

In-app Billing Overview ---

역자 .. 김경미 (nolleh7707@gmail.com / <http://nolleh.tistory.com>)

인앱 구매를 위해 제공하는 구글플레이의 서비스.

특정 상품에 대한 결제 요청을 어플리케이션에서 보내야 이용할 수 있다.

그러면 서비스가 지불의 유효성, 재무적 트랜잭션을 모두 포함해서 트랜잭션을 관리한다. 결제가 모두 완료되면, 어플리케이션에게로 구매 번호나, 구매시간이나, 가격에 대한 정보를 보낸다.

즉 어플리케이션은 가격 지불에 대한 (재무적) 트랜잭션은 전혀 신경쓰지 않아도 된다.

Product and Purchase Types

인앱빌링은 앱이 유연하게 화폐화 하도록 (monetize) 상품타입과 구매타입을 제공한다. 모든 경우에 구글플레이의 개발자 콘솔에서 상품타입, 구매타입, SKU, 가격, 설명등을 기재해야한다.

Product Types

인앱빌링에서, 상품타입에는 두가지가 있다. In – app Products 와 subscriptions 이다 . 두 타입은 많은 차이와 특징이 있지만 인앱빌링 api 는 두 상품 타입을 같은 소통모델(같은 방식으로 다룸)을 사용한다.

- in-app products - 한번에 하나 구매하는 상품. 한번 사면 청약철회 화면이 없다. 개발자에게 직접 연락해야 환불이 가능하다. 유저 계정마다 관리되는 형태 (managed per user account)로 팔 수도 , 관리하지 않는 (no managed) 형태의 구매타입 둘다 가능하다. 인앱상품은 항상 명시적으로 하나의 앱과만 관계가 있다. 즉, 어떤 앱은 다른 앱에서 발매한 인앱상품을 구매할 수 없다. 모든 인앱빌링 버전에서 지원된다.
- Subscriptions – 개발자가 정한, 기간을 두고 재귀적으로 결제를 수행해야하는 형태. 이 상품을 사면, 구글플레이와 그 지불 처리자(프로세서) 가 자동으로 특정 기간마다 청구한다. 한번 유저가 이 상품을 구매하면, 유저의 승인없이 결제가 된다. 언제든지 이 상품구매를 취소할 수 있다. 이 상품은 managed per user account 구매타입을 이용해야한다. 인앱상품처럼 환불 화면은 없다.

Purchase Types

인앱 빌링은 두가지 구매 타입을 제공한다. 계정마다 관리되는 형태와 관리하지 않는 형태이다. 구매타입은 구글플레이가 구매를 어떻게 관리하고 추적하는지 나타낸다.

- managed per user account – 구글플레이에서 유저가 한번만 살 수 있다. 한번 이 구매타입의 상품을 사면, 구글플레이는 영구적으로 그 트랜잭션에 대한 기록을 남긴다. 이 기록은 나중에 특정유저가 구매를 했었는지 구글플레이에게 질의할 수

있도록 제공한다. 이 타입은 게임 레벨이나 어플리케이션에 적용되는 효과등을 설정할때 유용하다 .

- unmanaged – 구글플레이에 트랜잭션에 대한 정보가 저장되지 않는 형태. 따라서 개발자가 직접 트랜잭션을 관리해야 한다. 소모형 아이템을 설정할때 유용하다.

In – app Billing Architecture

인앱빌링 서비스에 접속하는 어플리케이션은 디바이스에 설치되어 있는 구글플레이에서 노출하는 api에 접근한다 . 구글플레이는 비동기적으로 구글플레이 서버와 어플리케이션 사이에 빌링 요청 메시지를 전달하고 응답하기위해 메시지루프를 이용한다. 즉 어플리케이션은 구글플레이서버와 직접적으로 소통하지 않는다. 대신에 , 빌링요청을 구글플레이 어플리케이션으로 IPC (interprocess communication) 로 보내고 비동기 브로드캐스트 인텐트를 통해 응답한다. 따라서 클라이언트 어플리케이션은 빌링요청을 위한 네트워킹이 필요없다.

