

헬로, 안드로이드

10주차 - 연결된 세상

강대기

동서대학교 컴퓨터정보공학부

학습 목표

- 인텐트로 다른 액티비티나 프로그램을 실행시킬 수 있다.
- 웹뷰를 통해 웹 화면을 액티비티 화면의 일부로 구성할 수 있다.
- 자바스크립트를 통해 안드로이드 프로그램을 호출하는 방법을 안다.
- 안드로이드 응용 프로그램에서 웹 서비스를 이용하는 방법을 안다.

차례

- 인텐트로 브라우징하기
- 뷰 안의 웹
- 자바스크립트에서 자바로, 자바에서 자바스크립트로
- 웹 서비스 이용하기
- 요약
- 퀴즈
- 연습문제

인텐트로 브라우징하기

- 휴대폰은 더 이상 통화 용도로만 사용되지 않음
- 안드로이드 – WebKit 을 통한 웹 브라우징과 TCP/IP 표준 소켓
- 네 개의 예제
 - 브라우저 인텐트 (browser intent)
 - 브라우저 뷰 (browser view)
 - 로컬 브라우저 (local browser)
 - 구글 번역 프로그램 (translate) – 데이터 바인딩과 스레딩 및 웹 서비스를 사용

브라우저 인텐트 (Browser Intent)

- 프로젝트 생성
 - `BrowserIntent`
 - `org.example.browserintent`
 - `BrowserIntent`
 - `BrowserIntent`
- `main.xml` 에서 `EditText`와 `Button` 위젯 지정
- `strings.xml` 에 필요한 스트링 지정
- `BrowserIntent` 의 `onCreate()` 메서드에서 `EditText`와 `Button` 위젯에 대한 listener 코드 안에서 `openBrowser()` 함수 호출
- `openBrowser()` 함수는 `Intent`로 안드로이드 안의 기본 브라우저 호출

main.xml

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="horizontal"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">
  <EditText
    android:id="@+id/url_field"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1.0"
    android:lines="1"
    android:inputType="textUri"
    android:imeOptions="actionGo" />
  <Button
    android:id="@+id/go_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/go_button" />
</LinearLayout>
```

strings.xml

