



راهنمای استفاده از SDK درون خرید برای توسعه دهندگان اندروید

تاریخچه بازنگری

تاریخ	نسخه	شرح
آذر 97	1.0.0	سند ایجاد

• هدف

هدف از تهیه این سند ارائه راهنمایی مناسب برای توسعه دهندگان و برنامه نویسانی است که میخواهند از SDK درون خرید پکپی در نرم افزارهای تولیدی خود استفاده نمایند .

• محدوده

محدوده این سند تنها مواردی که مستقیماً با فرآیند درون خرید در ارتباط است را شامل میشود و مواردی همچون آموزش برنامه نویسی و یا کاربری ابزارهای برنامه نویسی را شامل نمیشود . لذا فرض بر آن است که خواننده آشنایی نسبی با برنامه نویسی و تولید نرم افزار در بستر Android را دارا می باشد . حداقل این آشنایی عبارتست از:

- تسلط بر برنامه نویسی اندروید
- آشنا با مفاهیم خرید درون برنامه

- راهنمای استفاده از SDK درون خرید پکی

- مراحل تعریف و استفاده از خدمات درون خرید به شرح زیر میباشد :
- اضافه کردن کتابخانه ی پرداخت درون خرید و پرداخت پکی
- معرفی و کاربرد متدهای SDK پکی

- اضافه کردن کتابخانه های پرداخت درون خرید پک پی

فایل کتابخانه های زیر را به پوشه libs پروژه خود انتقال دهید.

iablib-release-1.0.aar
inAppPurchase.aar
inAppSDK.aar

در فایل build.gradle مازول اصلی پروژه خود دستور زیر را اضافه نمایید.

```
repositories{
    flatDir{
        dirs 'libs'
    }
}
```

به صورت زیر، کتابخانه پک پی را به قسمت dependencies پروژه خود اضافه کنید.

```
dependencies {

    compile fileTree(dir: 'libs', include: ['*.jar'])

    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    implementation 'com.android.support:appcompat-v7:27.1.1'
    implementation 'com.squareup.retrofit2:retrofit:2.4.0'
    implementation 'com.squareup.retrofit2:converter-gson:2.4.0'
    implementation 'com.squareup.okhttp3:logging-interceptor:3.10.0'
    implementation 'com.android.support:design:27.1.1'
    implementation 'com.android.support.constraint:constraint-layout:1.0.2'

    implementation project(":iablib")
    implementation(name:'inAppPurchase', ext:'aar')
    implementation(name:'inAppSDK', ext:'aar')
}
```

gradle پروژه را سینک نمایید تا کتابخانه پک پی قابل استفاده شود.

• متدهای کتابخانه خرید پکی

جدول -1 لیست متدهای لایبری خرید درون برنامه ای اندروید

نام متد	توضیح
void purchase (InternalPurchaseInformationDto dto, AsyncCallback asyncCallback))	درخواست خرید محصولات یک بار خرید و چند بار خرید
void subscribe(InternalPurchaseInformationDto dto, AsyncCallback asyncCallback))	درخواست خرید محصولات آبونمانی
void getSubscriptionStatus(InternalPurchaseInformationDto dto, AsyncCallback asyncCallback)	دریافت وضعیت محصول آبونمانی
void consumablePurchaseReport(InternalPurchaseInformationDto dto, AsyncCallback asyncCallback)	گرفتن گزارش خرید محصولات چند بار خرید. دریافت گزارش محصولات که خریداری کرده و نوع محصولات ان از نوع چند بار خرید است.
void subscriptionPurchaseReport)InternalPurchaseInformationDto dto, AsyncCallback asyncCallback)	گزارش محصولات آبونمانی

- نحوه استفاده از متدهای لایبری

- معرفی کلاس `InternalPurchaseInformationDto`

این کلاس ، کلاس ارتباطی بین توسعه دهنده و sdk است .اطلاعات مورد نیاز هر متد از طریق آبجکتی که از این کلاس ساخته می شود به sdk ارسال می شود .

