



Native App Android 개발 가이드

Version 2.0

Android

Android Studio 에서 편집을 기준으로 알아봅니다.

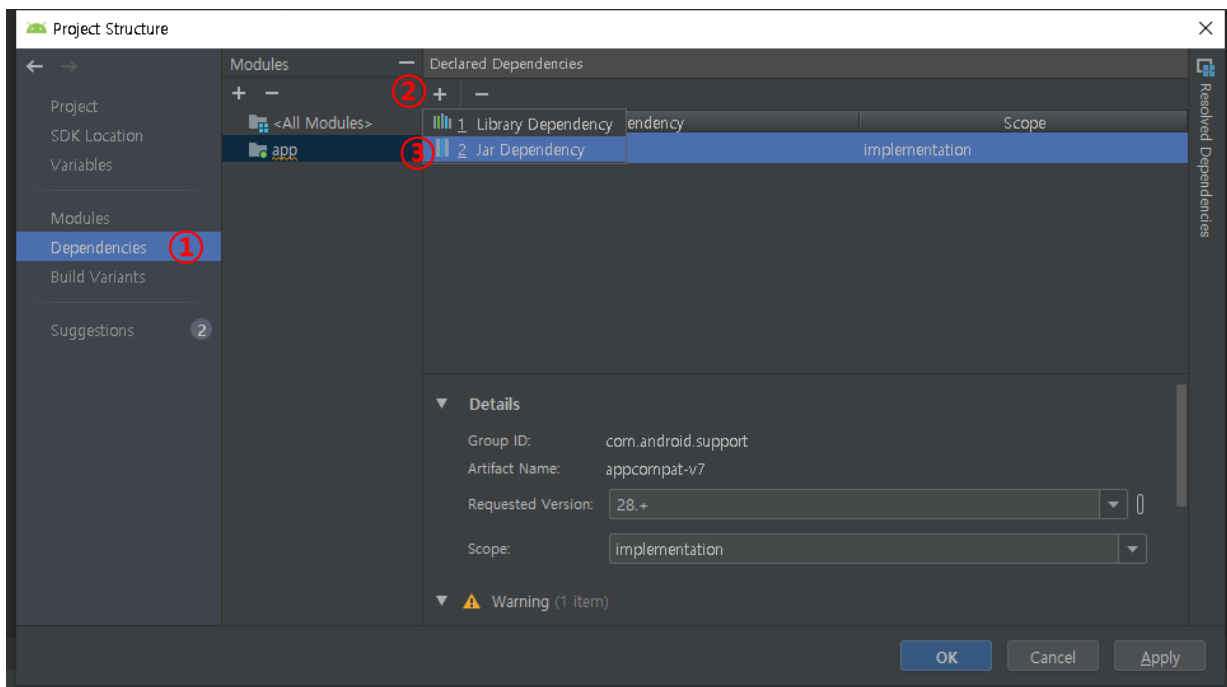
네파넬 API 적용 시 [64 페이지의 고급 설정](#) 방법을 참고하여 디버그 모드로 설정 후 진행하세요.