```
<resources>  
  <string  
    name="app_name">BrowserIntent</string>  
  <string name="go_button">Go</string>  
</resources>
```

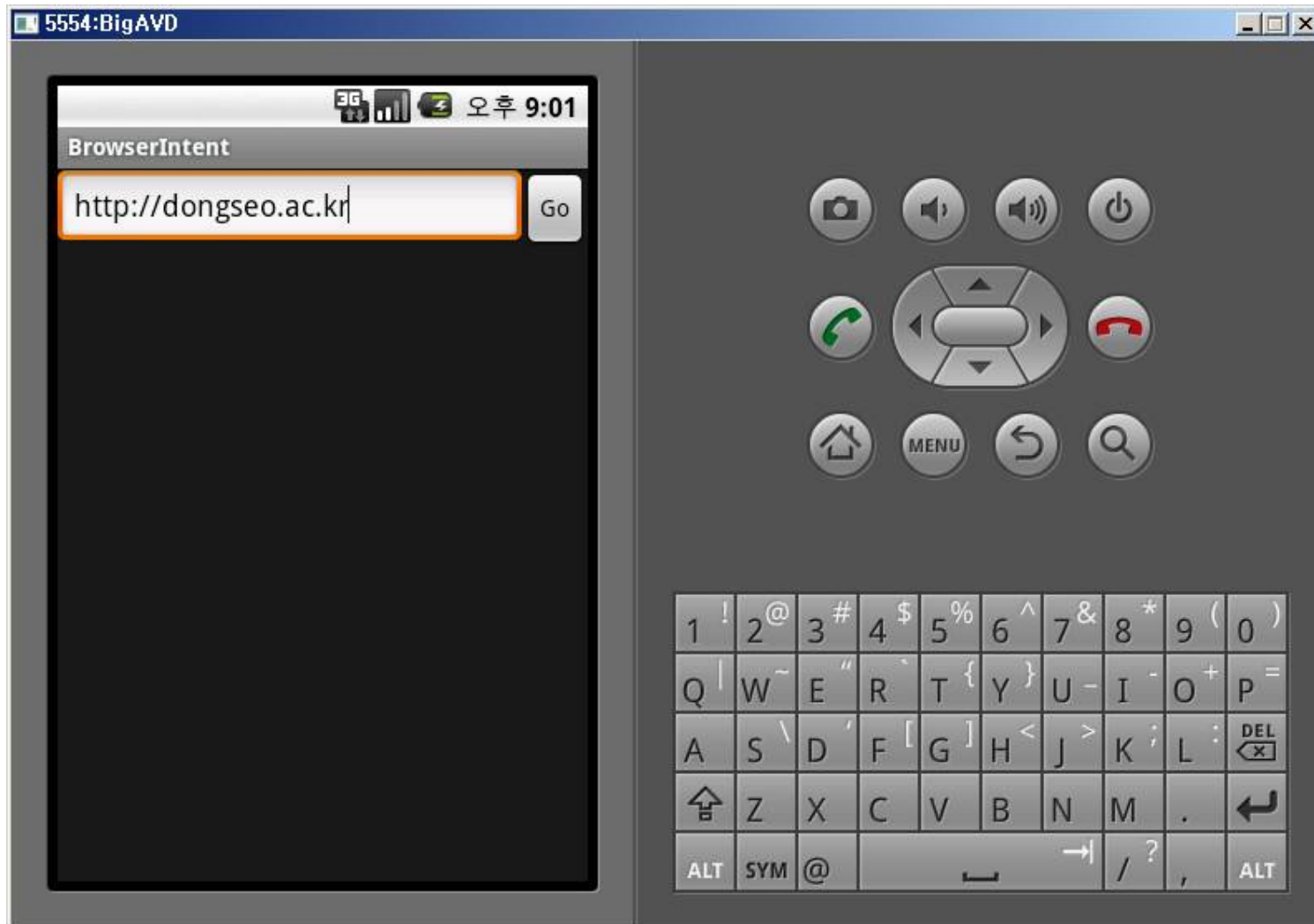
BrowserIntent.java

```
package org.example.browserintent;
...
public class BrowserIntent extends Activity {
    private EditText urlText;
    private Button goButton;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        // Get a handle to all user interface elements
        urlText = (EditText) findViewById(R.id.url_field);
        goButton = (Button) findViewById(R.id.go_button);
        // Setup event handlers
        goButton.setOnClickListener(new OnClickListener() {
            public void onClick(View view) {        openBrowser();    }
        });
        urlText.setOnKeyListener(new OnKeyListener() {
            public boolean onKey(View view, int keyCode, KeyEvent event) {
                if (keyCode == KeyEvent.KEYCODE_ENTER) {        openBrowser();        return true;    }
                return false;
            }
        });
    }

    /** Open a browser on the URL specified in the text box */
    private void openBrowser() {
        Uri uri = Uri.parse(urlText.getText().toString());
        Intent intent = new Intent(Intent.ACTION_VIEW, uri);
        startActivity(intent);
    }
}
```


브라우저 인텐트 실행 예



뷰 안의 웹 (WebView)

- 단순한 텍스트라도 웹브라우저로 디스플레이하는 게 좋음
- 프로젝트 생성
 - BrowserView
 - org.example.browserview
 - BrowserView
 - BrowserView
- main.xml 에는 WebView 추가
- BrowserView 의 onCreate() 메서드는 WebView 의 listener 코드 추가
- openBrowser() 함수는 WebView.loadUrl() 호출
- AndroidManifest.xml 에 퍼미션 추가

main.xml

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">
  <LinearLayout
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
    <EditText
      android:id="@+id/url_field"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_weight="1.0"
      android:lines="1" />
    <Button
      android:id="@+id/go_button"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/go_button" />
  </LinearLayout>
  <WebView
    android:id="@+id/web_view"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1.0" />
</LinearLayout>
```

