

안드로이드 하이브리드 앱 개발

CONNECT / 장영남

목차

1. 하이브리드 앱?
2. Android WebView
3. JavaScriptInterface
4. 하이브리드 앱 개발 기술
5. 간단한 실습

하이브리드 앱

하이브리드 앱

- 네이티브 앱과 웹앱이 결합된 형태
- 네이티브 앱처럼 설치하지만 실제로는 웹앱으로 구성
- JavaScript, HTML 및 CSS로 작성
- Webview로 실행 됨

Native app/web app/hybrid app

Native app

- Native app

- 일반적인 모바일 어플리케이션
- 현재 서비스 되고 있는 대부분이 Native app
- 모바일 운영체제에서 제공하는 모든 기능을 이용할 수 있음

Native app

- 장점

- 웹앱, 하이브리드앱에 비하여 성능이 높음.

- 네이티브API를 호출하여 사용함으로 플랫폼과 밀착되어 있음

- 개발언어에 익숙한 개발자라면 좀 더 쉽게 접근할 수 있음

Native app

- 단점

- 플랫폼에 한정적.
- 해당 플랫폼에서 요구하는 언어에 제약적
- 해당 언어와 플랫폼의 API를 다루는데 익숙해야 함
- 기능 추가 및 변경시 앱 업데이트가 필요함

web app

- web app

- 웹 브라우저를 통해 액세스되는 응용 프로그램
- 사실상 모바일 웹

web app

- 장점

- 웹사이트를 보는 것이기 때문에 따로 앱 설치가 필요 없음.
- 모든 기기와 브라우저에서 접근할 수 있음
(단말/OS의 특성이 없음)
- 유지보수가 용이함

web app

- 단점

- 플랫폼API(카메라 등)를 사용할 수 없음
- 네이티브 앱 보다 성능이 떨어짐
- 복잡한 기능을 구성하기 어려움

Hybrid app

- 네이티브앱과 웹앱이 결합된 형태
- 네이티브앱에 웹뷰를 띄워 웹앱을 실행시키는 방식

Hybrid app

- 장점

- 네이티브API와 브라우저API를 이용한 다양한 개발 가능
- 웹개발 기술을 사용해 앱을 개발할 수 있음
- 한 번의 개발로 여러 플랫폼에 대응할 수 있음
- 개발비용이 적게 듦

Hybrid app

- 단점

- 추가적인 기능구현을 위해 네이티브 개발 지식 필요
- 웹뷰의 설정에 따라 앱의 성능을 좌우함
- UI프레임워크 도구를 사용하지 않는 경우 개발자가 UI를 제작해야함

Hybrid app

- 네이티브앱과 웹앱이 결합된 형태
- 네이티브앱에 웹뷰를 띄워 웹앱을 실행시키는 방식
- 웹개발 기술을 사용해 앱을 개발할 수 있음
- 한 번의 개발로 여러 플랫폼에 대응할 수 있음
- 웹 개발 기술 + 앱 개발 기술

NAVER

연예 스포츠 웹툰·툼 차·테크 쇼핑 동영상

북미 '강대강' 대치...8월 한반도 긴장 더욱 높아져



北리용호-中왕이, 양자회담 개최...안보리제재 논의 주목

다주택자 추가대출 때 LTV·DTI 10%p ↓...전국도 일괄 조정

한미외교장관 회담...틸러슨 "안보리 결의, 좋은 결과"

폭염 계속...중부지방 국지적 강한비 | 기상특보

실시간급상승 5 이수근

세종대로에서 물놀이 한 판

국방과학연구소 창설 47주년 '무기 281종 국산화'

28°C 수원시 화서1동 어제보다 6° 낮음

https://m.naver.com

NAVER

뉴스 연예 스포츠 웹툰·툼 차·테크 쇼핑 동영상

북미 '강대강' 대치...8월 한반도 긴장 더욱 높아져

北리용호-中왕이, 양자회담 개최...안보리제재 논의 주목

다주택자 추가대출 때 LTV·DTI 10%p ↓...전국도 일괄 조정

한미외교장관 회담...틸러슨 "안보리 결의, 좋은 결과"

폭염 계속...중부지방 국지적 강한비 | 기상특보

실시간급상승 1 홍진영




세종대로에서 물놀이 한 판

국방과학연구소 창설 47주년 '무기 281종 국산화'

Android webview

webView?

