれかがクリックされた際にチェック
- デバッグ許可の検知 (重大度：低): `Transaction.doChallenge()` 実施中にチェック

iOS用セキュリティチェック

- エミュレーターの検知 (重大度：高): `Transaction.doChallenge()` の実施中にチェック
- デバッガーの検知 (重大度：中): `Transaction.doChallenge()` の実施中にチェック

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

- メソッド処理入れ替え(swizzling)フックの検知 (重大度：高): `Transaction.getAuthenticationRequestParameters()` または `Transaction.doChallenge()` の実施中にチェック
- フィッシング用フックの検知 (重大度：高): `Transaction.getAuthenticationRequestParameters()` または `Transaction.doChallenge()` の実施中にチェック
- アンチデバッグ (予防): `Transaction.getAuthenticationRequestParameters()` の実施中にチェック

これらの実行時チェックの結果、Security Event Listener `alarm` イベントが発生し、深刻度が高い問題ではアクティビティが終了します。SDKの開発バージョン（ファイル名に `_deploy` を含まない）では、これらのチェックは実行されません。

取引の作成

`createTransaction` メソッドから取引を開始できます。3-D Secureのプロセス全体を通じて、このトランザクションの参照を保持する必要があります。このメソッドは、SDKの初期化時に追加された `DirectoryServerInfo` コンフィグパラメータを指す、ディレクトリサーバーIDを受け取ります。2番目のパラメータは、プロトコルのバージョンを文字列で指定します。指定可能なバージョン文字列は `2.1.0` と `2.2.0` です。

Android iOS

```
sdkTransaction = ThreeDS2Service.INSTANCE.createTransaction(DSInfo.DS_ID,
"2.2.0");
```

DS_IDについては, ディレクトリサーバ情報 の項をご参照ください。

その後、取引が完了したら、Transaction オブジェクトの `close` メソッドで取引を欠かさずクローズしてください。

進捗状況の表示

3-D Secure プロセスのどの時点を処理しているか、エンドユーザーに対して進捗ダイアログを表示する必要があります。チャレンジを実行中は、3DS SDK は自動的に進捗ダイアログを表示します。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

アプリケーションが3DSサーバーと通信し、認証リクエストの結果を待っている間、進捗ダイアログもユーザーに表示する必要があります。3DS SDK が使用するダイアログは、必要に応じてアプリケーションで使用することができ、`Transaction` オブジェクトから `getProgressView` メソッドを介して以下のように取得することができます。:

Android iOS

```
// The progress dialog is an instance of android.app.ProgressDialog
ProgressDialog sdkProgressDialog;

// Retrieve the progress dialog from the transaction object instance
sdkProgressDialog = sdkTransaction.getProgressView(mActivity);
sdkProgressDialog.show();

// Later, to hide/dismiss:
sdkProgressDialog.dismiss();
```

3DS SDKは、チャレンジプロセスを開始する際に自動的にダイアログを隠すので、アプリケーションコード内で明示的にダイアログを解除する必要はありません。

認証処理

認証プロセスは `Transaction` オブジェクトの `getAuthenticationRequestParameters` メソッドを呼び出すことで始まります。このメソッドが呼び出されると、3DS SDKは初期化中に収集したデバイス情報を暗号化し、3DSサーバーが必要とする他のSDK情報とともに、返された `AuthenticationRequestParameters` オブジェクトで利用できるようにします。この `AuthenticationRequestParameters` オブジェクトは以下のフィールドで構成されています。:

名前	説明
SDKTransactionId	SDK側でランダム生成される取引のID
DeviceData	初期化中にSDKに収集されたデバイスデータと暗号化されたDSキー
SDKEphemeralPublicKey	SDK が生成する一時キーペアの公開鍵コンポーネント。
SDKAppId	アプリケーションのデバイスを一意に特定する3DS SDKが生成したUUIDの値
SDKReferenceNumber	EMVCoから認証を受けた後に弊社SDKに割り当てられる番号

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

名前	説明
MessageVersion	`createTransaction` から引継いだ値かSDKのデフォルト値(2.1.0)
AuthRequest	3DSサーバに送信されるパッケージ化された認証リクエストデータ

AuthRequest`プロパティは、他のプロパティをパッケージ化したものです。アプリは、3DS SDK以外との通信チャネルを介して、これを3DSサーバに送信することになります。ActiveServerとの認証メッセージ交換の実装は、3DSリクエスターの章の記載をご参照ください。 他社3DSサーバとの統合については、先方の実装ガイドをご確認ください。

レスポンスの待機

3DS サーバは、ディレクトリサーバからの認証応答を読み取り、クライアント（この場合はアプリ）に対してチャレンジが必要であるかどうかを示す役割を担います。ActiveServerの場合、認証メッセージの交換が完了した後、 `TransStatus` プロパティを確認します。 値が `C` または `D` の場合チャレンジが必要となります。

チャレンジが不要の場合(フリクションレス・フロー)、または何らかの理由で認証処理が実施されなかった場合、アクションは不要です。認証の終了 の章に進んで取引の終了方法についてご確認ください。

チャレンジが必要となる場合、3DSサーバーはアプリケーションがチャレンジプロセスを開始するために必要な値を返却します。 チャレンジプロセスの実施の章に進んでチャレンジの実施方法をご確認ください。

チャレンジプロセスの実施

チャレンジが必要な場合、チャレンジはチャレンジパラメーターを作成し（3DSサーバーからの認証応答を使用）、チャレンジの開始のセクションの手順に従うことによって実行されます。チャレンジの実行中、SDKはACSとの通信を行い、また必要に応じてユーザーに必要な入力を促します。チャレンジプロセスが完了すると、関連する チャレンジステータスレシーバー コールバックが起動します。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

チャレンジステータスレシーバー]

