

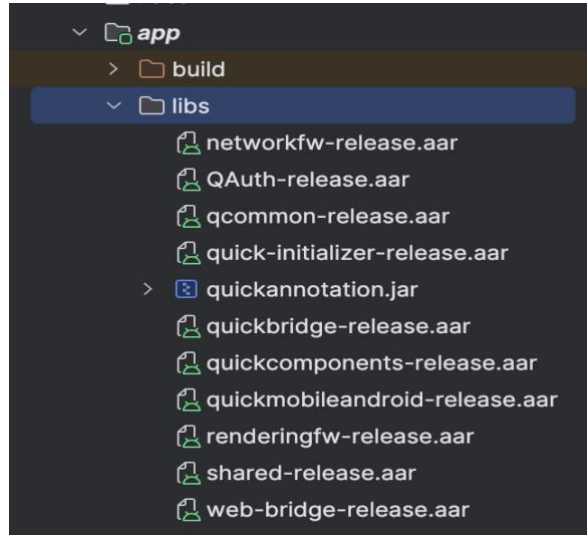
# Plateau Mobile Android SDK Entegrasyonu

Bu dokümanda Plateau Mobile Android SDK entegrasyonunu nasıl yapılacağı anlatılmıştır. Yapılması gerekenler maddeler halinde eklenmiştir.

## 1. SDK Kodlarını İçeren Aar Dosyalarının Kabuk Projeye Dahil Edilmesi

- Aşağıdaki linkten sdk'nın kodlarını içeren aar dosyaların indirilmesi gerekmektedir. Daha sonra bu indirilen aar dosyaları Şekil-1'deki gibi projenizin libs klasörü içerisine atılmalıdır.

Sdk repo linki: <https://github.com/STechQ/dist-plateau-mobile-android>



Şekil 1- Aar dosyalarının atılması gereken dizin

## 2. Gradle Bağımlılıklarının Eklenmesi

- Aar dosyaları projeye dahil edildikten sonra module seviyesinde build.gradle dosyasındaki android tagı içerisine öncelikle data binding Şekil-2'deki gibi eklenmelidir.

```
dataBinding {  
    enabled = true  
}  
  
buildFeatures {  
    dataBinding = true  
}
```

Şekil 2- Data binding eklenmesi

- Daha sonra dependencies tag'ı içerisinde sdk'nın aar dosyalarını implemente etmeliyiz. QAuth kütüphanesinin implementasyonu zorunlu değildir. Auth işlemleri yapılacaksa implemente edilmelidir. Şekil-3'deki gibi implementasyon yapılmalıdır.

```
dependencies {
    // Plateau Mobile SDK Files
    implementation fileTree(dir: 'libs', include: ['*.jar'])

    implementation (files("libs/quickcomponents-release.aar"))
    implementation (files("libs/qcommon-release.aar"))
    implementation (files("libs/renderingfw-release.aar"))
    implementation (files("libs/networkfw-release.aar"))
    implementation (files("libs/quickbridge-release.aar"))
    implementation (files("libs/quickmobileandroid-release.aar"))
    implementation (files("libs/shared-release.aar"))
    implementation (files("libs/quick-initializer-release.aar"))
    implementation (files("libs/web-bridge-release.aar"))

    // Optional
    //implementation (files("libs/QAuth-release.aar"))
}
```