라이브러리 복사

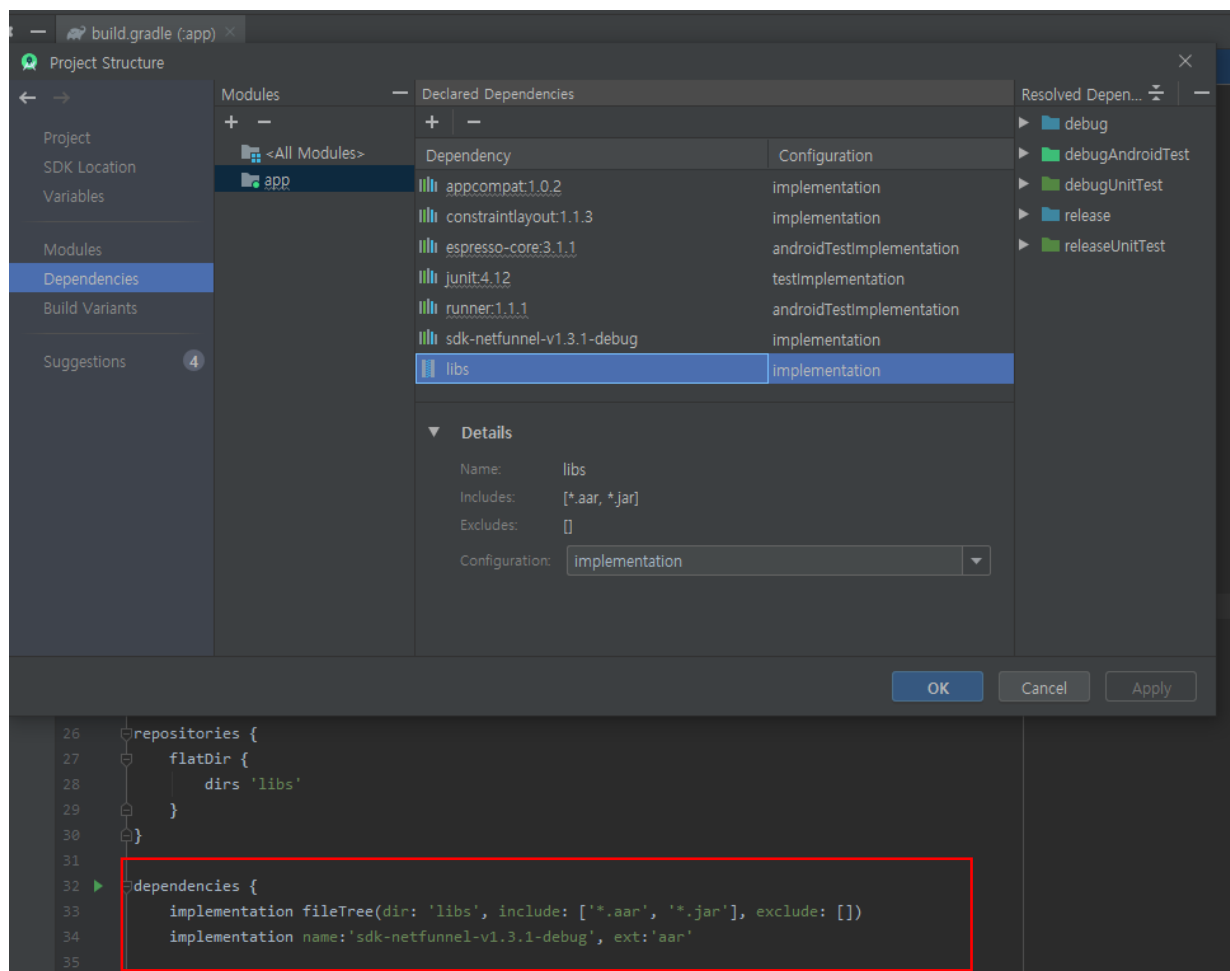
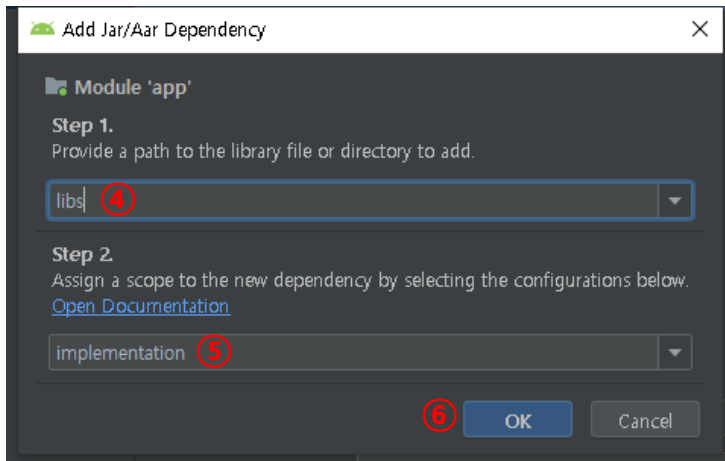
jar 파일 복사

- 원본 (version 은 상이할 수 있음)
 - Release/배포 버전(log 를 남기지 않음): /libs/sdk-netfunnel-v1.6.0.aar
 - Debug/테스트 버전(log 남김): /libs/sdk-netfunnel-v1.6.0-debug.aar
- 복사 위치
 - project/app/libs/~.aar