`ChallengeStatusReceiver` はSDKのチャレンジプロセス結果を取得するインターフェースです。これはアプリケーションで実装され、チャレンジの開始時に `doChallenge` メソッドに引き渡される必要があります。 チャレンジプロセスが完了した時、その成否に関わらず、以下のメソッドのうちどれか1つが呼び出されます。:

名前	説明
completed	チャレンジ処理（取引）が完了したときに呼び出される。取引が完了すると、そのステータスが得られる。
cancelled	カード会員がチャレンジ画面で取引をキャンセルするオプションを選択したときに呼び出される。
timedout	`doChallenge` 呼び出し時に指定されたタイムアウト間隔に達した時、またはそれを超えた時に呼び出される。
protocolError	3DS SDKがACSからEMV 3-D Secureプロトコル定義のエラーメッセージを受信したときに呼び出される。
runtimeError	3DS SDKがチャレンジ処理中にエラーが発生したときに呼び出される。これらのエラーは、`protocolError` メソッドによってカバーされるものを除くすべてのエラーを含む。

各メソッドに対応する実装例は以下のとおりです。:

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Android iOS

```
@Override
public void completed(CompletionEvent completionEvent) {
    showToast("Challenge completed with transactionStatus " +
completionEvent.getTransactionStatus());
    closeTransaction();
}

@Override
public void cancelled() {
    showToast("Challenge cancelled.");
    closeTransaction();
}

@Override
public void timedout() {
    showToast("Challenge timed out.");
    closeTransaction();
}

@Override
public void protocolError(ProtocolErrorEvent protocolErrorEvent) {
    showToast("Challenge protocolError: " +
        protocolErrorEvent.getErrorMessage().getErrorDescription() + "\t" +
        protocolErrorEvent.getErrorMessage().getErrorDetails());
    closeTransaction();
}

@Override
public void runtimeError(RuntimeErrorEvent runtimeErrorEvent) {
    showToast("Challenge runtimeError: " + runtimeErrorEvent.getErrorMessage());
    closeTransaction();
}

// Helper method
public void closeTransaction() {
    if (sdkTransaction != null) {
        sdkTransaction.close();
        sdkTransaction = null;
    }
    sdkProgressDialog = null;
}
```

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

チャレンジパラメータ

`ChallengeParameters` クラスは、チャレンジ処理を行うために必要なパラメータを保持します。これは チャレンジの開始 際に `doChallenge` メソッドに渡されます。ゲッターとセッターでアクセスできる、利用可能なプロパティは以下の通りです。:

名前	型	説明
3DSServerTransactionID	String	3DSサーバートランザクションID
AcsTransactionID	String	ACSのトランザクションID
AcsRefNumber	String	ACSのリファレンス番号
AcsSignedContent	String	ACS 署名済みコンテンツ。このデータには、ACSのURL、ACSの一時公開鍵、SDKの一時公開鍵が含まれる。
ThreeDSRequestorAppURL	String	3DSリクエスターアプリのURL
ThreeDSServerAuthResponse	String	3DSサーバーから返却された認証情報

これらのプロパティは、3DS サーバーから返却される内容に基づいて個別に設定することができます。3DS Serverコンポーネントを使用している場合、 `ThreeDSServerAuthResponse` に生成された `ClientAuthResponse` の値を設定すると、他のプロパティはこの値に基づいて自動的に設定されます。

Android iOS

```
ChallengeParameters challengeParameters = new ChallengeParameters();
challengeParameters.setThreeDSServerAuthResponse(authResponse);
```

チャレンジの開始

3DS サーバーからの応答がチャレンジが必要であることを示す場合、 `doChallenge` メソッドを使用してチャレンジプロセスを開始します。このメソッドが呼び出されると、UIのコントロールは3DS SDKに引き渡されます。チャレンジはSDKによって実行され、終了すると適切なチャレンジステータスレシーバ イベントが発生します（完了、キャンセルなど）。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Android iOS

```
sdkTransaction.doChallenge(currentActivity, challengeParameters, this, 5);
```

`doChallenge` メソッドは下記のパラメータが必要になります:

Android用のチャレンジパラメータ

名前	型	説明
currentActivity	android.app.Activity	チャレンジプロセス時にSDKが使うアクティビティ
challengeParameters	ChallengeParameters	3DS SDKが取引中のチャレンジプロセスを実施するにあたって必要なACSの詳細情報。詳細はチャレンジパラメータの章をご確認ください。
challengeStatusReceiver	ChallengeStatusReceiver	チャレンジstatusを3DSリクエスターアプリケーションに通知するコールバックオブジェクト。詳細はチャレンジステータスレシーバの章をご確認ください。
timeout	Integer	チャレンジプロセス完了までのタイムアウトインターバル（分単位）。最小タイムアウトインターバルは5分。

iOS用のチャレンジパラメータ

名前	型	説明
navigationController	UIViewController	チャレンジプロセス用のSDKのビューコントローラ。
challengeParameters	ChallengeParameters	3DS SDKが取引中のチャレンジプロセスを実施するにあたって必要なACSの詳細情報。詳細はチャレンジパラメータの章をご確認ください。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

名前	型	説明
challengeStatusReceiver	チャレンジstatusを3DSリクエスターアプリケーションに通知するコールバックオブジェクト。詳細はチャレンジステータスレシーバの章をご確認ください。	
timeout	Integer	チャレンジプロセス完了までのタイムアウトインターバル（分単位）。最小タイムアウトインターバルは5分。

認証処理の終了

`completed` コールバックが起動すると、 `CompletionEvent` パラメータに認証結果に関する詳細な情報が格納されます。これには2つのプロパティがあります。 `sdkTransactionID` と `transactionStatus` です。 `transactionStatus` には、最後のCResパケットのtransStatusが含まれ、以下の値のいずれかになります。

- **Y:** 正常終了 / 認証成功Authentication successful
- **A:** 正常終了 / Attemptプロセスを実施
- **N:** 拒否 / 認証不備(拒否)
- **R:** 拒否 / 認証の拒否
- **U:** 技術的な問題による失敗 / 認証処理は未実施