BrowserView.java

```
package org.example.browserview;
...
public class BrowserView extends Activity {

    private EditText urlText;
    private Button goButton;
    private WebView webView;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        // Get a handle to all user interface elements
        urlText = (EditText) findViewById(R.id.url_field);
        goButton = (Button) findViewById(R.id.go_button);
        webView = (WebView) findViewById(R.id.web_view);
        // Setup event handlers
        goButton.setOnClickListener(new OnClickListener() {
            public void onClick(View view) {        openBrowser();    }
        });
        urlText.setOnKeyListener(new OnKeyListener() {
            public boolean onKey(View view, int keyCode, KeyEvent event) {
                if (keyCode == KeyEvent.KEYCODE_ENTER) {        openBrowser();        return true;    }
                return false;
            }
        });
    }
    /** Open a browser on the URL specified in the text box */
    private void openBrowser() {
        webView.loadUrl(urlText.getText().toString());
        webView.requestFocus();
    }
}
```

AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.example.browserview"
    android:versionCode="1"
    android:versionName="1.0.0">

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

    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
        <activity android:name=".BrowserView"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

뷰 안의 웹 실행 예



WebView 메서드 전체 목록

- `addJavascriptInterface()` - 자바 스크립트의 자바 객체 액세스 허용
- `createSnapshot()` - 현 페이지의 스크린샷 생성
- `getSettings()` - 설정을 조정하는 `WebSettings` 객체 반환
- `loadData()` - 브라우저에 주어진 문자열 데이터 로딩
- `loadDataWithBaseURL()` - 기준 URL을 사용해 주어진 데이터 로딩
- `loadUrl()` - 주어진 URL을 사용해 웹 페이지 로딩
- `setDownloadListener()` - 사용자가 .zip 이나 .apk 파일을 다운로드하는 것과 같은 다운로드 이벤트의 콜백 등록
- `setWebChromeClient()` - 제목 또는 진행줄 표시줄을 업데이트 하거나 자바스크립트 대화상자를 여는 등의, `WebView` 영역 밖에서 실행되는 이벤트의 콜백 등록
- `setWebViewClient()` - 리소스 로드하기, 키누르기, 인증 요청 등의 이벤트를 방해하도록 애플리케이션이 브라우저에 고리를 설정함
- `stopLoading()` - 현재 페이지 로딩 멈추기

자바스크립트와 자바(on 안드로이드)

- 자바스크립트를 통한 메서드 호출
 - 임베딩된 웹뷰를 통해 웹 페이지에서 안드로이드의 기능에 접근하는 것
 - WebView 클래스 안의 `addJavascriptInterface()` 메서드를 통해, 임베딩된 브라우저 내에서 DOM을 확장하고 자바스크립트가 접근할 새 객체를 정의할 수 있음 → 자바스크립트 코드가 객체에 메서드를 호출하면 안드로이드 프로그램 안의 메서드가 호출됨
- 역으로, 안드로이드 프로그램에서 자바스크립트 메서드 호출도 가능함
 - `loadUrl()`을 호출하고 `javascript:code-to-execute` 형식으로 URL을 전달함
 - 브라우저는 새 페이지가 아닌 현재 페이지에서 자바스크립트를 실행시킴
 - 본 방식은 ASP, ASP.NET 등의 프로그래밍에서도 그대로 사용됨
- 보안 문제를 신중히 고려해야 함

자바스크립트에서 자바 호출

- 프로젝트 생성
 - LocalBrowser / org.example.localbrowser / LocalBrowser / LocalBrowser
- main.xml
- index.html
 - res 가 아닌 asset 디렉토리 안에 들어감. 컴파일된 리소스가 아니기 때문임. asset 의 모든 파일들은 프로그램이 설치될 때, 로컬 스토리지에 그대로 복사됨
 - callJS() 함수를 통해 안드로이드 프로그램 호출
 - 첫번째 링크는 window.alert() 함수 부른 후 짧은 메시지 출력
 - 두번째 링크는 window.android 객체 위에 callAndroid() 메서드 호출
 - 여기서 window.android는 정의되어 있지 않으며, 애플리케이션이 브라우저를 임베딩해서 쓸 때, 객체를 직접 정의해 페이지가 사용할 수 있게 함