어떤 인앱 빌링 구현체는 내용물을 전달하고 트랜잭션을 유효화하기 위한 개인적인 리모트 서버를 가질 수 있지만 이는 생략가능하다. 리모트 서버는 유저의 단말기로 전송이 필요한 디지털 콘텐츠(미디어파일, 사진) 를 판매할 때 유용할 수 있다. 또는 인앱빌링 보안 관리 차원에서 유저 트랜잭션 히스토리를 관리할때 사용할 수 있다.

일반적인 인앱빌링 구현은 다음 3가지 구성요소와 관계가 있다.

A Service (BillingService)

- 어플리케이션의 구매 메시지를 처리하고 빌링 요청을 구글플레이의 인앱빌링 서비스로 보냄.

A BroadcastReceiver (BilingReceiver)

- 구글플레이로부터 비동기적으로 빌링응답을 받음.

A security component (Security)

- 보안관련 작업을 수행. 시그니처 확인이나 한번뿐인 난수생성등.

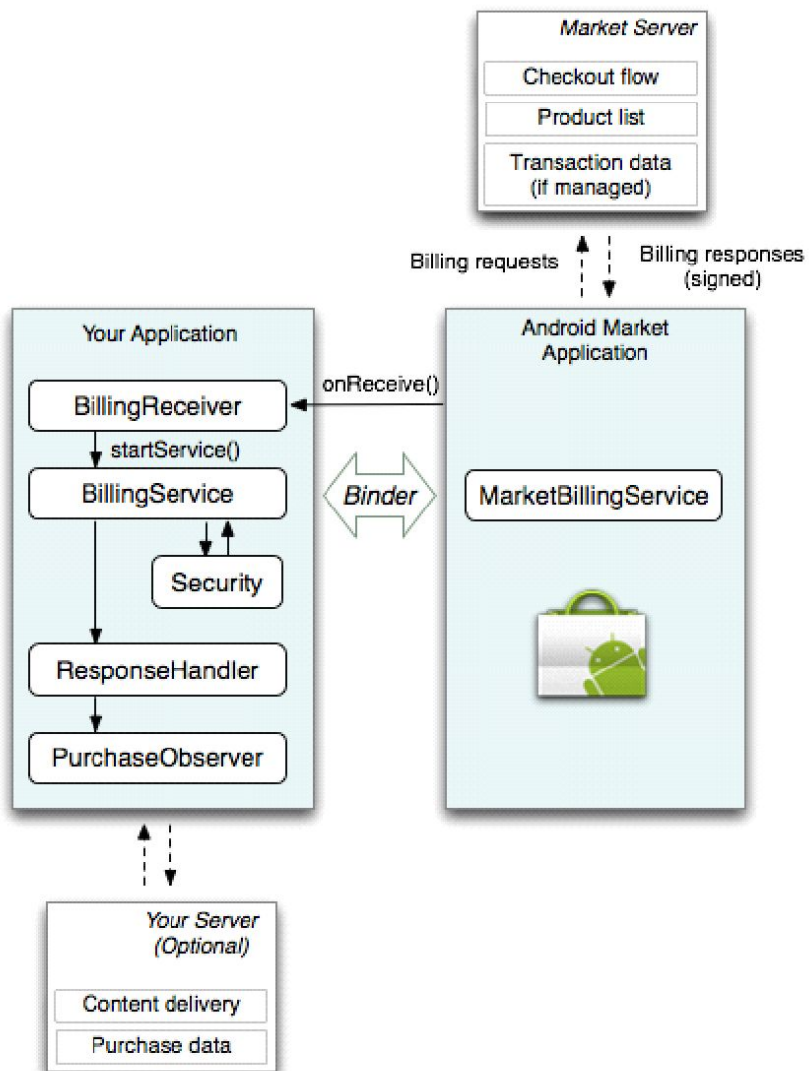
다음과 같은 2 요소를 추가적으로 구성할 수 있다.