リソースの管理

`cleanup` メソッドは3DS SDKによって使用されるリソースを解放します。このメソッドは、1つの3DS Requestor Appセッションの間に1度だけ呼び出されます。

Android iOS

```
ThreeDS2Service.INSTANCE.cleanup(applicationContext);
```

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

認証結果の取得

フリクションレスフローの場合、認証プロセスが終了すると、モバイルアプリに認証結果が返されます。チャレンジフローでは、モバイルアプリに「transStatus=C」が返され、Active SDKとACSの間でチャレンジ処理が実行されます。チャレンジが完了すると、ActiveServerの場合、モバイルアプリは `getResult` エンドポイントを呼び出して認証結果を取得する必要があります。 `getResult` エンドポイントの詳細については、リクエスターの章をご確認ください。

アプリケーションのパーミッションについて

SDK自体は、インターネットアクセス以外の特定のパーミッションを必要とせず、アプリケーションが要求したパーミッションのみを使用します。アプリはSDKが初期化される前にパーミッションを要求する必要があります。これは仕様上必要なことであり、パーミッションが付与されていない場合、一部のデバイスパラメータが使用できなくなります。

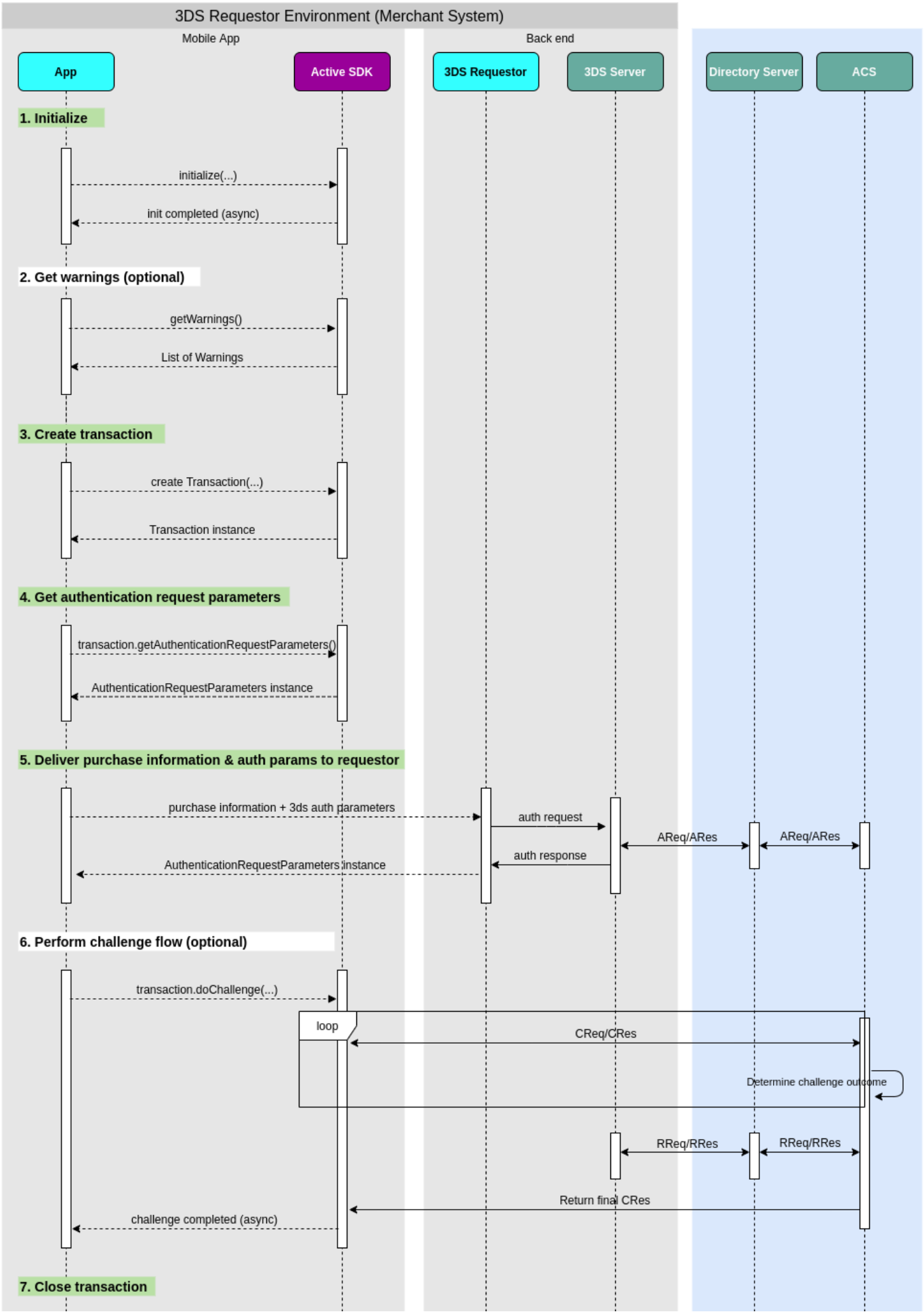
各デバイスのパラメータに必要な権限の一覧は、EMVCoから入手可能なEMV 3-D Secure SDK Device Informationドキュメントをご参照ください。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