WebView

added in API level 1

Summary: Nested Classes | Inherited XML Attrs |
Constants | Inherited Constants | Inherited Fields | Ctors
| Methods | Protected Methods | Inherited Methods |
[Expand All]

public class WebView

extends [AbsoluteLayout](#) implements [ViewTreeObserver.OnGlobalFocusChangeListener](#), [ViewGroup.OnHierarchyChangeListener](#)

[java.lang.Object](#)

↳ [android.view.View](#)

↳ [android.view.ViewGroup](#)

↳ [android.widget.AbsoluteLayout](#)

↳ [android.webkit.WebView](#)

A View that displays web pages. This class is the basis upon which you can roll your own web browser or simply display some online content within your Activity. It uses the WebKit rendering engine to display web pages and includes methods to navigate forward and backward through a history, zoom in and out, perform text searches and more.

<https://developer.android.com/reference/android/webkit/WebView.html>

WebView?

- Webkit 기반으로 웹 페이지를 렌더링 하여 제공하는 view 의 종류
- Web브라우저 형태를 구성하거나 Activity에 웹 콘텐츠 제공 가능
- 페이지 네비게이션 가능

WebKit

- 웹 브라우저를 만드는 데 기반을 제공하는 오픈 소스 응용 프로그램 프레임워크
- 원래는 사파리 웹 브라우저 엔진으로 사용하기 위해 권커러 브라우저의 KHTML 소프트웨어 라이브러리에서 가져온 것
- 옴니웹, 시이라, 아로라, 미도리, 유티클, icab, 어도비 통합 런타임, 휴대 전화(아이폰 포함), 노키아의 Series 60 브라우저, 구글의 안드로이드 플랫폼에 사용되고 있음.

<https://webkit.org>

<https://ko.wikipedia.org/wiki/웹킷>

사용방법

- AndroidManifest.xml

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

- layout.xml

```
<WebView  
    android:id="@+id/webview"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"/>
```

- MainActivity.java

```
webView = (WebView) findViewById(R.id.webview);  
webView.loadUrl("https://m.naver.com");
```



WebViewTest



NAVER



N



뉴스 연예 스포츠 웹툰·툼 푸드 쇼핑 등

왕이 "사드결정 양국관계에 찬물"...강경화 "소통으로 풀어야"

한미·북중 연쇄 외교장관 회담...한미압박에 北반발·中설득

부동산 시장 안정 되찾나? 앞으로 1~2주가 관건

'세기의 재판' 이재용 구형 D-1...범죄 혐의 입증 됐나

열대야 폭염 계속...월요일도 찜통 더위 | 기상특보

실시간급상승 4 여건육



대북제재 결의 2371호는 북한을 어느 정도 압박할까



주차장 알바 "더위보다 짜증내는 손님이 더 힘들죠"

webView

- goBack();
- goForward();
- reload();
- canGoBack();
- canGoForward();

- clearCache(true);
- stopLoading();

WebSetting

- WebView 의 설정상태를 관리
- WebView에 종속적
- WebSetting 항목
 - setSaveFormData();
 - setJavaScriptEnabled();
 - setJavaScriptCanOpenWindowsAutomtically();
 - setSupportZoom();
 - setSupportMultipleWindows();
 - setBuiltInZoomControls();
 - setLoadsImagesAutomatically();
 - setUseWideViewPort();
 - setCacheMode();
 - setPluginState();

WebViewclient

- WebView에서 발생하는 이벤트를 처리 하기 위한 class
- webView.setWebViewclient() 메소드로 webView 에 WebViewclient 구현체를 적용
- onPageStarted()
- onLoadResource()
- onPageFinished()
- onReceivedError()
- onUnhandledKeyEvent()
- shouldOverrideUrlLoading()

webchromecient

- 브라우저 속성과 관련된 항목을 제어하는 class
- `webView.setWebchromecient()` 메소드로 webView에 webchromecient 구현체를 적용

- `void onCloseWindow()`
- `boolean onConsoleMessage()`
- `void onConsoleMessage()`
- `boolean onCreateWindow()`
- `void onGeolocationPermissionsHidePrompt()`
- `void onGeolocationPermissionsShowPrompt()`
- `boolean onJsAlert()`
- `boolean onJsConfirm()`
- `boolean onJsPrompt()`
- `void onProgressChanged()`
- `void onReceivedIcon()`
- `void onReceivedTitle()`
- `boolean onShowFileChooser()`