A responseHandler (ResponseHandler)

- 어플리케이션의 구매관련 처리를 제공(noti피케이션, 에러 , 상태 메시지)

An Observer (PurchaseObserver)

- 어플리케이션이 구매 상태와 정보를 통해 콜백을 받을 수 있음



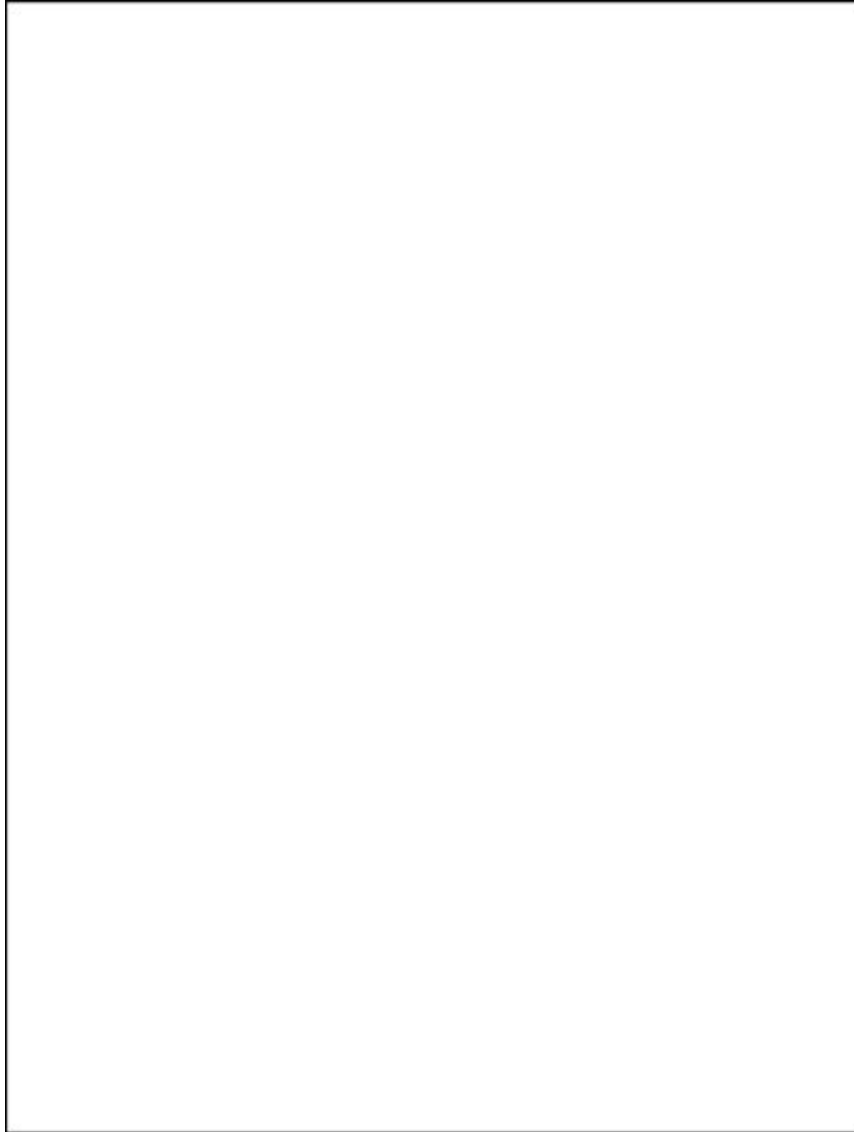


Figure 1. Your application sends and receives billing messages through the Google Play application, which handles all communication with the Google Play server.

유저 구매이력을 저장하고 이를 정렬하여 보여주는 형태를 제공할 수도 있다.

In – app Billing Messages

유저가 구매를 할때 어플리케이션은 빌링요청을 구글플레이의 인앱빌링서비스 (MarkeyBillingService) 로 IPC메서드 콜을 통해 보낸다 .

구글플레이 어플리케이션은 모든 빌링요청에 대응하며 상태 noti피케이션과 같은 다른 정보를 제공한다. 다음 섹션은 기본적인 요청 – 응답에 관한 메시징을 설명한다.

In-app billing Requests

설명했듯이 , 인앱빌링 요청을 구글플레이의 노출된 인터페이스의 ipc 메서드 (`sendBillingRequest()`) 를 통해 보낸다. 이 인터페이스는 AIDL (Android Interface Definition Language) 파일에 정의가 되어있다.