認証シーケンス

以下のシーケンス図は3DS2のAPIのコンセプトを説明し、各3DS2コンポーネント間の相互関係を視覚化します。

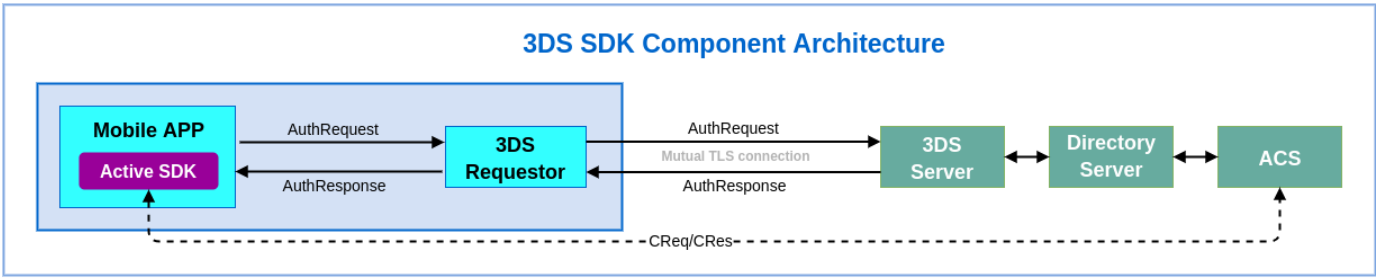
Copyright ©2019 GPayments Pty Ltd. All rights reserved.



Copyright ©2019 GPayments Pty Ltd. All rights reserved.

3DSリクエスター

ActiveSDKを実装するには、加盟店サイトはバックエンドに**3DS Requestor**を実装する必要があります。以下の図はActiveSDK、3DS Requestor、およびその他のコンポーネントとの関係性を表します。EMVCo 3D Secure 2仕様書に基づき、加盟店アプリと3DS Server間の通信は相互認証される必要があります。そのため、バックエンドには3DS Serverと相互TLSコネクションを確立するための3DS Requestorを実装する必要があります。



このドキュメントでは、ActiveSDKを弊社の3DS Requestorサンプル・コードと統合するプロセスについて説明します。このデモでは、3DS Requestorが接続する3DS Serverは**ActiveServer**になります。ただし、3DS Requestorのサンプル・コードを修正すればどの3DS Serverにも接続することができます。

以下はこのガイドを使用するための前提条件です：

- このいずれに関する知識： Java、PHP、C#、もしくはGo

サンプル・コードをチェックアウトして実行する

3DS Requestorサンプル・コードはダウンロード・ページよりダウンロードできます。クライアント証明書の取得および3DSリクエスターデモの設定の手順を参照して、3DS Requestorを実行してから次のステップへ進んでください。

> **注釈**
>
> 3DS Requestorサンプル・コードを実行するには、アクティブ化されて稼働中のActiveServerインハウスのインスタンス（もしくはActiveServer SaaSのテナント）が必要になります。詳しくは**GPayments社**へご連絡ください。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

3DS Requestorと3DS Server間の実装

3DS Requestorはモバイルアプリから `AuthRequest` を受信し、そのリクエストを3DS Serverへ送信します。また、3DS Requestorは3DS Serverから `AuthResponse` を受信し、その結果をモバイルアプリへ転送します。

3DS Requestorのサンプル・コードはバックエンドの実装方法を以下のサーバー側言語で提示します：

- **Java**: Java版はSpringbootのフレームワークを基に実現されています。Springbootに関する詳細については、 `https://spring.io/projects/spring-boot` を参照してください。
- **C#**: C#版はASP.netを基に実現されています。
- **PHP**: PHP版はPHP 7.2とcURL (Client URL Library)を基に実現されています。
- **Go**: Go版はGo 1.12とGo Module対応を基に実現されています。全てのディペンデンシーは `go.mod` ファイルにリストアップされています。

3DS Serverと認証処理を開始する前に、3DS Requestorは3DS Serverと相互TLS認証コネクションを確立する必要があります。クライアント証明書の取得および3DSリクエスターデモの設定の手順を確認してください。

> **HTTPクライアントのTLS設定**
>
> - Java: TLS設定とクライアント証明書のロードは `RestClientConfig.java` クラスから参照できます。
> - C#: TLS設定とクライアント証明書のロードは `RestClientHelper.cs` クラスから参照できます。
> - PHP: TLS設定とクライアント証明書のロードは `RestClientConfig.php` クラスから参照できます。
> - Go: TLS設定とクライアント証明書のロードは `https.go` クラスから参照できます。

認証を実行する

認証を実行するには、3DS Requestorは以下のタスクのために `/v2/auth/app` エンドポイントを実装する必要があります：

- モバイルアプリから `AuthRequest` を受信する。
- そのリクエストを3DS Serverに転送する。
- 3DS Serverから `AuthResponse` を受信する。
- レスポンスのデータをモバイルアプリへ返却する。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Java C# PHP Go

```
//AuthControllerV2.java
@PostMapping("/v2/auth/app")
@ResponseBody
public Message app(@RequestParam(value = "trans-type", required = false) String
transType,
    @RequestBody Message request) {
  return authServiceV2.app(transType, request);
}

//AuthServiceV2.java
public Message app(String transType, Message request) {

  //generate requestor trans ID
  String transId = UUID.randomUUID().toString();
  request.put(THREE_DS_REQUESTOR_TRANS_ID, transId);

  String appAuthUrl = config.getAsAuthUrl() + "/api/v2/auth/app";

  //Add parameter trans-type=prod in the appAuthUrl to use prod DS, otherwise
use TestLabs DS
  //For more details, refer to: https://docs.activeserver.cloud
  if ("prod".equals(transType)) {
    appAuthUrl = appAuthUrl + "?trans-type=prod";
  }

  logger.info("appAuthRequest on url: {}, body: \n{}", appAuthUrl, request);

  Message response =
      sendRequest(appAuthUrl, request, HttpMethod.POST);
  logger.info("appAuthResponse: \n{}", response);
  return response;
}
```

注目

このサンプル・コードは、 `AuthRequest` を**ActivesServer**へ送信するためのものです。ActiveServerの `AuthRequest` のデータ構造を確認するには、APIドキュメントを参照してください。他社の3DS Serverへ `AuthRequest` を送信する際は該当製品のAPIドキュメントを参照してください。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