JavaScriptInterface

JavaScriptInterface

- `webView.addJavaScriptInterface()` 메소드를 통해 JavaScript method 가 구현된 객체를 webView에 주입
- 웹페이지에서 JavaScript를 호출하여 Native 메소드를 호출

addJavaScriptInterface

added in [API level 1](#)

```
void addJavaScriptInterface (Object object,  
                             String name)
```

Injects the supplied Java object into this WebView. The object is injected into the JavaScript context of the main frame, using the supplied name. This allows the Java object's methods to be accessed from JavaScript. For applications targeted to API level [JELLY_BEAN_MR1](#) and above, only public methods that are annotated with [JavaScriptInterface](#) can be accessed from JavaScript. For applications targeted to API level [JELLY_BEAN](#) or below, all public methods (including the inherited ones) can be accessed, see the important security note below for implications.

Note that injected objects will not appear in JavaScript until the page is next (re)loaded. For example:

```
class JsObject {  
    @JavaScriptInterface  
    public String toString() { return "injectedObject"; }  
}  
webView.addJavaScriptInterface(new JsObject(), "injectedObject");  
webView.loadData("", "text/html", null);  
webView.loadUrl("javascript:alert(injectedObject.toString())");
```


하이브리드 앱 개발기술

PhoneGap / cordova

- 니토비(Nitobi)가 만들고 어도비 시스템즈가 인수한 모바일 개발 프레임워크
- 자바스크립트, HTML5, CSS3를 이용하여 모바일 기기를 위한 응용 프로그램들을 만들 수 있게 함
- 결과물은 하이브리드 형태
- 1.9버전 이후 네이티브 코드와 하이브리드 코드 일부를 자유롭게 섞어서 사용 가능
- 어도비가 PhoneGap을 인수하여 Apache cordova로 이름을 바꾸고 오픈 소스로 공개 함

<http://docs.phonegap.com/getting-started/1-install-phonegap/desktop/>

<https://ko.wikipedia.org/wiki/폰갭>

<https://www.phonegap.com>

React Native

- Facebook에서 개발하고 공개한 모바일 프레임워크
- 웹 개발을 쉽게 하기 위해 만든 React에서 파생되어 Android, iOS를 개발하는 라이브러리로 확장 됨
- 자바스크립트, HTML5, CSS3를 이용하여 앱을 개발할 수 있음

<https://facebook.github.io/react-native/>

React Native

- 장점

- 자바스크립트로 빠르게 개발할 수 있고 서드파티 라이브러리 모듈을 이용할 수 있음
- UI를 수정하여도 다시 빌드할 필요 없이 화면을 바로 갱신하여 확인할 수 있음(라이브리딩)
- 카메라, GPS등의 네이티브 모듈을 이용할 수 있음

React Native

- 단점

- 모든 코드를 재사용할 수 없음
- 특정 Native 기능을 이용하기 위하여 플랫폼별 다른 API를 사용해야 함
- 계속 개발이 진행되기 때문에 버전에 따라 기능이 달라질 수 있음

ionic framework

- 하이브리드 모바일 앱 개발을 위한 완벽한 오픈 소스 SDK
- 2013 년에 출시되었으며 AngularJS 및 Apache cordova를 기반으로 제작
- CSS, HTML5 및 SASS와 같은 웹 기술을 사용
- cordova를 활용하여 하이브리드 형태로 배포 가능
- ionic creator를 이용하여 드래그앤드롭 방식으로 화면 구현 가능

Etc..

- AppPresso
- Sencha Touch
- OnsenUI

스
3
스

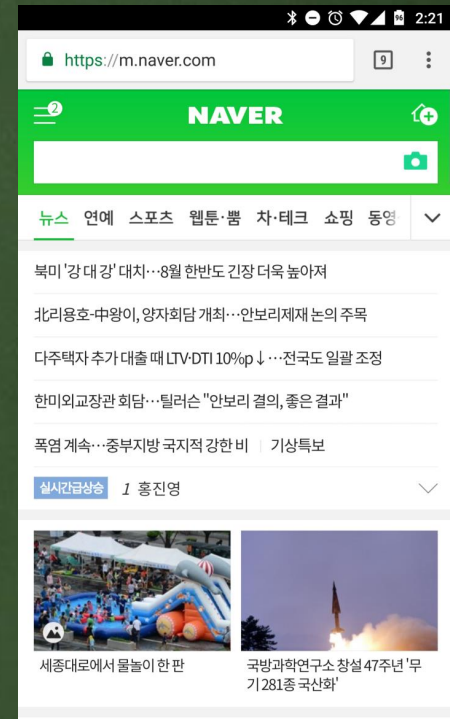
WebView Application 만들기

<https://goo.gl/BsAS8o>

<https://m.naver.com>

WebViewclient

- 페이지 이동 처리
 - Naver 페이지에서 페이지 이동을 하는 경우 브라우저가 실행 됨
 - 이 페이지 이동이 웹뷰 내에서 처리 되도록 수정할 것
 - 테스트 : <http://naver.com>
- Page start, finish 를 Toast 를 통해 noti 할 것
- Progress bar 추가