라이브러리 모듈 등록



<번호 순서대로 라이브러리 등록 >



<build.gradle 적용 내용 확인 >

전역 변수 파일 복사

- 원본
libs/values/netfunnel_strings.xml
libs/values/netfunnel_styles.xml
- 복사 위치
project/app/src/main/res/values/

dialog_netfunnel.xml 복사

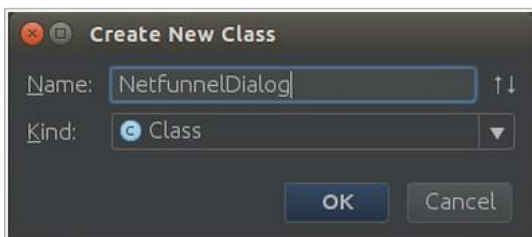
- 원본
libs/layout/dailog_netfunnel.xml
- 복사 위치
project/app/src/main/res/layout/

drawable 폴더 복사

- 원본
libs/drawable
- 복사 위치
project/app/src/main/res/

dialog 추가

[project > app > src > main > java] 마우스 오른쪽 [new > java class] 선택한 후 아래 그림처럼 추가하세요.



추가 후에 libs/java/NetfunnelDialog.java 파일을 열어서 상단의 import 부분부터 복사해서 붙여 넣으세요.

환경 설정

불필요한 drawable 삭제

- drawable-hdpi
- drawable-mdpi
- drawable-xhdpi
- drawable-xxhdpi

화면 회전, 리사이즈 시 동작하도록 추가

■ 회전

- AndroidManifest.xml 에 아래 내용을 추가하세요.

```
01. <uses-permission android:name="android.permission.INTERNET" />
```

■ 리사이즈

AndroidManifest.xml 에 아래 내용을 추가하세요.

```
01. <activity>
02.     android:configChanges="orientation|screenSize"
03.     ...
04. </activity>
```

Apache HTTP 클라이언트 사용

■ Apache HTTP 클라이언트 계속 사용 (Android 9/API 28 이상)

Android 9 이상을 타겟팅하는 앱이 Apache HTTP 클라이언트를 계속 사용하려면

AndroidManifest.xml 에 아래 내용을 추가하세요.

```
01. <application
02.     <uses-library android:name="org.apache.http.legacy" android:required="false" />
03.     ...
04. </application>
```

■ HTTPS 가 아닌 HTTP 를 사용할 경우

AndroidManifest.xml 에 아래 내용을 추가하세요.

```
01. <application
02.     android:usesCleartextTraffic="true"
03.     ...
04. </application>
```

API 적용

Native App 적용

전용 애플리케이션에서(Native App) 적용 방법을 알려줍니다.

■ 설정 가능 값 (import com.netfunnel.api.Property)

```
01. Property property =Property.getDefaultInstance();
02. property.setProtocol("https");           // protocol (Required)
    property.setHost("nf2.netfunnel.co.kr"); // host (Required)
    property.setPort(443);                   // port (Required)
    property.setTimeout(3); //넷퍼넬 서버 장애 여부를 판단하는 기준 값 (단위:초)
    property.setRetry(1); //넷퍼넬 서버에서 응답이 없을 경우 retry하는 횟수 (단위:횟수)
```

Request 전달

■ 요청 Type

» BEGIN (Required)

넷퍼넬 서버에 자원 사용 허가를 받는 요청입니다. BEGIN 요청 시 순번을 부여받게 되며 자신의 순번에 도달하지 않았다면, 사용자 화면에 대기창을 출력하여 대기를 시켜줍니다.