ActiveServerを使用するお客様へ

デフォルトでは、上記URLはテスト目的で `AuthRequest` をGPayments TestLabsへ送信します。本番環境に移行する際にAPIリクエストを各国際ブランドのDirectory　Serverへ送信するには、 `trans-type` クエリー・パラメーターをそのAPI URLに追加する必要があります。使用方法については、APIの説明を参照してください。

Master Auth API クライアント証明書を使用して、加盟店の代わりに**Business Admin**ユーザーを認証する場合、バックエンドはHTTPリクエストに `AS-Merchant-Token` フィールドを含んだHTTPヘッダーを追加する必要があります。なお、これを加盟店プロフィールで設定されたmerchantTokenへ設定する必要があります。実装方法の詳細については、こちらを参照してください。

認証結果を取得する

フリクションレス・フローの場合、認証結果は**認証実行**処理が終了する際にモバイルアプリへ返却されます。

チャレンジ・フローの場合、**認証実行**処理が終了する際はモバイルアプリへ `transStatus=C` が返却され、ActiveSDKとACSの間にチャレンジ処理が実行されます。チャレンジ処理が終了する際、モバイルアプリは `/v2/auth/result` エンドポイントを呼び出し、認証結果を取得します。3DS Requestorはその結果を取得し、モバイルアプリへ返却するためには、ActiveServerへリクエストを送信する必要があります。

Java　C#　PHP　Go

```java
//AuthControllerV2.java
@ResponseBody
@GetMapping("/v2/auth/brw/result")
public Message resultBRW(@RequestParam("txid") String serverTransId) {
  return authServiceV2.getBRWResult(serverTransId);
}

//AuthServiceV2.java
public Message getBRWResult(String serverTransId) {
  //ActiveServer url for Retrieve Results
  String resultUrl = config.getAsAuthUrl() +
      "/api/v2/auth/brw/result?threeDSServerTransID=" +
      serverTransId;

  //Get authentication result from ActiveServer
  Message response = sendRequest(resultUrl, null, HttpMethod.GET);
  logger.info("authResponse: \n{}", response);

  return response;
}
```

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

モバイルアプリと3DS Requestor間の実装

モバイルアプリは3DS Requestorへ `AuthRequest` を送信する必要があります。以下はAndroidアプリの場合にそのリクエストを送信する際のサンプル・コードです。iOSの場合はこれと類似します。

> **警告**
>
> モバイルアプリと3DS Requestorの間の通信を実装するのは加盟店の役割です。**安全な通信**をするために、適切な認証処理の実施は必要です。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Android

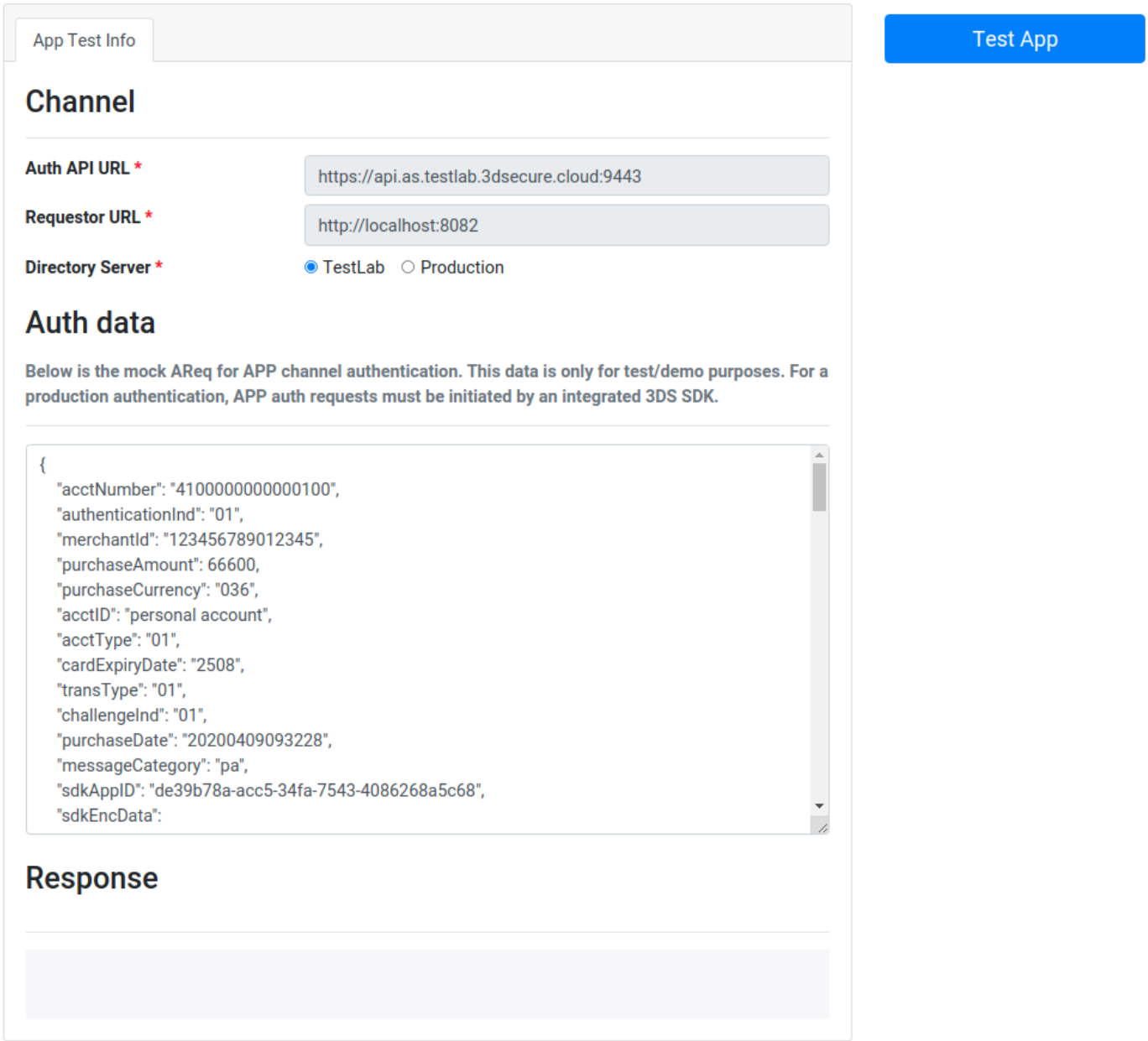
```
private Button purchaseBtn;
private OkHttpClient client = new OkHttpClient();
...
@Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  ...
  purchaseBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        ...
        AuthenticationRequestParameters authParams = transaction
            .getAuthenticationRequestParameters();

        PurchaseRequest appAuthReq = new PurchaseRequest(authParams,
            cardNumberEdit.getText().toString());

        final Request request = new Request.Builder()
            .url(REQUESTOR_URL + "/v2/auth/app")
            .post(RequestBody.create(JSON, appAuthReq.toJson()))
            .build();

        client.newCall(request).enqueue(new Callback() {
          ...
          @Override
          public void onResponse(Call call, Response response) throws
IOException {
            progress.dismiss();

            final String responseBody = response.body().string();
            Log.d(TAG, "Success purchase request: " + responseBody);
            ...
          }
        }
    }
}
```