자바스크립트에서 자바 호출

- LocalBrowser.java

- Handler 를 통해 자바스크립트를 위한 브라우저용의 특별한 스레드와 안드로이드 사용자 인터페이스 호출을 위한 메인 스레드 간의 전환을 다룸
- AndroidBridge는 자바스크립트에 노출된 객체로 자바 코드를 호출하게 해 줌. 자바스크립트가 callAndroid()를 호출하면, 애플리케이션은 새로운 Runnable 객체를 만들어 Handler.post()를 통해 메인 스레드의 실행 중인 큐에 집어넣음. 나중에 메인 스레드는 기회가 되면 run() 메서드를 호출하고 이것은 setText()를 호출해 TextView 객체 내의 텍스트를 변경함
- onCreate() 메서드 안에서 자바스크립트를 활성화하고 AndroidBridge를 자바스크립트에 등록함.
- 임의의 WebChromeClient 객체를 만들어 setWebChromeClient() 메서드로 등록함. 브라우저가 window.alert()를 사용해 자바스크립트 경고를 여는 작업을 수행함. onJSAlert() 안에 Toast 클래스를 이용해서 3초간 보이는 메시지 창을 만듦
- 여기서 일반적으로 크롬은 브라우저 창 주위를 정리하는 모든 작업을 의미함
- loadUrl() 을 통해 로컬 웹 페이지를 로드함. file:/// 로 / 를 3개 사용함.
- 자바에서 자바스크립트로의 호출을 위해 버튼 클릭 리스너 설정. 버튼이 눌리면 onClick()이 호출되고, 이 메서드는 WebView.loadUrl()을 호출하여 브라우저에서 확인될 자바스크립트 코드가 전달됨. 이 코드는 index.html에 정의된 callJS() 함수 호출.

main.xml

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">
  <WebView android:id="@+id/web_view" android:layout_width="fill_parent"
    android:layout_height="fill_parent" android:layout_weight="1.0" />
  <LinearLayout android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent" android:layout_weight="1.0"
    android:padding="5sp">
    <TextView android:layout_width="fill_parent" android:layout_height="wrap_content"
      android:textSize="24sp" android:text="TextView" />
    <Button android:id="@+id/button"
      android:text="@string/call_javascript_from_android"
      android:layout_width="wrap_content" android:layout_height="wrap_content"
      android:textSize="18sp" />
    <TextView android:id="@+id/text_view" android:layout_width="fill_parent"
      android:layout_height="wrap_content" android:textSize="18sp" />
  </LinearLayout>
</LinearLayout>
```

index.html

```
<html>
<head>
<script language="JavaScript">
  function callJS(arg) {
    document.getElementById('replaceme').innerHTML = arg;
  }
</script>
</head>
<body>
<h1>WebView</h1>
<p>
<a href="#" onclick="window.alert('Alert from JavaScript')">
  Display JavaScript alert</a>
</p>
<p>
<a href="#" onclick="window.android.callAndroid('Hello from Browser')">
  Call Android from JavaScript</a>
</p>
<p id="replaceme">
</p>
</body>
</html>
```

LocalBrowser.java (1/2)