- Service ID 와 Action ID 를 지정하여 요청하는 방법

```
Netfunnel.BEGIN(String service_id, String action_id
    , Netfunnel.Listener firstListener) { ... } // listener/handler 방식 모두 지원
```

```
01. Netfunnel.BEGIN("service_1","act_1"
    ,NetfunnelDialog.Show(this)
    ,new Netfunnel.Listener(){
        @Override
        public void netfunnelMessage(Netfunnel netfunnel, Netfunnel.EvnetCode code) {
```

- Service ID 와 Action ID 를 지정하여 요청하는 방법, nid 포함

```
Netfunnel.BEGIN(String service_id, String action_id, Integer inNodeID
    , Netfunnel.Listener firstListener) { ... } // listener/handler 방식 모두 지원
```

```
01. Netfunnel.BEGIN("service_1","act_1", 1
    ,NetfunnelDialog.Show(this)
    ,new Netfunnel.Listener(){
        @Override
        public void netfunnelMessage(Netfunnel netfunnel, Netfunnel.EvnetCode code) {
```

*. nid(inNodeID)는 Integer type 으로 (1~9)사이의 숫자만 사용 가능(병렬처리 구분자)

action에서 nid 지정 시 Complete/Alive에서도 동일한 nid 값 지정 필수

» END (Required)

점유했던 자원을 반환하는 요청으로 BEGIN 을 실행한 후에 자원을 모두 사용한 이후에는 꼭 실행해줘야 합니다. 만약 실행하지 않는다면 불필요한 대기가 발생할 수 있습니다.

- 기본적인 END(자원 반납) 요청 방법

```
01. Netfunnel.END();
```

- END(자원 반납) 요청 방법, nid 포함 (parameter 에 nid 등록, action 의 nid 와 같게)

```
01. Netfunnel.END( 1 );
```

» ALIVE (Optional)

자원을 점유한 이후에 일정 시간 동안 계속 자원을 점유하고자 할 때는 ALIVE 요청을 해야 합니다. (하나의 transaction 제어로 끝나는 것이 아닌, 여러 단계에 걸쳐 제어하고자 하는 경우에 사용)

- 기본적인 ALIVE(자원 유지) 요청 방법

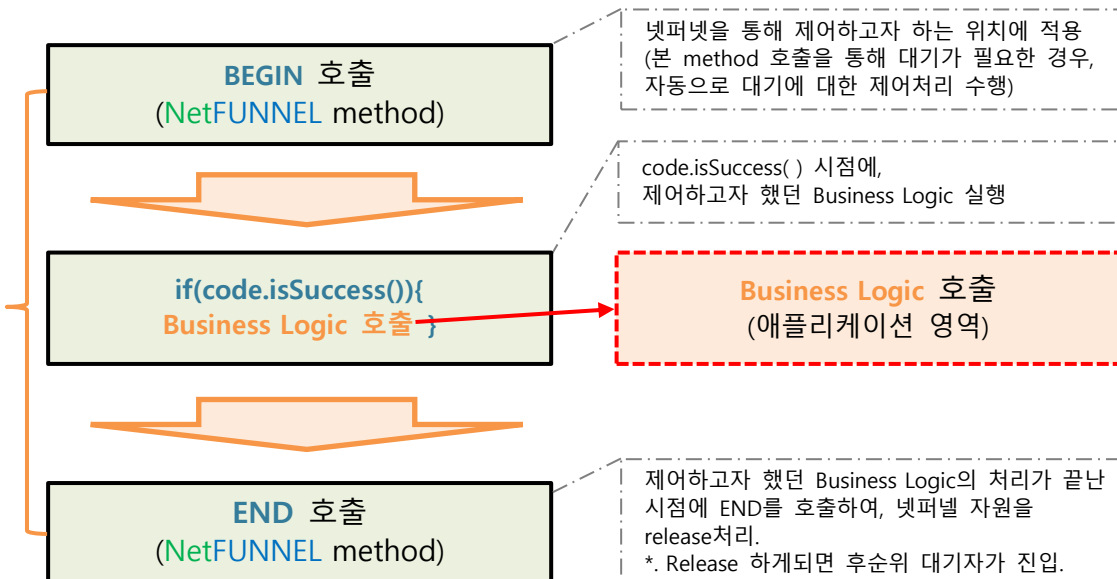
```
01. Netfunnel.ALIVE();
```

- ALIVE(자원 유지) 요청 방법, nid 포함 (parameter 에 nid 등록, action 의 nid 와 같게)

```
01. Netfunnel.ALIVE( 1 );
```