Şekil 3- SDK dosyalarının implementasyonu

- Daha sonra yine dependencies tag'ı içerisinde sdk'nın ihtiyaç duyduğu third party kütüphanelerin implementasyonları Şekil 4 ve Şekil 5'deki gibi yapılmalıdır. Auth kütüphaneleri yine opsiyoneldir.

```
// Third Party Libraries
implementation 'androidx.core:core-ktx:1.7.0'
implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
implementation 'androidx.activity:activity:1.8.0'
implementation 'androidx.appcompat:appcompat:1.2.0'
implementation 'com.google.android.material:material:1.10.0'
testImplementation 'junit:junit:4.13.2'
androidTestImplementation 'androidx.test.ext:junit:1.1.5'
androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'

implementation 'com.facebook.soloader:soloader:0.10.4'
implementation 'com.airbnb.android:lottie:4.2.0'
implementation 'com.google.dagger:dagger-android:2.35.1'
implementation 'com.google.code.gson:gson:2.8.6'
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'
implementation 'io.reactivex.rxjava3:rxandroid:3.0.0'
implementation 'com.squareup.retrofit2:converter-scalars:2.9.0'
implementation 'com.squareup.retrofit2:adapter-rxjava3:2.9.0'
implementation 'com.jakewharton.timber:timber:4.7.1'
implementation 'org.conscrypt:conscrypt-android:2.2.1'
implementation 'com.squareup.okhttp3:okhttp:4.9.3'
implementation 'commons-beanutils:commons-beanutils:1.9.4'
implementation 'org.apache.commons:commons-lang3:3.11'
implementation 'com.eclipsesource.j2v8:j2v8:6.2.10aas'
implementation 'com.facebook.yoga:yoga-layout:2.0.0'

implementation 'com.github.bumstech.glide:glide:4.14.2'
implementation 'com.jakewharton.rxbinding4:rxbinding4-core:4.0.0'
implementation 'commons-codec:commons-codec:1.15'
implementation 'com.squareup.okhttp3:logging-interceptor:4.9.3'
annotationProcessor 'com.github.bumstech.glide:compiler:4.14.2'
annotationProcessor 'com.google.dagger:dagger-android-processor:2.27'
annotationProcessor 'com.google.dagger:dagger-compiler:2.27'

implementation 'androidx.room:room-runtime:2.4.0-alpha03'
annotationProcessor 'androidx.room:room-compiler:2.4.0-alpha03'

implementation 'com.google.firebase:firebase-analytics:17.4.1'
implementation platform('com.google.firebase:firebase-bom:32.1.0')
implementation 'com.google.firebase:firebase-crashlytics'

implementation 'com.squareup.retrofit2:retrofit:2.9.0'
implementation 'androidx.security:security-crypto:1.1.0-alpha03'
implementation 'com.github.PhilJay:MPAndroidChart:v3.1.0'

// Optional
//implementation 'net.openid:appauth:0.11.1'
//implementation 'com.auth0.android:jwtdecode:2.0.0'
```

Şekil 4- Third party libraries

Şekil 5- Third party libraries devamı

### 3. Kabuğa gerekli metotların eklenmesi

#### 3.1 Eklenmesi zorunlu olanlar

- Gradle'daki gerekli implementasyonlar yapıldıktan sonra kabuk activity / fragment'a sdk implementasyonu yapmamız gerekiyor. Öncelikle Şekil 6'daki gibi gerekli implementasyon ve değişken tanımlamaları yapılmalıdır.

```
class MainActivity :  
    AppCompatActivity(),  
    QuickService.AsyncInitialListener,  
    ScreenNavController,  
    QuickService.LoadingJsonServiceListener,  
    QuickService.QuickCallBackListener,  
    QuickService.QuickActivityController {  
  
    private var baseUrl = "https://otogb7m8y8f23d18ym35w7.z6.web.core.windows.net/"  
    private var settingsUrl = "settings/settings_mobile.json"  
    private var quickService: QuickService? = null  
    private lateinit var lottieAnimationView: LottieAnimationView  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        lottieAnimationView = findViewById(R.id.lottieLoading)  
        setLoadingJsonLocal()  
  
        initQuickService()  
    }  
}
```