```
package org.example.localbrowser;
```

```
...
```

```
public class LocalBrowser extends Activity {
```

```
    private static final String TAG = "LocalBrowser";
```

```
    private final Handler handler = new Handler();
```

```
    private WebView webView;
```

```
    private TextView textView;
```

```
    private Button button;
```

```
    /** Object exposed to JavaScript */
```

```
    private class AndroidBridge {
```

```
        public void callAndroid(final String arg) { // must be final
```

```
            handler.post(new Runnable() {
```

```
                public void run() {
```

```
                    Log.d(TAG, "callAndroid(" + arg + ")");
```

```
                    textView.setText(arg);
```

```
                }
```

```
            });
```

```
        }
```

```
    }
```

LocalBrowser.java (2/2)

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    // Find the Android controls on the screen
    webView = (WebView) findViewById(R.id.web_view);
    textView = (TextView) findViewById(R.id.text_view);
    button = (Button) findViewById(R.id.button);
    // Rest of onCreate follows...

    // Turn on JavaScript in the embedded browser
    webView.getSettings().setJavaScriptEnabled(true);

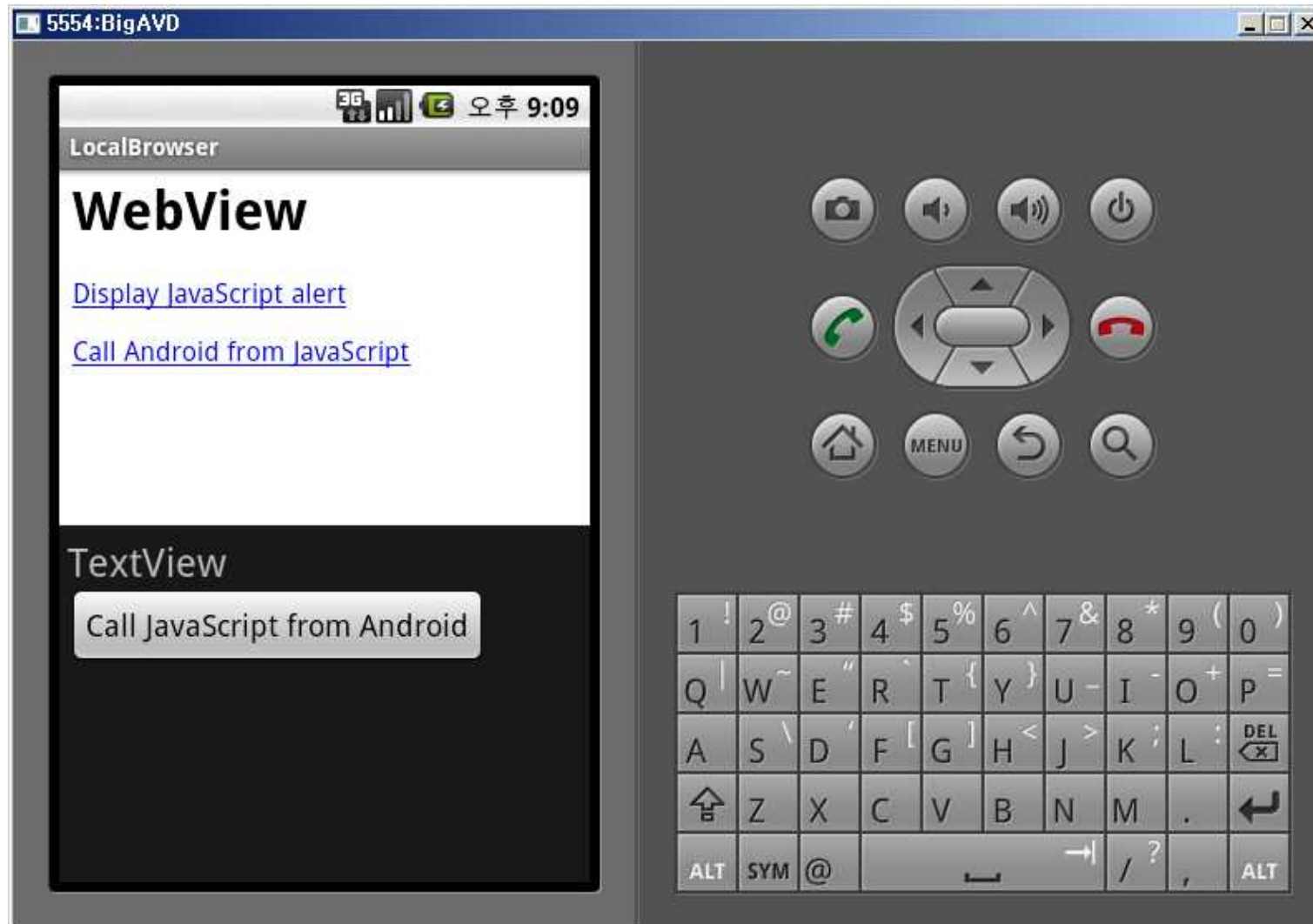
    // Expose a Java object to JavaScript in the browser
    webView.addJavaScriptInterface(new AndroidBridge(), "android");

    // Set up a function to be called when JavaScript tries
    // to open an alert window
    webView.setWebChromeClient(new WebChromeClient() {
        @Override
        public boolean onJsAlert(final WebView view, final String url, final String message,
            final String result) {
            Log.d(TAG, "onJsAlert(" + view + ", " + url + ", " + message + ", " + result + ")");
            Toast.makeText(LocalBrowser.this, message, 3000).show();
            result.confirm();
            return true; // I handled it
        }
    });

    // Load the web page from a local asset
    webView.loadUrl("file:///android_asset/index.html");

    // This function will be called when the user presses the
    // button on the Android side
    button.setOnClickListener(new OnClickListener() {
        public void onClick(View view) {
            Log.d(TAG, "onClick(" + view + ")");
            webView.loadUrl("javascript:callJS('Hello from Android')");
        }
    });
}
```