시험 페이지

<http://test.sweettracker.co.kr/boostcamp.html>

<https://goo.gl/gW2Gc2>



WebViewTest

App Link Scheme

[전화걸기](#)

[이메일](#)

Chrome Client

alert

confirm

prompt

Javascript Interface

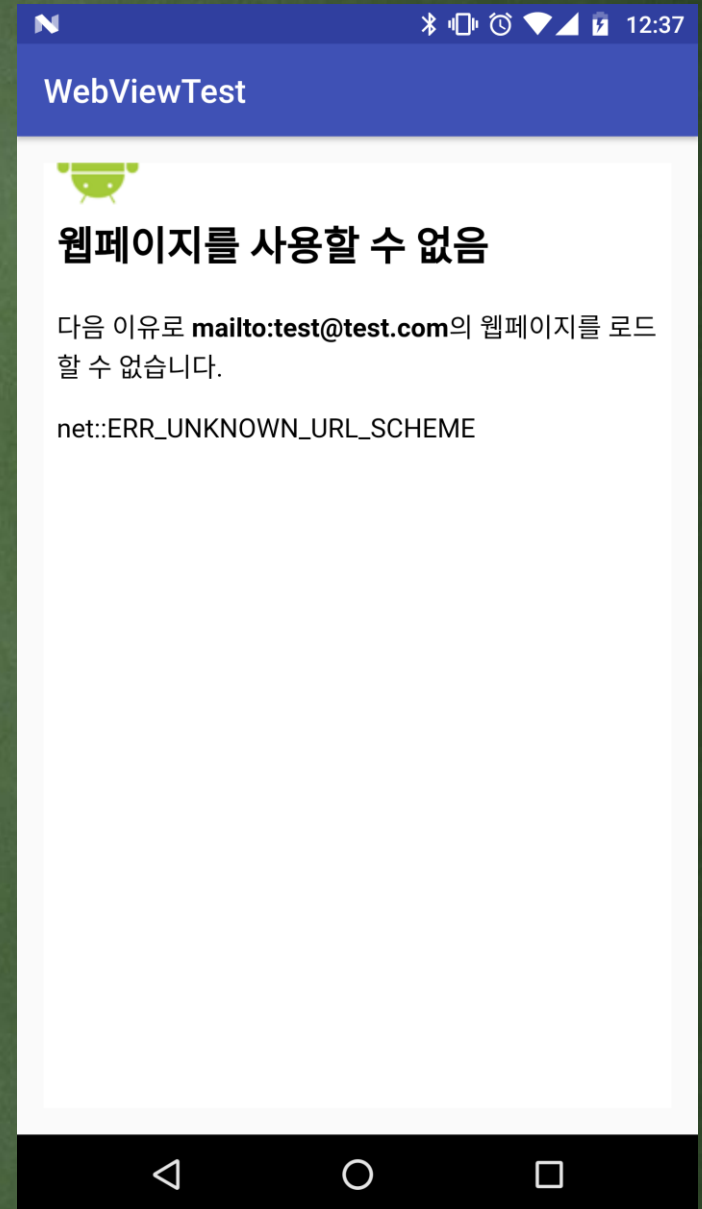
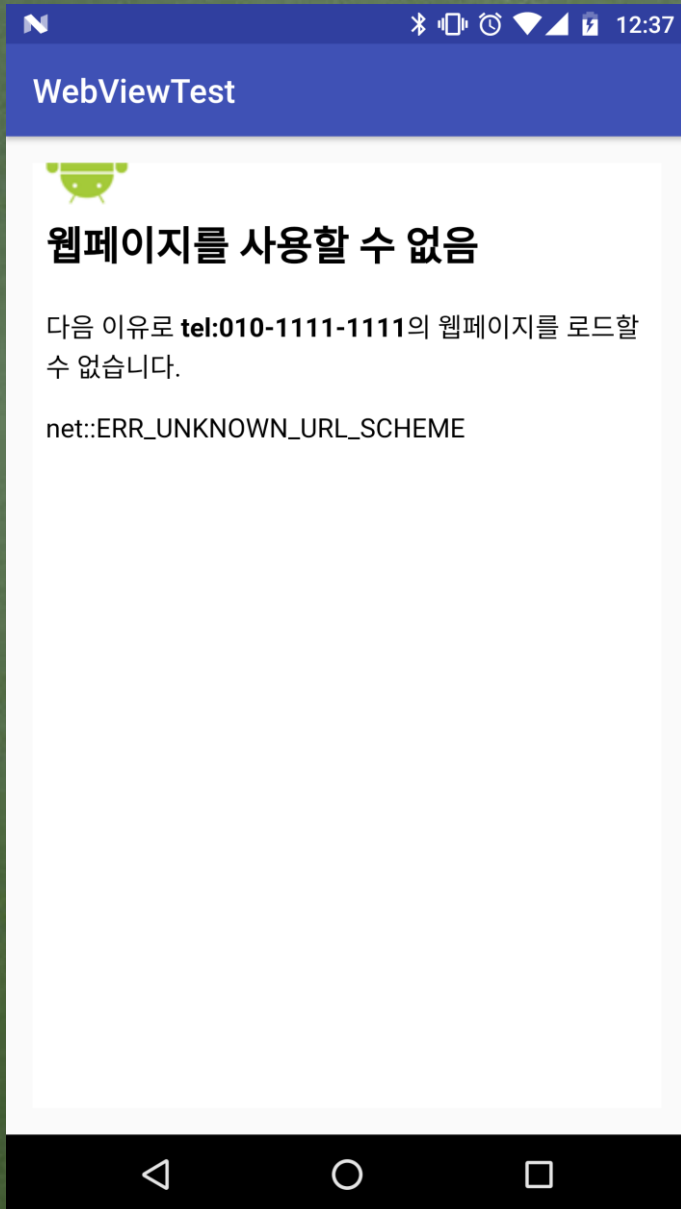
toast