3DS Requestorのトラブルシューティング

3DS Requestorでは、トラブルシューティングのために、フロントエンドには**アプリ・テスト**ページが実装されています。 `Test pages -> App` を選択すれば、 **アプリ・テスト**ページへ移動することができます。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

そのページには、テスト/デモのみの目的でダミー `AuthRequest` データが事前に入力されています。 `Test App` ボタンを押すと、ダミー・データは3DS Requestorのバックエンドの `/v2/auth/app` エンドポイントに送信されます。 `AuthResponse` は `Response` の部分に表示されます。ちなみに、本番で認証を実施する場合は統合された3DS SDKから `AuthRequest` を起動する必要があります。



Copyright ©2019 GPayments Pty Ltd. All rights reserved.

用語集

この章では3Dセキュア2に関連する用語を列挙いたします。

Term	Definition
3DSクライアント	EMV 3-Dセキュア・プロトコルを開始するための消費者と3DSリクエスターのやり取りを容易にする、ブラウザベースまたはモバイルアプリのオンラインショッピングサイトなどの消費者向けコンポーネント。
3DSインテグレーター	3DSリクエスター環境を容易化および統合し、オプションで加盟店とアクワイアラー間の統合を容易にするEMV 3-Dセキュア参加者。
3DSリクエスター	AReqメッセージとも呼ばれるEMV 3-Dセキュア認証リクエストのイニシエーター。たとえば、購入フロー内で認証をリクエストする加盟店やデジタル・ウォレットなどです。
3DSリクエスターApp	3DS SDKを使用して、3-Dセキュア取引を処理できるコンシューマー・デバイス上のアプリ。3DSリクエスターアプリは、3DS SDKとの統合によって有効化されます。
3DSリクエスター Environment	これは、通常、3DSインテグレーターによって容易化される加盟店/アクワイアラードメインの、3DSリクエスター制御コンポーネントを指します。これらのコンポーネントには、3DSリクエスターアプリ、3DS SDK、3DSサーバーが含まれます。3DSリクエスター環境の実装は、3DSインテグレーターで定義されているとおりに変化します。
3DS SDK	3-Dセキュア・ソフトウェア開発キット。3DSリクエスターアプリに組み込まれているコンポーネント。3DS SDKは、3DSサーバーの代わりに3-Dセキュアに関連する機能を実行します。
3DSサーバー	オンライン取引を処理し、3DSリクエスターおよびディレクトリ・サーバー間の通信を容易にする3DSインテグレーターのサーバーまたはシステムを指します。
3-Dセキュア (3DS)	3ドメイン・セキュア。バージョン2以降では、決済、非決済、およびアカウント確認カード取引の安全な処理が可能なeコマース認証プロトコル。
Access Control Server (ACS)	イシュアードメインで機能するコンポーネント。カード番号およびデバイスタイプで認証が利用可能かどうかを確認し、特定のカード会員を認証します。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Term	Definition
Authentication	3-Dセキュアのコンテキストでは、eコマース取引を行った個人に決済カードを使用する資格があることを確認するプロセスです。
Authentication Request(AReq) Message	認証プロセスを開始するために、DSを介して3DSサーバーからACSに送信されるEMV 3-Dセキュア・メッセージ。
Authentication Response (ARes) Message	認証リクエスト・メッセージに対するレスポンスで、DSを介してACSから返されるEMV 3-Dセキュア・メッセージ。
Authentication Value(AV)	オーソリゼーション処理中にオーソリゼーションシステムが認証結果の完全性を検証するための方法を提供する、ACSによって生成された暗号値。AVアルゴリズムは各決済システムによって定義されます。
Authorisation	イシュアーまたはイシュアーの代理のプロセッサーが決済の取引を承認するプロセス。
Authorisation System	決済システムがイシュアーおよびアクワイアラーに対してオンライン金融処理、オーソリゼーション、清算、決算サービスを提供するシステムおよびサービス。
Bank Identification Number (BIN)	発行金融機関を一意に識別する決済カードアカウント番号の最初の6桁。ISO 7812ではイシュアー識別番号（IIN）とも呼ばれます。
Base64	RFC 2045で定義される、認証値データ要素に適用されるエンコーディング。
Base64url	RFC 7515で定義される、3DSメソッドデータ、デバイス情報、CReq/CResメッセージに適用されるエンコーディング。
Card	カードは_EMV 3-Dセキュア・プロトコルおよびコア機能仕様_において、決済カードのアカウントと同義です。
Cardholder	カードの発行対象の個人、またはそのカードの使用がオーソリゼーションされた個人。
Challenge	ACSが3DSクライアントと通信し、カード会員とのやり取りを通じて追加情報を取得するプロセス。
Challenge Flow	EMV 3-Dセキュア・プロトコルおよびコア機能仕様で定義される、カード会員の操作に関する3-Dセキュア・フロー。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Term	Definition
Challenge Request(CReq) Message	3DS SDKまたは3DSサーバーによって送信されるEMV 3-Dセキュア・メッセージ。認証プロセスをサポートするためにカード会員からACSに追加情報が送信されます。
Challenge Response(CRes)	CReqメッセージに対するACSレスポンス。カード会員認証の結果を示したり、アプリベース・モデルの場合は、認証の完了にはさらなるカード会員操作が必要であることを示す信号を示したりすることができます。
Consumer Device	カード会員が認証や購入を含む決済アクティビティを実行するのに使用するスマートフォン、ノートPC、またはタブレットなど、カード会員によって使用されるデバイス。
Device Channel	取引の発生元のチャネルを示します。以下のいずれか：• アプリベース（01-APP）• ブラウザベース（02-BRW）• 3DSリクエスター起因（03-3RI）
Device Information	認証プロセスで使用される、コンシューマー・デバイスによって提供されるデータ。
Directory Server (DS)	相互運用ドメインで機能するサーバーコンポーネント。以下を含むいくつかの機能を実行します： 3DSサーバーの認証、3DSサーバーおよびACS間のメッセージのルーティング、3DSサーバー、3DS SDK、および3DSリクエスターの検証。
Directory Server Certificate Authority (DS CA)	相互運用ドメインで機能するコンポーネント。認証局（DS）が選択したデジタル証明書を生成し、3-Dセキュアに参加しているコンポーネントに分配します。通常、DSが接続されている決済システムがCAを操作します。
Directory Server ID (directoryServerID)	決済システムに対して一意の登録済みアプリケーション・プロバイダー識別子（RID）。RIDはISO 7816-5標準によって定義されています。
Electronic Commerce Indicator (ECI)	カード会員を認証するための試行の結果を示す、ACSによって提供される決済システム固有の値。
Frictionless	カード会員の介入なく行われた場合に、認証プロセスを説明するのに使用されます。
Frictionless Flow	EMVCoコア仕様セクション2.5.1で定義される、カード会員が介入しない3-Dセキュア・フロー。
Merchant	決済カードを使用して行われた決済を受理するためにアクワイアラーと契約しているエンティティ。加盟店は、カード番号を取得してカード会員のオンラインショッピング体験を管理し、その後決済認証を行う3DSサーバーに制御を移します。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