`sendBillingRequest` 메서드는 하나의 번들 파라미터를 갖고있다. 이 번들은 key-value 쌍을 가진 개발자가 빌링요청에 보내고 싶어하는 정보를 전달한다. 빌링요청의 타입과 같은 정보나, 상품타입이나 구매타입, 요청을 만드는 어플리케이션과 같은 정보 같은 것이다.

이 번들에는 BILLING_REQUEST 키가 반드시 들어가야 한다. 이 키는 빌링요청의 타입을 정한다. 다음 다섯가지 중 하나를 value 로 가진다.

CHECK_BILLING_SUPPORTED

- **구글플레이 어플리케이션이 인앱빌링에 관한 검증을 제공.** 인앱빌링을 위한 UI를 인에이블/디스에이블시키고 싶다면 이용하는게 편함.

REQUEST_PURCHASE

- 이 요청은 구글플레이 어플리케이션에게 구매 메시지를 보내는 **기본적인 플러그.** 구글플레이가 체크아웃 인터페이스를 유저에게 보여줌으로써 재무적 트랜잭션에 관한 모든 작업을 진행한다.

GET_PURCHASE_INFORMATION

- **이 요청은 구매 상태 변경에 대한 정보를 구해낸다.** 구매의 상태변경은 구매요청의 빌링이 성공적으로 이뤄지거나 체크아웃 트랜잭션중에 사용자가 요청을 취소했을때 이뤄진다. 또 이전 상품 구매를 취소할때도 발생한다.구글플레이는 이런 구매상태의 변경시 어플리케이션이 트랜잭션 정보를 보존할수 있도록 이를 알려준다.

CONFIRM_NOTIFICATIONS

- **이 리퀘스트는 구매 상태 변경의 정보를 받았음을 알린다.** 구글플레이는 구매 상태 변경정보를 받았음을 어플리케이션이 확인을 보내줄때까지 계속해서 보낸다.

RESTORE_TRANSACTIONS

- **managed purchases 와 subscriptions를 위해 트랜잭션 상태를 복원하는 요청.** 이 요청은 유저의 트랜잭션 상태를 어플리케이션이 복원될 필요가 있을때 , 예를 들면 어플리케이션이 재 설치 됐을때나 어떤 디바이스에 처음 설치 됐을때 사용한다.

In-app Billing Response

구글플레이 어플리케이션은 동기 / 비동기적으로 요청에 응답한다. 다음 3가지의 번들 키는 동기화된 응답이다.

RESPONSE_CODE

이 키는 상태정보와 에러정보를 제공한다.

PURCHASE_INTENT

이 키는 체크아웃 액티비티를 실행하는 PendingIntent를 제공한다.

REQUEST_ID

이 키는 어떤 요청에 대한 응답인지를 확인하기 위한 요청아이디를 제공한다.

각각 broadcast를 통해 돌아오는 비동기 응답 키들은 다음과 같다.

Com.android.vending.billing.RESPONSE_CODE

- 이 응답은 구글플레이 서버의 응답코드를 포함하고 있고 인앱빌링 요청을 보낸 후 보내진다. 서버의 응답코드는 빌링 요청이 성공적으로 구글플레이(서버)로 보내진 것을 나타내거나, 빌링요청 중에 에러가 발생했음을 나타낼 수 있다.

Com.android.vending.billing.IN_APP_NOTIFY

- 이 응답은 구매가 상태를 변경했음을 나타낸다. 구매성공, 취소, 환불등. 한개이상의 notification ID를 갖고 있고 이것은 서버 사이드의 메시지와 일치하며, 각각의 메시지는 하나이상의 트랜잭션에 대한 정보를 보유하고있다. 이 메시지를 받은 후 어플리케이션에서는 notification ID를 포함한 GET_PURCHASE_INFORMATION 를 보내 자세한 정보를 얻을 수 있다.