send Message



web page - Native app link

- 전화걸기
 - tel:010-1111-1111
- 이메일
 - mailto:test@test.com



URI Scheme

<https://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml>

<https://goo.gl/sbSzRG>

URI Scheme

- file:
- ftp:
- http:
- https:
- mailto:
- news:
- telnet:
- market:

URI Scheme

- intent:
- naversearchapp:
- kakaotalk:
- kakaolink:
- fb:

URI Scheme

<https://developer.android.com/training/basics/intents/filters.html?hl=ko>

<https://goo.gl/PvsjS4>

```
<activity android:name="ShareActivity">
  <!-- filter for sending text; accepts SENDTO action with sms URI schemes -->
  <intent-filter>
    <action android:name="android.intent.action.SENDTO"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <data android:scheme="sms" />
    <data android:scheme="smsto" />
  </intent-filter>
  <!-- filter for sending text or images; accepts SEND action and text or image data -->
  <intent-filter>
    <action android:name="android.intent.action.SEND"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <data android:mimeType="image/*"/>
    <data android:mimeType="text/plain"/>
  </intent-filter>
</activity>
```

web page - Native app link

- 전화걸기

- Intent.ACTION_DIAL

```
Intent intent = new Intent(Intent.ACTION_DIAL, Uri.parse(url));  
startActivity(intent);
```