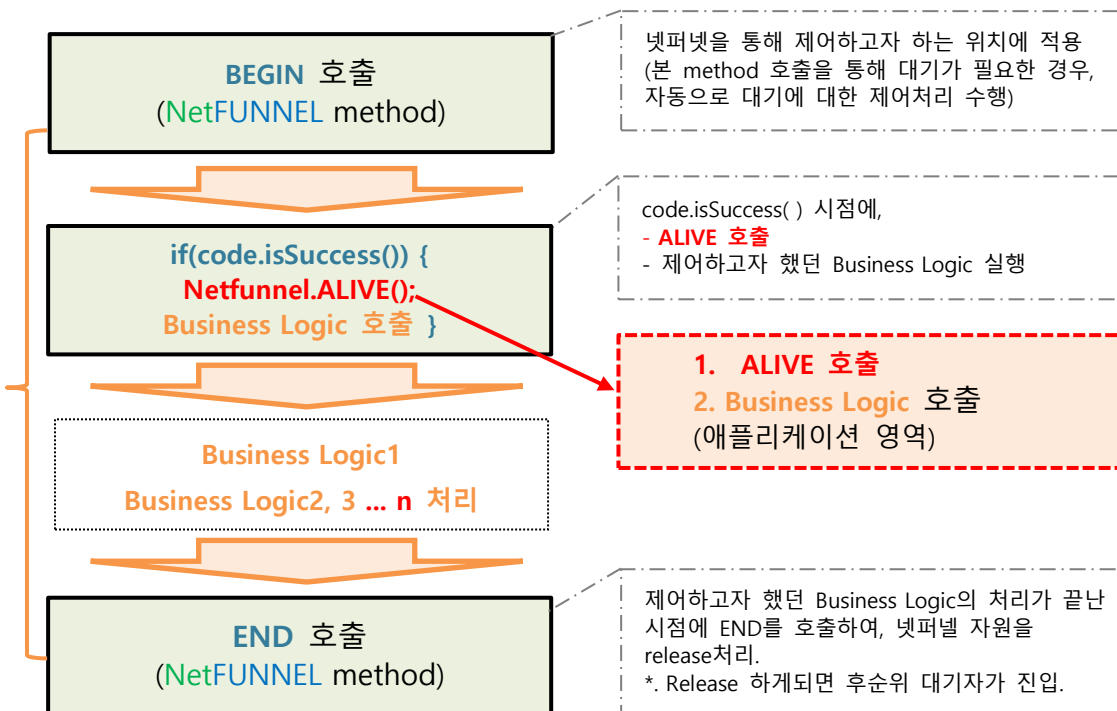
단일 구간 적용 방식

제어하고자 하는 Business Logic 을 넷퍼널에서 제공하는 method(BEGIN, END)로 감싸는 형태로 적용



여러 구간을 하나로 묶어서 적용하는 방식

제어하고자 하는 Business Logic 을 넷퍼널에서 제공하는 method(BEGIN, END)로 감싸고, code.isSuccess() 시점에 ALIVE 를 호출하여 이후 여러 구간을 지나는 동안 timeout 에 빠지지 않게 구성하는 형태



handler 적용

전용 애플리케이션에서(Native App) **handler** 방식의 적용 방법을 알려줍니다.

■ 설정: 넷퍼널 서버와 연동할 수 있도록 기본 정보를 설정해야 한다

```

01. @Override
02. protected void onCreate(Bundle savedInstanceState) {
03.     super.onCreate(savedInstanceState);
04.     setContentView(R.layout.activity_my);
05.
06.     Property property = Property.getDefaultInstance();
07.     // 넷퍼널 서버 기본 설정을 변경한다. (필수)
08.     property.setProtocol("https");           // 넷퍼널 서버 protocol (http / https)
09.     property.setHost("nf2.netfunnel.co.kr"); // 넷퍼널 서버 host (ip / domain)
10.     property.setPort(443);                   // 넷퍼널 서버 port (80 / 443)
11.     property.setTimeout(3);                 // 넷퍼널 서버 장애 여부를 판단하는 기준 값 (단위:초)
12.     property.setRetry(1);                   // 넷퍼널 서버에서 응답이 없을 경우 retry하는 횟수 (단위:횟수)
13. }