نام فیلد	نوع فیلد	توضیح
<code>serviceCode</code>	String	کد سرویس
<code>productCode</code>	Long	کد محصول
<code>refreshToken</code>	String	مقدار این پارامتر را باید از پنل در برنامه خود قرار دهید
<code>clientId</code>	String	مقدار این پارامتر را باید از پنل در برنامه خود قرار دهید
<code>clientSecretId</code>	String	مقدار این پارامتر را باید از پنل در برنامه خود قرار دهید
<code>operator</code>	String	نوع اپراتور
<code>userNumber</code>	String	شماره تلفن کاربر

به منظور استفاده از هر کدام از متد های فوق نیاز دارید تا یک آبجکت از کلاس فوق بسازید .و مقدارهای مور نیاز هر متد را داخل آن ست کنید.

```
InternalPurchaseInformationDto internalPurchaseInformationDto = new
InternalPurchaseInformationDto();
```

سپس از کلاس `ServiceClient` یک آبجکت به صورت زیر بسازید :

```
ServiceClient serviceClient = new ServiceClient(getBaseContext(),
internalPurchaseInformationDto);
```

پس از آن میتوانید تمامی متد های موجود در برنامه را با استفاده از آبجکت ساخته شده از کلاس `ServiceClient` صدا بزنید .

- import کردن کلاسهای مورد نیاز

```
import ir.packpay.iablib.data.dto.InternalPurchaseInformationDto;
import ir.packpay.iablib.data.dto.FaultResponse;
import ir.packpay.iablib.handler.AsyncCallback;
import ir.packpay.iablib.model.ServiceClient;
```

- تشریح بکارگیری متد های لایبری

- متد purchase

از این متد جهت درخواست خرید محصولات یک بار خرید و چند بار خرید می توانید استفاده کنید.
 ابجکت InternalPurchaseInformationDto به عنوان پارامتر ورودی این متد شامل مقادیر زیر می باشد :

```
InternalPurchaseInformationDto dto = new InternalPurchaseInformationDto()
    .setRefreshToken(REFRESH_TOKEN)
    .setClientId(CLIENT_ID)
    .setClientSecretId(SECRET_ID)
    .setServiceCode("201")
    .setProductCode(201002L);
```

سپس می توانید متد

```
purchase(InternalPurchaseInformationDto dto, AsyncCallback asyncCallback())
```

را صدا بزنید .

```
ServiceClient serviceClient = new ServiceClient(getBaseContext(),
internalPurchaseInformationDto);
```

```
serviceClient.purchase(internalPurchaseInformationDto, new AsyncCallback() {
    @Override
    public void onSuccess(Object serverResponse) {
        Log.i(MainActivity.class.getSimpleName(), "calling purchase Success");
    }
}) ;
```

```
    @Override
    public void onFailed(FaultResponse faultResponse) {
        Log.i(MainActivity.class.getSimpleName(), "calling purchase failed");
    }
} ;
```

- متد subscribe

می توانید از این متد برای درخواست خرید محصولات آبونمانی استفاده کنید.

ابجکت InternalPurchaseInformationDto به عنوان پارامتر ورودی این متد شامل مقادیر زیر می باشد :

```
InternalPurchaseInformationDto dto = new InternalPurchaseInformationDto()  
    .setRefreshToken(REFRESH_TOKEN)  
    .setClientId(CLIENT_ID)  
    .setClientSecretId(SECRET_ID)  
    .setServiceCode("201")  
    .setProductCode(201002L);
```

سپس می توانید با ابجکت ServiceClient ایجاد شده متد

```
subscribe (InternalPurchaseInformationDto dto, AsyncCallback asyncCallback)
```

را صدا بزنید.

```
InternalPurchaseInformationDto dto = new InternalPurchaseInformationDto()  
    .setRefreshToken(REFRESH_TOKEN)  
    .setClientId(CLIENT_ID)  
    .setClientSecretId(SECRET_ID)  
    .setServiceCode("201")  
    .setProductCode(201002L);  
  
ServiceClient serviceClient = new ServiceClient(getBaseContext(), dto);  
  
serviceClient.subscribe(dto, new AsyncCallback() {  
    @Override  
    public void onSuccess(Object serverResponse) {  
        Log.i(MainActivity.class.getSimpleName(), "calling subscribe Success");  
    }  
  
    @Override  
    public void onFailed(FaultResponse faultResponse) {  
        Log.i(MainActivity.class.getSimpleName(), "calling subscribe failed");  
    }  
});
```

- متد `getSubscriptionStatus`

میتوانید از این متد جهت دریافت وضعیت محصول آبونمانی استفاده نمایید.