Term	Definition
One-Time Passcode (OTP)	コンピューターシステムまたはその他のデジタル・デバイス上の1ログイン・セッションまたは取引に対してのみ有効なパスコード。
Out-of-Band (OOB)	3-Dセキュア・フロー外で並行して実行されるチャレンジ・アクティビティ。最終チャレンジ・リクエストは、ACSによってチェックされるデータの送信には使用されませんが、認証が完了したことのみを通知します。ACS認証方式または実装は3-Dセキュア仕様では定義されません。
Registered Application Provider Identifier (RID)	決済システムに対して一意の登録済みアプリケーション・プロバイダー識別子（RID）。RIDはISO 7816-5標準によって定義されており、ISO/IEC 7816-5登録局によって発行されます。RIDは5バイトです。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

サポート

GPaymentsについて

当社について

GPaymentsはオンライン取引向けの支払い認証製品に特化したオーストラリアの会社で、世界中の多くの国でカードスキーム、金融機関（発行人と取得者の両方）、オンラインサービスプロバイダー、加盟店やカード会員向けにソリューションを提供しています。 15年以上の経験を持つGPaymentsは業界で先駆け・イノベーションを起こしてきた歴史があり、セキュアオンラインコマースのリーダーとしてのポジションを獲得しています。

当社のミッション

当社はオンライン取引向けの堅牢な支払い・認証ソリューションの提供にフォーカスしています。金融機関（発行人と取得者の両方）、サービスプロバイダー、加盟店やカード会員向けにサービスを提供しています。 20年以上の技術開発経験で、当社はセキュアなEコマースのリーダーとして確立しています。

当社の業務

当社はVisa、Mastercard SecureCode、JCB J/Secure、 American Express SafeKeyとDiners Club International ProtectBuyにより証明された3-Dセキュアプロトコルをベースとした、統合された認証製品を全範囲で提供しています。これらの製品は世界で30カ国、国際的な大企業で導入されています。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

なぜ当社か

グローバルマーケット向けの多通貨・多言語サポートを含むオープンかつスケーラブルなソリューションの実績で、当社は支払い認証業界のイノベーターとして確立しています。オンライン支払業界についてのトレンドを調査・評価を常に行うことで進化と革命を起こす努力を怠りません。また当社では大規模なパートナーシップのネットワークを通してシステム統合とカスタマーサポートを提供しています。

GPaymentsについての詳細な情報は下記ウェブサイトをご参照ください。
www.gpayments.com.

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

GPayments
a neurocom company

Language | Login

Our Solutions | Resources | About Us | Contact Us

Frictionless Fraud Prevention Simplified

Securing the payment ecosystem end-to-end with fraud and risk management solutions for financial institutions, payment service providers and merchants.