Şekil 6- Gerekli implementasyonlar ve değişken tanımlamaları

- Daha sonra SDK'nın ayağa kalkabilmesi için gerekli SDK initialize kodları Şekil 7'deki gibi eklenmelidir. Burada sertifika eklemek isteğe bağlıdır.

```
private fun initQuickService() {  
    // If you want to add a certificate, you should add it like this and use setQuickCertificate  
    val certificateBuilder = QuickCertificate.Builder.newQuickCertificate()  
    certificateBuilder.add( pattern: "host", ...pins: "public_key")  
  
    SoLoader.init(androidApplication, nativeExopackage: false)  
  
    val builder = QuickSdk.Builder.newInstance()  
        .setSettingsUrl(settingsUrl)  
        .setAppId(null)  
        .setLanguage("tr-TR")  
        .maxRequestRetryCount( count: 0)  
        .timeOutRequestSeconds( seconds: 60)  
        .setDeepLinkParameters(null)  
        .setQuickCertificate(certificateBuilder.build())  
        .setClientCustomFunctionTriggerListener(this)  
        .setPlatformInfo(QPlatform(baseContext).platformInfo)  
        .setBaseUrl(baseUrl)  
  
    quickService = builder.build( navController: this).quickService  
    quickService!!.initializeAsync( initializeListener: this)  
}
```

Şekil 7- SDK initialize builder kodları

- Daha sonra override edilmesi gereken bazı metotlar bulunmaktadır. Bu metotlar aşağıdaki Şekil 8, Şekil 9, Şekil 10, Şekil 11 ve Şekil 12 belirtilmiştir.

```
// Required for page transition
override fun onQuickFragmentCreated(p0: Boolean, p1: QFragment, p2: String) {
    if (!isFinishing) {
        val fragmentTransaction = supportFragmentManager.beginTransaction()
        fragmentTransaction.addToBackStack(p2)

        fragmentTransaction.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_OPEN).add(
            R.id.q_content_fragment_layout, p1,
            p2
        )

        if (!p1.isStateSaved()) {
            fragmentTransaction.commit()
        } else {
            fragmentTransaction.commitAllowingStateLoss()
        }
    }
}

override fun onQuickBackPressed() {
    super.onBackPressed()
}

@SuppressLint("MissingSuperCall")
override fun onBackPressed() {
    quickService!!.handleBack().subscribe { handled ->
        if (!handled) {
            super@MainActivity.onBackPressed()
        }
    }
}
```

Şekil 8- Override edilmesi gereken metotlar

```
override fun showLoading() {
    lottieAnimationView.post(Runnable {
        lottieAnimationView.setVisibility(View.VISIBLE)
        lottieAnimationView.playAnimation()
    })
}

override fun hideLoading() {
    lottieAnimationView.post(Runnable {
        lottieAnimationView.setVisibility(View.GONE)
        lottieAnimationView.pauseAnimation()
    })
}

override fun onDestroy() {
    super.onDestroy()
    release()
}
```

Şekil 9- Override edilmesi gereken metotlar

```

// Consent information to be received from the application using the SDK
// Dummy data added for example purposes
override fun callFunction(
    functionName: String?, params: QV8Element?,
    resultListener: FunctionCallbackListener
): Boolean {
    val v8Object = QV8Object()
    val resultData = QV8Object()
    var handled = false

    when (functionName) {
        "GetToken" -> {
            v8Object.add( property: "token", value: "123456789")
            handled = true
        }

        "GetUserInfo" -> {
            v8Object.add( property: "UserNameSurname", value: "testNameSurname")
            v8Object.add( property: "SicilNo", value: "123456")
            handled = true
        }

        else -> Log.e( tag: "MyApp", msg: "Default Case")
    }

    if (handled) {
        resultData.add( property: "isSuccess", value: true)
        resultData.add( property: "retVal", v8Object)
        resultListener.onFunctionResult(resultData)
    }

    return handled
}

```

Şekil 10- Override edilmesi gereken metotlar

```

override fun getUserOrDeviceInfo(): HashMap<String, String> {
    val returnObject = java.util.HashMap<String, String>()
    val isLogin = true

    if (isLogin) {
        returnObject["user"] = "123" //userId
    } else {
        returnObject["device"] = "123" //deviceId
    }

    return returnObject
}

private fun release() {
    quickService?.release()
    quickService = null
}

override fun onInitialized(quickService: QuickService) {
    quickService.injectActivity( activity: this)
    quickService.startRender( url: null)
}

override fun setLoadingJson(loadingJson: String) {
    val compositionTask = LottieCompositionFactory.fromString(loadingJson, cacheKey: null)
    compositionTask.addListener { result: LottieComposition ->
        lottieAnimationView.setCacheComposition(true)
        lottieAnimationView.setComposition(result)
        lottieAnimationView.setMaxFrame(31)
        lottieAnimationView.setMinFrame(1)
    }
}