Com.android.vending.billing.PURCHASE_STATE_CHANGED

- 이 응답은 하나 이상의 트랜잭션에 대한 자세한 정보를 포함한다. JSON 형태. 이 JSON은 서명되어 보내진다. 보안차원에서, 이 서명을 검증하는 것이 좋다. 다음과 같은 형태로 리턴된다.

```
{ "nonce" : 1836535032137741465,
  "orders" :
  [{ "notificationId" : "android.test.purchased",
    "orderId" : "transactionId.android.test.purchased",
    "packageName" : "com.example.dungeons",
    "productId" : "android.test.purchased",
    "developerPayload" : "bGoa+V7g/yqDXvKRqq+JTFn4uQZbPiQJo4pf9RzJ",
    "purchaseTime" : 1290114783411,
    "purchaseState" : 0,
    "purchaseToken" : "rojeslcdyyiapnqcyknkjyyjh" }]
}
```

Messaging sequence

전형적인 구매 요청의 시퀀스는 figure2와 같다. `sendBillingRequest()` 메서드는 bold 로 표시되어있고 broadcast intents는 italic으로 표시되어있다. 가독성을 위해 모든 리퀘스트마다 전달되는 `RESPONSE_CODE` broadcast intents는 표시되지 않았다.

클라이언트의 메시지, 구글플레이의 메시지

- 개발자의 어플리케이션에서 상품아이디와 다른 파라미터를 포함한 구매요청을 보냄. (`REQUEST_PURCHASE`)
- 구글플레이 어플리케이션이 `RESPONSE_CODE (구매요청)`, `PURCHASE_INTENT`, `REQUEST_ID` 키를 보유한 번들을 개발자의 어플리케이션에 보냄. `PURCHASE_INTENT`는 주어진 상품 아이디로 체크아웃 UI의 액티비티를 실행시키는 Pending Intent가 들어있다.
- 개발자의 어플리케이션에서 2의 pendingIntent를 실행시킴. (어플리케이션 컨텍스트가 아닌 액티비티 컨텍스트로 실행시켜야함)
- 체크아웃 플로우가 완료 되면 (유저가 아이템을 사거나 취소하거나) 구글 플레이는 개발자의 어플리케이션으로 `IN_APP_NOTIFY`를 보냄. Notification ID를 포함하고 있고 이걸로 트랜잭션을 참조가능.
- 개발자의 어플리케이션에서 notification ID 를 포함, `GET_PURCHASE_INFORMATION` 을 보냄.
- 구글플레이 어플리케이션이 `RESPONSE_CODE(트랜잭션 정보)` 키와 `REQUEST_ID`키를 포함한 번들을 보냄.
- 구글플레이가 `PURCHASE_STATE_CHANGED` broadcast intent로 트랜잭션 정보를 개발자의 어플리케이션에 보냄.
- 트랜잭션 정보를 개발자의 어플리케이션에서 받았음을 `CONFIRM_NOTIFICATIONS`를 통해 알림.
- 구글플레이 어플리케이션이 `RESPONSE_CODE(트랜잭션 정보확인)` 를 보냄.