```

■ 적용:

```

01. public void btnLibApiHandler_Click(View view) {
02.     // 서비스 ID, 액션 ID를 지정한다.
03.     Netfunnel.BEGIN("service_1","act_1"
04.         ,NetfunnelDialog.Show(this)
05.         ,new Handler(){
06.             public void handleMessage(Message msg) {
07.                 try {
08.                     Bundle data = msg.getData();
09.                     Netfunnel.EvnetCode code = Netfunnel.EvnetCode.toEnum(msg.what);
10.                     Netfunnel netfunnel = (Netfunnel)msg.obj;
11.                     if(code.isContinue()){
12.                         ContinueData continue_data = netfunnel.getContinueData();
13.                         if (code == Netfunnel.EvnetCode.ContinueInterval) {
14.                             } else { // Netfunnel.EvnetCode.Continue
15.                                 String continue_msg = " 예상대기시간:" + continue_data.getCurrentWaitTimeSecond() + "초\n"
16.                                     + " 진 행 율:" + continue_data.getCurrentWaitPercent() + "%\n"
17.                                     + " 대기자수(앞):" + continue_data.getCurrentWaitCount() + "명\n"
18.                                     + " 대기자수(뒤):" + continue_data.getCurrentNextCount() + "명\n"
19.                                     + " 초당처리량(TPS):" + netfunnel.getResponse().getTPS() + "\n"
20.                                     + " 확인주기(TTL):" + netfunnel.getResponse().getTTL() + "초\n"
21.                                     + " notice account:" + continue_data.getAccountNotice() + "\n"
22.                                     + " update account:" + continue_data.getUpdateAccount() + "\n"
23.                                     + "UI 표시대기시간:" + netfunnel.getProperty().getUiWaitTimeLimit() + "초\n"
24.                                     + "UI 표시대기자수(앞):" + netfunnel.getProperty().getUiWaitCountLimit() + "명\n"

```

```

25.         + "UI 표시대기자수(뒤):" + netfunnel.getProperty().getUiNextCountLimit() + "명";
26.
27.         Log.d("netfunnel", "-----\n" + continue_msg);
28.     }
29.
30.     return;
31. }
32.
33. NetfunnelDialog.Close();
34.
35. if(code.isSuccess()){
36.     // TODO: 자체처리영역
37.     // Business Logic 호출 위치
38.
39.     if(code == Netfunnel.EvnetCode.Success) {
40.         // 주의) 여기에만 AliveNotice()을 사용할수 있는 곳
41.     }else if(code == Netfunnel.EvnetCode.NotUsed){
42.     }else if(code == Netfunnel.EvnetCode.Bypass){
43.     }else if(code == Netfunnel.EvnetCode.ErrorBypass){
44.     }else if(code == Netfunnel.EvnetCode.ExpressNumber){
45.     }
46.
47. }else if(code.isBlocking()){
48.     // TODO: 자체처리영역
49.
50.     if(code == Netfunnel.EvnetCode.Block) {
51.         // 서비스 차단
52.     }else if(code == Netfunnel.EvnetCode.IpBlock){
53.         // 사용자 사용패턴에 의한 차단
54.     }
55.
56. // 정지 버튼 눌렀을때 이벤트 처리
57. }else if(code.isStop()){
58.     Toast.makeText(getApplicationContext(), "onUserStop:"+code, Toast.LENGTH_LONG).show();
59.     // TODO: 자체처리영역
60. }else{ // }else if(code.isError()){
61.     Toast.makeText(getApplicationContext(), "onError:"+code, Toast.LENGTH_LONG).show();
62.     // TODO: 자체처리영역
63. }
64.
65.     Netfunnel.END();
66.
67. }catch (Exception e){
68.     e.printStackTrace();
69. }
70.
71. }
72.
73. });
74.
75. }

```

대기창 없이 background에서 제어하고자 하는 경우

!. 특수한 경우에만 사용하시길 바랍니다. (권고하지 않음)

NetfunnelDialog.Show(this) 대신 ==> NetfunnelDialog.isShow(this, false) 사용하여,
대기창 없이 제어할 수 있습니다.