```

Şekil 11- Override edilmesi gereken metotlar

```

override fun setAppId(p0: String) {
    // no-opt
}

override fun stopMiniApp() {
    // no-opt
}

override fun setTheme(p0: QThemeAttributes) {
    // no opt
}

override fun getAndroidApplication(): Application {
    return application
}

override fun setPageAndStatusColor(p0: MutableMap<String, Any>?) {
    // no opt
}

```

Şekil 12- Override edilmesi gereken metotlar

```

override fun clickNavigation(p0: Any?) {
    // no opt
}

override fun goNativePage(p0: String?, p1: MutableMap<String, Any>?, p2: String?) {
    // no opt
}

override fun setLoadingJsonLocal() {
    // no-opt
}

override fun addNavigationComponent(component: QBaseComponent<*>?) {
    // no-opt
}

override fun closeNavigation() {
    // no-opt
}

override fun openNavigation() {
    // no-opt
}

```

Şekil 13- Override edilmesi gereken metotlar

### 3. 2 Eklenmesi isteğe bağlı olanlar

- Eğer kullanıcı izinleri ile ilgili işlemleri (kamera, konum izni vs.) kabukta kendiniz yönetmek istiyorsanız kabukta `QRuntimePermissionHandler` interface'i implemente edilmelidir. Sonrasında Şekil 13'deki metotlar override edilmelidir.

```
override fun requestRuntimePermissions(  
    permission: Array<out String>,  
    permissionListener: QRuntimePermissionListener  
) {  
    if (hasPermission(*permission)) {  
        permissionListener.onRuntimePermissionGranted(permission)  
    } else {  
        this.permissionListener = permissionListener  
        // request runtime permission  
    }  
}  
  
override fun cancelPermission() {  
    // No-op  
}  
  
override fun confirmPermission() {  
    // No-op  
}  
  
override fun hasPermission(vararg p0: String): Boolean {  
    p0.forEach {  
        if (ContextCompat.checkSelfPermission(  
            applicationContext,  
            it  
        ) != PackageManager.PERMISSION_GRANTED  
        ) {  
            return false  
        }  
    }  
    return true  
}
```

Şekil 14- Override edilmesi gereken permission metotları



#### 4. Kabuk activity / fragment'in örnek xml dosyasının oluşturulması

- Kabuğun xml'i Şekil 14'deki örnek gibi oluşturulmalıdır.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/q_drawerLayout"
    android:layout_width="match_parent"
    android:fitsSystemWindows="true"
    android:layout_height="match_parent">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <androidx.fragment.app.FragmentContainerView
            android:id="@+id/q_content_fragment_layout"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">

        </androidx.fragment.app.FragmentContainerView>

        <com.airbnb.lottie.LottieAnimationView
            android:id="@+id/lottieLoading"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_centerInParent="true"
            android:layout_gravity="center"
            android:background="#60000000"
            android:fitsSystemWindows="true"
            android:scaleType="centerInside"
            app:lottie_autoPlay="true"
            app:lottie_loop="true"
            app:lottie_speed="1" />

    </RelativeLayout>

</androidx.drawerlayout.widget.DrawerLayout>
```

Şekil 15- Örnek kabuk xml dosyası

#### SDK ile ilgili bilgiler

- Bazı versiyon bilgileri şu şekildedir:  
minSdkVersion = 23  
targetSdkVersion = 34  
compileSdkVersion = 34

#### Destek ve İletişim

- Entegrasyon sırasında sorun yaşamanız durumunda [plateaustudiomobile@softtech.com.tr](mailto:plateaustudiomobile@softtech.com.tr) adresine mail atabilirsiniz.