Explore products | Get in touch

AS SEEN ON FINANCIAL REVIEW

Robust cardholder authentication solutions for online transactions

We provide a complete range of integrated authentication products based on the 3D Secure protocol, Visa Secure, Mastercard Identity Check, JCB J/Secure, American Express SafeKey, UnionPay International 3-D Secure and Discover ProtectBuy.

bambora | ERSTE | OKU | Paynova | UPC | zalando | SecurePay | DS Corporation

Solutions built to help

Payment Gateways & Merchants

- Decrease cart abandonment rates with 3DS2's Frictionless Flow and mobile SDK
- Reduce credit card fraud and eliminate chargebacks via liability shift
- Comply with local regulations, such as PSD2

See Related Products

Card Issuers

- Stay in control of authenticating transactions
- Collects a rich data set for high authentication accuracy
- Reduce credit card fraud

See Related Products

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

お問合せ

この文書内でエラーを発見した場合、もしくは追加のサポートが必要な場合にはGPayments技術サポートまでご連絡ください。 techsupport@gpayments.com.

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

ドキュメントコントロール

更新日時(翻訳日)	ActiveSDK バージョン	ドキュメントバージョン	変更点の詳細
17/02/22(2022/07/29)	**Android**: 2.2.8070 **iOS**: 2.2.8054	2.2.5:1	• リリースノートの更新
03/02/22(2022/07/29)	**Android**: 2.2.8054 **iOS**: 2.2.8054	2.2.4:1	• AndroidX使用時の記載をはじめにに追記 • チャレンジステータスレシーバー内のAndroidおよびiOSのサンプルコードを更新 • **ThreeDSServerAuthResponse**をチャレンジパラメータ内のテーブルに追記しサンプルコードを修正。
24/11/21(2022/11/22)	**Android**: 2.2.7989 **iOS**: 2.2.7989	2.2.3:1	• AMEX、Discover、JCBおよびUnionPay用の3DS SDKにあらかじめ搭載された証明書情報を 設定済ディレクトリサーバのDS_IDの利用 内の表に追記。 • Swiftの対応バージョン用としてPodをアップロード。詳細はCocoaPodsによるインストール に記載。
15/10/21	**Android**: 2.2.7936 **iOS**: 2.2.7936	2.2.2:2	• 統合ガイド内のサンプルコードにあるAndroid **applicationContext**を修正
01/10/21	**Android**: 2.2.7936 **iOS**: 2.2.7936	2.2.2:1	• 統合ガイドに事前設定のディレクトリサーバーDS_IDの使用に関するセクションを追加。 • 統合ガイドにAndroid Dev SDKのサンプルリポジトリリンクを追加。 • 統合ガイドにCocoaPodリポジトリのアクセス方法に関する注意事項を追加。 • 統合ガイドに、ディレクトリサーバー情報にDS証明書を追加する例を追加。 • リリースノートを追加

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

更新日時(翻訳日)	ActiveSDK バージョン	ドキュメントバージョン	変更点の詳細
20/09/21	**Android**: 2.2.7722 **iOS**: 2.2.7876	2.2.1:3	• Androidの開発用SDKガイドを統合ガイドから削除。
20/08/21	**Android**: 2.2.7722 **iOS**: 2.2.7876	2.2.1:2	• CocoaPodでの統合を追加。 • ドキュメントコントロールを追加。
30/07/21	**Android**: 2.2.7722 **iOS**: 2.2.7876	2.2.1:1	• 初回リリース

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

リリースノート

ActiveSDK v2.2.5

[リリース日: 2022/02/17]

SDKビルドバージョン番号(リポジトリ用):

- Android: 2.2.8070
- iOS: 2.2.8054

変更点	説明
チャレンジメッセージ (Android)	challengeDataEntryおよびchallengeHTMLDataEntryのCReqでのフォーマットを修正。

ActiveSDK v2.2.4

[リリース日： 03/02/2022]

SDKビルドバージョン番号(リポジトリ用):

- Android: 2.2.8054
- iOS: 2.2.8054

変更点	説明
AndroidX	Android Support LibraryではなくAndroidXで構築されるように仕様変更しました。
チャレンジUI (iOS)	protocolErrorでcloseメソッドを呼び出したときに、Challenge UIが閉じない不具合を修正しました。
チャレンジUI (iOS)	ACS が cancel 中にエラーを返した場合、cancel と protocolError の両方が発生する不具合を修正しました。この場合、protocolError のみを発生させる必要があります。
チャレンジUI (Android)	再送信ボタンをクリックした際にprotocolErrorが発生した場合、チャレンジUIが閉じない不具合を修正しました。

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

変更点	説明
Fix	SDKMaxTimeoutを実装しました。タイムアウトが発生した場合、SDKはトランザクションUIを閉じるよう修正しました。

ActiveSDK v2.2.3

[リリース日： 24/11/2021]

SDKビルドバージョン番号(リポジトリ用):

- Android: 2.2.7989
- iOS: 2.2.7989

変更点	説明
証明書更新	AMEX、Discover、JCB、UnionPayのディレクトリサーバーの設定済み証明書を更新しました。
パッケージの新フォーマット	XCFrameworkは、すべてのターゲットプラットフォームとアーキテクチャに対して単一の依存関係を持つため、個々のターゲットプラットフォームDeviceやSimulatorを個別に指定する必要はありません。
アクティビティリーク(Android)	クリーンアップを呼び出すと、リソースが適切に解放されず、メモリリークとRuntimeException エラーが発生しました。このエラーは解決済です。
エンコーディングのバグ	SDKのUnicodeデータのエンコーディングに問題がありました。このエラーは解決されました。

ActiveSDK v2.2.2

[リリース日： 01/10/2021]

SDKビルドバージョン番号(リポジトリ用):

- Android: 2.2.7936
- iOS: 2.2.7936

Copyright ©2019 GPayments Pty Ltd. All rights reserved.

変更点	説明
証明書更新	AMEX、Discover、JCB、UnionPayのディレクトリサーバーの設定済み証明書を更新しました。

ActiveSDK v2.2.1

[リリース日: 30/07/2021]

SDKビルドバージョン番号(リポジトリ用):

- Android: 2.2.7722
- iOS: 2.2.7876

変更点	説明
リリース	初回リリース

Copyright ©2019 GPayments Pty Ltd. All rights reserved.