- 대기창을 표시하지 않는 적용 방법

■ MainActivity

```
01. public void btnLibApiHandler_Click(View view) {
02.     // 서비스 ID, 액션 ID를 지정한다.
03.     Netfunnel.BEGIN("service_1","act_1"
04.     ,NetfunnelDialog.isShow(this, false)
05.     ,new Handler(){
06.         public void handleMessage(Message msg) {
```

■ NetfunnelDialog.java

isShow() function 구현 (샘플 프로젝트 파일 참조)

```
01. public static NetfunnelDialog isShow(Context context, boolean isShow) {
02.     NetfunnelDialog dialog = NetfunnelDialog.getInstance(context);
03.     dialog.show();
04.     if(!isShow) {
05.         dialog.hide();
06.     }
07.     return dialog;
08. }
```

- 대기가 발생하는 상황에서(background) 대기 제어를 종료하고자 하는 경우에
netfunnel.StopContinue(); 호출을 통해 제어를 중지할 수 있습니다.

■ MainActivity

```
01. public void btnLibApiHandler_Click(View view) {
02.     // 서비스 ID, 액션 ID를 지정한다.
03.     Netfunnel.BEGIN("service_1","act_1"
04.     ,NetfunnelDialog.isShow(this, false)
05.     ,new Handler(){
06.         public void handleMessage(Message msg) {
07.             try {
08.                 Bundle data = msg.getData();
09.                 Netfunnel.EvnetCode code = Netfunnel.EvnetCode.toEnum(msg.what);
10.                 Netfunnel netfunnel = (Netfunnel)msg.obj;
```

```

        if (code.isContinue()){
            netfunnel.StopContinue();
        }
    }

```

Listener 적용

전용 애플리케이션에서(Native App) **listener** 방식의 적용 방법을 알려줍니다.

■ 설정: 넷퍼널 서버와 연동할 수 있도록 기본 정보를 설정해야 한다

```

01. @Override
02. protected void onCreate(Bundle savedInstanceState) {
03.     super.onCreate(savedInstanceState);
04.     setContentView(R.layout.activity_my);
05.
06.     Property property =Property.getDefaultInstance();
07.     // 넷퍼널 서버 기본 설정을 변경한다. (필수)
08.     property.setProtocol("https");           // 넷퍼널 서버 protocol (http / https)
09.     property.setHost("nf2.netfunnel.co.kr");  // 넷퍼널 서버 host (ip / domain)
10.     property.setPort(443);                   // 넷퍼널 서버 port (80 / 443)
11.     property.setTimeout(3);                  // 넷퍼널 서버 장애 여부를 판단하는 기준 값 (단위:초)
12.     property.setRetry(1);                    // 넷퍼널 서버에서 응답이 없을 경우 retry하는 횟수 (단위:횟수)
13. }

```

■ 적용:

```

01. public void btnLibApiListener_Click(View view) {
02.     // 서비스 ID, 액션 ID를 지정한다.
03.     Netfunnel.BEGIN("service_1","5"
04.         ,NetfunnelDialog.Show(this)
05.         ,new Netfunnel.Listener(){
06.             @Override
07.             public void netfunnelMessage(Netfunnel netfunnel, Netfunnel.EvnetCode code) {
08.                 try {
09.                     Log.d("netfunnel", "netfunnelMessage -> code: " + code + " / "+netfunnel.getResponse().getCode() + " ttl=" +
netfunnel.getResponse().getTTL());
10.                     if (code.isContinue()) {
11.                         ContinueData continue_data = netfunnel.getContinueData();
12.                         if (code == Netfunnel.EvnetCode.ContinueInterval) {
13.                             } else { // Netfunnel.EvnetCode.Continue
14.                                 String continue_msg = " 예상대기시간:" + continue_data.getCurrentWaitTimeSecond() + "초\n"
15.                                     + " 진 행 율:" + continue_data.getCurrentWaitPercent() + "%\n"
16.                                     + " 대기자수(앞):" + continue_data.getCurrentWaitCount() + "명\n"
17.                                     + " 대기자수(뒤):" + continue_data.getCurrentNextCount() + "명\n"
18.                                     + " 초당처리량(TPS):" + netfunnel.getResponse().getTPS() + "\n"
19.                                     + " 확인주기(TTL):" + netfunnel.getResponse().getTTL() + "초\n"
20.                                     + " notice acount:" + continue_data.getAccountNotice() + "\n"
21.                                     + " update acount:" + continue_data.getUpdateAccount() + "\n"