ابجکت `InternalPurchaseInformationDto` به عنوان پارامتر ورودی این متد شامل مقادیر زیر می باشد :

```
InternalPurchaseInformationDto dto = new InternalPurchaseInformationDto()  
    .setRefreshToken(REFRESH_TOKEN)  
    .setClientId(CLIENT_ID)  
    .setOperator("MCI")  
    .setClientSecretId(SECRET_ID)  
    .setServiceCode("201")  
    .setProductCode(201002L);
```

سپس می توانید با آجکت `ServiceClient` ایجاد شده متد

```
getSubscriptionStatus (  
    InternalPurchaseInformationDto dto,  
    AsyncCallback syncCallback  
)
```

را صدا بزنید

```
ServiceClient serviceClient = new ServiceClient(getBaseContext(), dto);  
  
serviceClient.getSubscriptionStatus(dto, new AsyncCallback() {  
    @Override  
    public void onSuccess(Object serverResponse) {  
        Log.i(MainActivity.class.getSimpleName(), "calling unsubscribe  
onOAuthSuccess");  
    }  
  
    @Override  
    public void onFailed(FaultResponse faultResponse) {  
        Log.i(MainActivity.class.getSimpleName(), "calling unsubscribe  
onOAuthFailed");  
    }  
});
```

به ازای اجرای موفق این متد یک آجکت `json` به صورت زیر دریافت خواهید کرد.

```
{  
    "status": "0",  
    "message": "successful",  
    "auto_charge": "ACTIVE"  
}
```

- متد consumablePurchaseReport

می‌توانید از این متد برای دریافت گزارش خرید محصولات چند بار خرید استفاده کنید.

ابجکت InternalPurchaseInformationDto به عنوان پارامتر ورودی این متد شامل مقادیر زیر می باشد :

```
InternalPurchaseInformationDto dto = new InternalPurchaseInformationDto()
    .setRefreshToken(REFRESH_TOKEN)
    .setClientId(CLIENT_ID)
    .setClientSecretId(SECRET_ID)
    .setServiceCode("201")
    .setProductCode(201999L)
    .setOperator("MCI")
    .setUserNumber("09122979012")
;
```

سپس می‌توانید با آجکت ServiceClient ایجاد شده متد

```
consumablePurchaseReport (
InternalPurchaseInformationDto dto,
AsyncCallback asyncCallback
)
```

را صدا بزنید.

```
ServiceClient serviceClient = new ServiceClient(getBaseContext(), dto);
serviceClient.consumablePurchaseReport(dto, new AsyncCallback() {
    @Override
    public void onSuccess(Object serverResponse) {
        Log.i(MainActivity.class.getSimpleName(), "calling unsubscribe
onOAuthSuccess");
    }

    @Override
    public void onFailed(FaultResponse faultResponse) {
        Log.i(MainActivity.class.getSimpleName(), "calling unsubscribe
onOAuthFailed");
    }
});
```

به ازای اجرای موفق این متد یک آجکت JSON به صورت زیر دریافت خواهید کرد.

```
{
  "status": "0",
  "message": "successful",
  "data": [
    {
      "service_code": "201",
      "product_code": "201999",
      "first_purchase_time": 1535282361646,
      "total_number_of_purchases": 1
    }
  ]
}
```


- متد `subscribePurchaseReport`

جهت گرفتن گزارش محصولات آبونمائی از این متد استفاده نمایید.

ابجکت `InternalPurchaseInformationDto` به عنوان پارامتر ورودی این متد شامل مقادیر زیر می باشد :

```
InternalPurchaseInformationDto dto = new InternalPurchaseInformationDto()  
    .setRefreshToken(REFRESH_TOKEN)  
    .setClientId(CLIENT_ID)  
    .setClientSecretId(SECRET_ID)  
    .setServiceCode("201")  
    .setProductCode(201005L)  
    .setOperator("MCI")  
    .setUserNumber("9122979012")  
;
```

سپس می توانید با اَبجکت `ServiceClient` ایجاد شده متد

`subscriptionPurchaseReport (InternalPurchaseInformationDto dto, AsyncCallback)`

را صدا بزنید

```
ServiceClient serviceClient = new ServiceClient(getBaseContext(), dto);  
serviceClient.subscriptionPurchaseReport(dto, new AsyncCallback() {  
    @Override  
    public void onSuccess(Object serverResponse) {  
        Log.i(MainActivity.class.getSimpleName(), "calling unsubscribe  
onOAuthSuccess");  
    }  
  
    @Override  
    public void onFailed(FaultResponse faultResponse) {  
        Log.i(MainActivity.class.getSimpleName(), "calling unsubscribe  
onOAuthFailed");  
    }  
});
```

به ازای اجرای موفق این متد یک اَبجکت `JSON` به صورت زیر دریافت خواهید کرد.

```
{  
  "status": "0",  
  "message": "successful",  
  "data": [  
    {  
      "service_code": "201",  
      "product_code": "201005",  
      "purchase_time": 1535272912343,  
      "expire_time": 1535359312343,  
      "product_status": "ACTIVE",  
      "auto_charge": "ACTIVE"  
    }  
  ]  
}
```