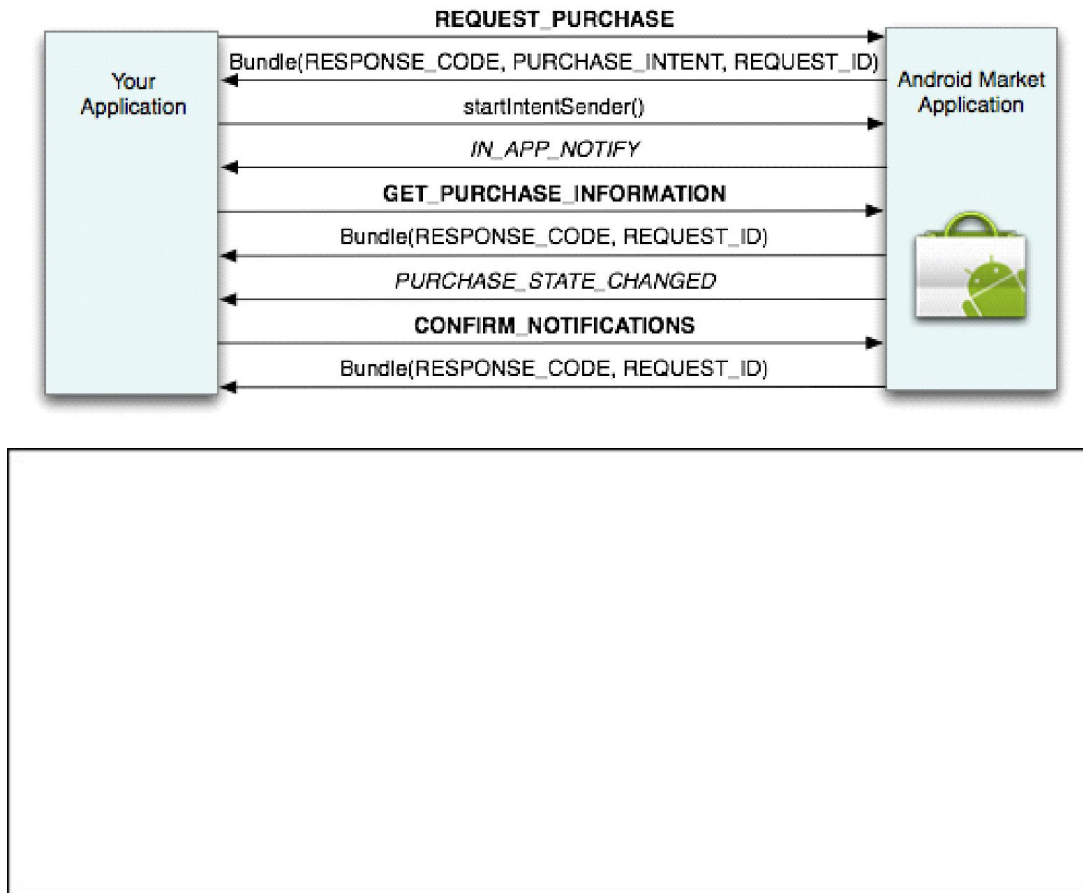


Figure 2. Message sequence for a purchase request.

8번 과정을 빼먹으면 구글플레이가 IN_APP_NOTIFY 를 계속 전송하니 꼭 빼먹지말자. 좋은 방법은, 유저에게 아이템이 전달되기 전까지 CONFIRM_NOTIFICATION을 보내지 말자. 앱이 크래쉬로 죽는다거나해서 사용자가 상품을 전달받기 전에 중단된다면 다음에 다시 IN_APP_NOTIFY를 받아서 사용자에게 상품을 전달하면 된다. 또, 좋은 인앱빌링 구현 어플리케이션은 IN_APP_NOTIFY가 다종의 주문을 포함하고 있는 메시지로 넘어올때도 처리할 수 있어야한다.

트랜잭션 보존에 대한 메시징 시퀀스는 Figure3에 나타난다. 요청은 3응답을 부른다 첫번째는 RESPONSE_CODE, REQUEST_ID이고 그 후, 구글플레이 어플리케이션이 RESPONSE_CODE 인텐트를 브로드캐스팅한다.

RESTORE_TRANSACTIONS 요청타입은 PURCHASE_STATE_CHANGED 브로드캐스트 인텐트를 부른다. (3번째) 이 인텐트는 구매 요청때 보내줬던 트랜잭션 정보와 같은 형태의 정보를 포함하고 있다. 하지만 CONFIRM_NOTIFICATIONS 메시지를 보낼 필요는 없다.

NOTE : RESTORE_TRANSACTIONS은 처음 디바이스에 어플리케이션을 설치하거나 어플리케이션이 지워져서 다시 깔았을때만 호출해야한다.