```

```

22.         + "UI 표시대기시간:" + netfunnel.getProperty().getUiWaitTimeLimit() + "초\n"
23.         + "UI 표시대기자수(앞):" + netfunnel.getProperty().getUiWaitCountLimit() + "명\n"
24.         + "UI 표시대기자수(뒤):" + netfunnel.getProperty().getUiNextCountLimit() + "명";
25.         Log.d("netfunnel", "-----\n" + continue_msg);
26.     }
27.     return;
28. }
29.     NetfunnelDialog.Close();
30.     if (code.isSuccess()) {
31.         // TODO: 자체처리영역
32.         // Business Logic 호출 위치
33.         if(code == Netfunnel.EvnetCode.Success) {
34.             // 주의) 여기에만 AliveNotice()을 사용할수 있는 곳
35.         }else if(code == Netfunnel.EvnetCode.NotUsed){
36.         }else if(code == Netfunnel.EvnetCode.Bypass){
37.         }else if(code == Netfunnel.EvnetCode.ErrorBypass){
38.         }else if(code == Netfunnel.EvnetCode.ExpressNumber){
39.         } else if (code.isBlocking()) {
40.             // TODO: 자체처리영역
41.             if(code == Netfunnel.EvnetCode.Block) {
42.                 // 서비스 차단
43.             }else if(code == Netfunnel.EvnetCode.IpBlock){
44.                 // 사용자 사용패턴에 의한 차단
45.             }
46.         } else if (code.isStop()) {
47.             // TODO: 자체처리영역
48.         } else { // }else if(code.isError()){
49.             // TODO: 자체처리영역
50.         }
51.         Netfunnel.END();
52.     } catch (Exception e)
53.     { e.printStackTrace();
54.     }
55. }
56. });
57. }

```

대기창 없이 background에서 제어만 희망하는 경우

! 특수한 경우에만 사용하시길 바랍니다. (권고하지 않음)

NetfunnelDialog.Show(this) 대신 ==> NetfunnelDialog.isShow(this, false) 사용하여, 대기창 없이 제어할 수 있습니다.

- 대기창을 표시하지 않는 적용 방법

■ MainActivity

```
01. public void btnLibApiListener_Click(View view) {
02.     // 서비스 ID, 액션 ID를 지정한다.
03.     Netfunnel.BEGIN("service_1","act_1"
04.         ,NetfunnelDialog.isShow(this, false)
05.         , new Netfunnel.Listener() {
06.     public void netfunnelMessage(Netfunnel netfunnel, Netfunnel.EvnetCode code) {
```

■ NetfunnelDialog.java

isShow() function 구현 (샘플 프로젝트 파일 참조)

```
01. public static NetfunnelDialog isShow(Context context, boolean isShow) {
02.     NetfunnelDialog dialog = NetfunnelDialog.getInstance(context);
03.     dialog.show();
04.     if(!isShow) {
05.         dialog.hide();
06.     }
07.     return dialog;
08. }
```

- 대기가 발생하는 상황에서 대기창도 표시하지 않고, 더 이상 넷퍼널 제어를 받지 않고자 하는 경우에 netfunnel.StopContinue(); 호출을 통해 제어를 멈출 수 있습니다.

■ MainActivity

```
01. public void btnLibApiListener_Click(View view) {
02.     // 서비스 ID, 액션 ID를 지정한다.
03.     Netfunnel.BEGIN("service_1","act_1"
04.         ,NetfunnelDialog.isShow(this, false)
05.         ,new Netfunnel.Listener() {
06.         @Override
07.         public void netfunnelMessage(Netfunnel netfunnel, Netfunnel.EvnetCode code) {
08.             try {
09.                 if (code.isContinue()){
10.                     netfunnel.StopContinue();
11.                 ...
12.             }
```



(주)에스티씨랩

서울시 강남구 논현로 419 PMK빌딩 8층