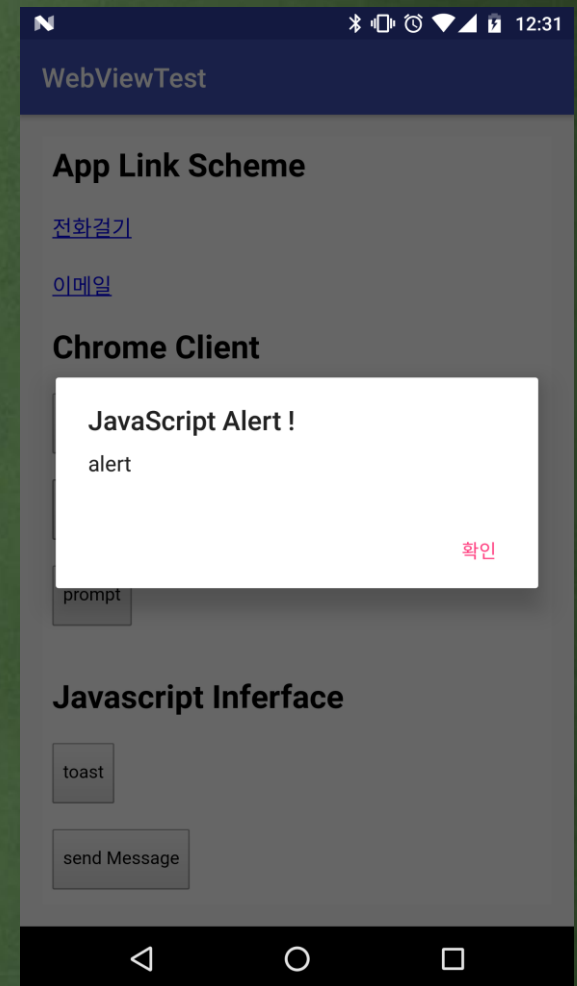
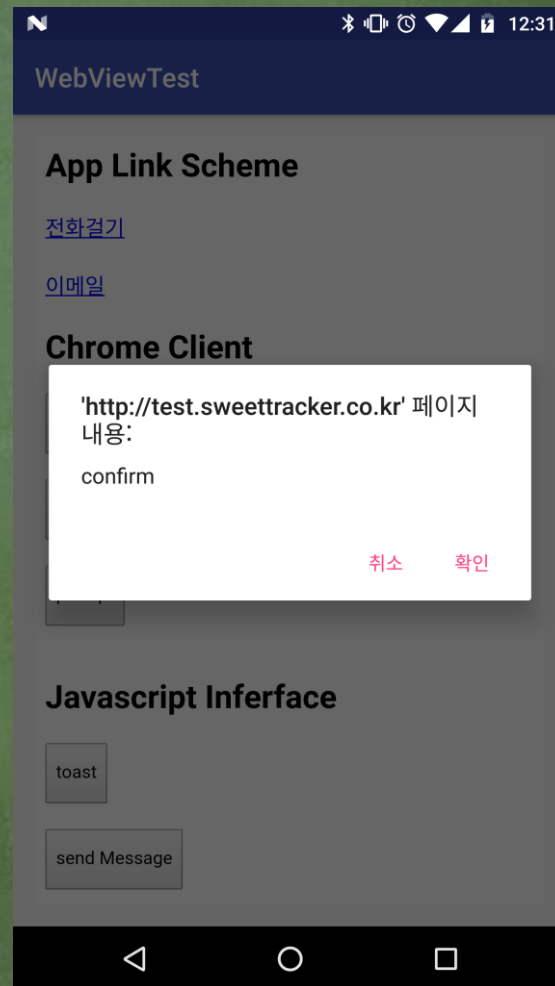
- 이메일보내기

- Intent.ACTION_SENDTO

```
Intent intent = new Intent(Intent.ACTION_SENDTO, Uri.parse(url));  
startActivity(intent);
```

webchromeclient

- alert
- confirm
- prompt



JavaScriptInterface

- `webView.addJavaScriptInterface()` 메소드를 통해 JavaScript method 가 구현된 객체를 webView에 주입
- 웹페이지에서 JavaScript를 호출하여 Native 메소드를 호출

addJavaScriptInterface

added in [API level 1](#)

```
void addJavaScriptInterface (Object object,  
                             String name)
```

Injects the supplied Java object into this WebView. The object is injected into the JavaScript context of the main frame, using the supplied name. This allows the Java object's methods to be accessed from JavaScript. For applications targeted to API level [JELLY_BEAN_MR1](#) and above, only public methods that are annotated with [JavaScriptInterface](#) can be accessed from JavaScript. For applications targeted to API level [JELLY_BEAN](#) or below, all public methods (including the inherited ones) can be accessed, see the important security note below for implications.

Note that injected objects will not appear in JavaScript until the page is next (re)loaded. For example:

```
class JsObject {  
    @JavaScriptInterface  
    public String toString() { return "injectedObject"; }  
}  
webView.addJavaScriptInterface(new JsObject(), "injectedObject");  
webView.loadData("", "text/html", null);  
webView.loadUrl("javascript:alert(injectedObject.toString())");
```

JavaScriptInterface

- Toast
 - 인자로 받은 string을 Toast로 보여줄 것
 - `boostcamp.showToast(toast);`
- sendMessage
 - 인자로 받은 string을 alert Dialog로 보여줄 것
 - `boostcamp.sendMessage(msg);`
- camera
 - 단말의 카메라 앱을 실행할 것
 - `boostcamp.camera();`