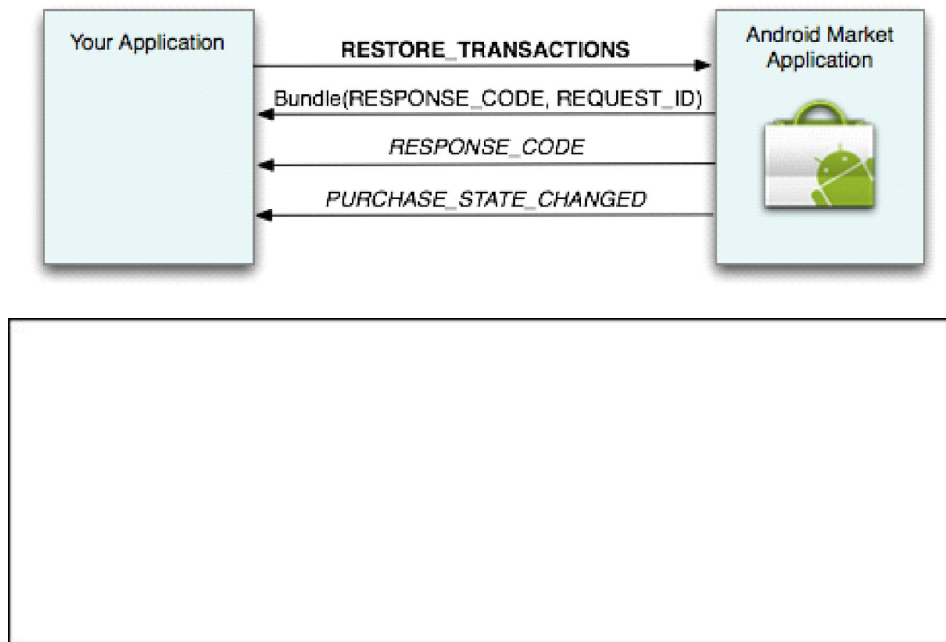


Figure 3. Message sequence for a restore transactions request.

인앱빌링이 지원되는지를 체크하는 메시지 시퀀스는 Figure 4에 나타난다. CHECK_BILLING_SUPPORTED 요청에 대한 동기화된 응답이 서버의 응답코드를 포함한 번들을 제공한다. RESULT_OK 는 지원된다는 뜻이다; 반대로 RESULT_BILLING_UNAVAILABLE은 어플리케이션의API버전이 인지하지 못하거나 유저가 인앱 빌링을 이용할수 없음을 나타낸다(예를 들면, 인앱빌링이 허가되지 않은 나라에 살고 있을 수 있다.) SERVER_ERROR 는 구글플레이 서버에 문제가 있음을 나타낸다.

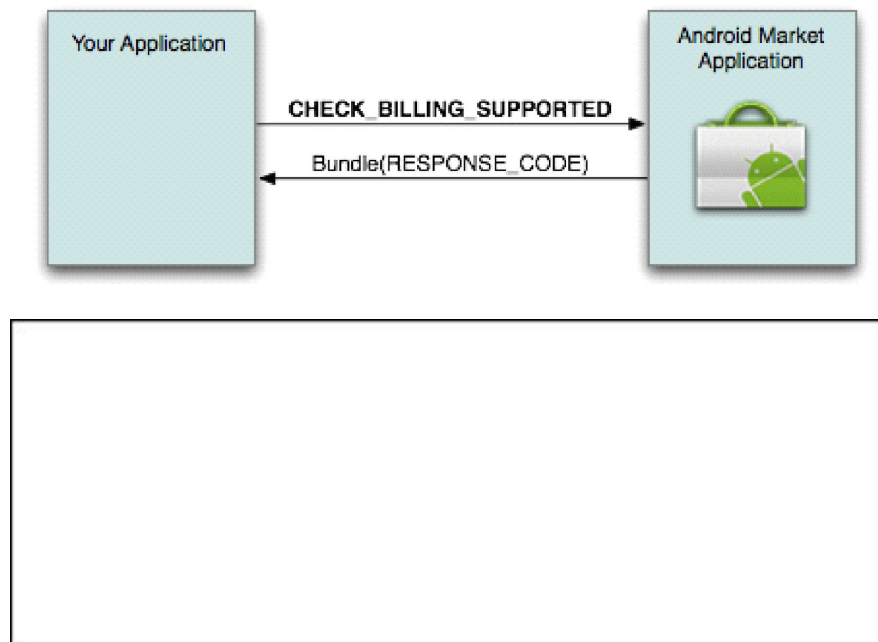


Figure 4. Message sequence for checking whether in-app billing is supported.

Handling IN_APP_NOTIFY messages

보통, 개발자의 어플리케이션은 REQUEST_PURCHASE의 응답으로 IN_APP_NOTIFY를 받는다. 구매 상태가 변경되었음을 의미하고, 이 상세 정보를 얻기 위해 GET_PURCHASE_INFORMATION 리퀘스트를 당신의 어플리케이션에서 보낼것이다. 이에 대한응답으로 PURCHASE_STATE_CHANGED로 브로드캐스트가 올것이며, 이를 확인했음을 CONFIRM_NOTIFICATION으로 당신의 어플리케이션에서 응답할 것이다.

어떤 경우에, 이미 구매 정보를 받았음에도 IN_APP_NOTIFY 메시지를 받을수도 있고, 반대로 구매하지도 않았는데 이 메시지를 받을 수도 있을 것이다. 이런 경우의 처리도 필요하다.

Handling multiple IN_APP_NOTIFY messages

구글플레이가 CONFIRM_NOTIFICATIONS 메시지를 보냈던 PURCHASE_STATE_CHANGED 메시지에 대해 받을때, 보통 IN_APP_NOTIFY를 보내는것을 멈춘다. 그러나, 네트워크 커넥션이 불안정하다든가의 이유로 CONFIRM_NOTIFICATIONS가 전달되지 않아 계속 보낼지도 모른다. 그러므로 당신의 어플리케이션은 이를 처리해야하며, 이는 JSON 에 포함된 orderId 를 이용하면 된다. 모든 트랜잭션은 유니크한 orderId를 가지기 때문이다.

Handling refunds and other unsolicited IN_APP_NOTIFY messages

보내지도 않았는데 이 메시지를 받는 경우는 두가지가 있다.

첫번째는, 두 디바이스 이상에서 당신의 어플리케이션을 설치하고 어느 한쪽에서 구매를 실행하는 경우다. 이 경우에는 REQUEST_PURCHASE를 실행한 것과 동일하므로 해당메시지가 전송된다. (구매타입이 managed per user account 일때만)

두번째는, 구글 월렛으로부터 환불 요청을 받았을때다. 환불정보는 PURCHASE_STATE_CHANGED의 JSON에 포함되어있다. 이때 purchaseState 필드는 2이다.

Security Controls

트랜잭션의 보안상의 무결성을 위해 PURCHASE_STATE_CHANGED의 JSON string을 서명해서 보낸다. 이 서명은 당신의 출시계정 (publisher account)와 연관해서 private Key를 이용해서 만든 것이다. Publisher site에서 RSA 키 페어를 생성하는데 당신은 여기서 퍼블릭키를 얻을 필요가 있다. (프로필페이지에서 얻을 수 있다)

이 퍼블릭키로 서명을 검증하라.

인앱빌링은 또한 nonces (한번만 사용되는 난수) 를 사용한다. 당신의 어플리케이션은 하나의 nonce 를 생성해서 GET_PURCHASE_INFORMATION 요청과 RESOTRE_TRANSACTIONS에 실어 보내는게 좋다. 구글플레이가 그 요청을 받을때, JSON string에 그 nonce를 실어 보낼 것이다.
그럼 **서명과 함께 nonce 를 이중으로 검사**할 수 있다.

In-app Billing Requirements and Limitations

다음 요구사항과 제한사항을 확인하라.

- 구글플레이를 통해 출시된 어플리케이션에서만 이용하세요!
- 구글월렛상인(Google wallet Merchant) 계정을 갖고 있어야 해요.
- 2.3.4 이상의 구글플레이앱만이 가능해요. Subscriptions는 3.5 버전이상의 구글플레이 앱이 필요해요. 안드로이드 3.0 버전에서는 5.0.12 이상이 필요해요 .
- 안드로이드 1.6 이상이어야 해요.
- 디지털 콘텐츠만 인앱빌링으로 판매가 가능해요.
- 구글플레이는 구매한 콘텐츠를 유저에게 배달하는 작업은 관여하지 않아요.
- 네트워크가 연결되어있어야해요.