자바스크립트와 자바 실행 예



웹 서비스 이용하기

- Blocking I/O – 입출력 호출을 하고 결과를 기다림 (동기 호출)
- Non-Blocking I/O – 입출력 호출을 하고 결과가 나오는 동안, 프로그램 수행이 가능함 (비동기 호출)
- Non-Blocking I/O 를 다루기 위한 `java.util.concurrent` → Java 5에 통합됨
- 웹 서비스를 이용한 번역 프로그램
 - `Translate / org.example.translate / Translate / Translate`
- `AndroidManifest.xml` – 인터넷 사용 허가
- `main.xml` – `TableLayout` 사용
- `arrays.xml` – 스피너를 위한 스트링 배열
- `strings.xml`
- `TranslateTask.java` – 교재에서 설명이 생략되어 있음

Translate.java

- `findViews()` – 레이아웃 파일에 정의된 사용자 인터페이스 구성요소들을 연결함
- `setAdapters()` – 스피너의 데이터 소스 정의. Adapter 클래스는 데이터 소스와 사용자 컨트롤을 연결함.
- `setListeners()` – 사용자 인터페이스 처리기 설정
- `initThreading()` – 입력텍스트가 바뀔 때와 번역 언어가 변할 때 각각 호출되는 두 개의 리스너 정의.
 - 텍스트 변화는 1초, 언어 변화는 1/5 초
 - `TranslateTask` 인스턴스를 `ExecutorService` 클래스가 가지는 쓰레드 큐에 삽입함. `Future` 클래스 인스턴스가 반환됨.
 - `queueUpdate()` 호출 – `Handler`를 사용해 메인 스레드의 작업 목록에 지연된 업데이트 요청

지연된 요청과 쓰레드

- 매번 키 입력마다 요청하면 네트워크 트래픽 낭비인 반면, 사용자가 입력이 끝났음을 알리게 하는 건 사용자를 귀찮게 하고 AJAX와 같은 고수준 입출력이 아님.
- 따라서, 이를 처리하기 위한 지연되는 요청과 쓰레드는 효율적이면서도 효과적인 입력 처리를 위해 매우 중요함.

AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.example.translate"
    android:versionCode="1"
    android:versionName="1.0">

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

    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
        <activity android:name=".Translate"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-sdk android:minSdkVersion="3" />
</manifest>
```

main.xml

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent" android:layout_height="fill_parent">
    <TableLayout android:layout_width="fill_parent" android:layout_height="fill_parent"
        android:stretchColumns="1" android:padding="10dip">
        <TableRow>
            <TextView android:text="@string/from_text" />
            <Spinner android:id="@+id/from_language" />
        </TableRow>
        <EditText android:id="@+id/original_text" android:hint="@string/original_hint"
            android:padding="10dip" android:textSize="18sp" />
        <TableRow>
            <TextView android:text="@string/to_text" />
            <Spinner android:id="@+id/to_language" />
        </TableRow>
        <TextView android:id="@+id/translated_text" android:padding="10dip"
            android:textSize="18sp" />
        <TextView android:text="@string/back_text" />
        <TextView android:id="@+id/retranslated_text" android:padding="10dip"
            android:textSize="18sp" />
    </TableLayout>
</ScrollView>
```

arrays.xml

```
<resources>
  <array name="languages">
    <item>Bulgarian (bg)</item>
    <item>Chinese Simplified (zh-CN)</item>
    <item>Chinese Traditional (zh-TW)</item>
    <item>Catalan (ca)</item>
    <item>Croatian (hr)</item>
    <item>Czech (cs)</item>
    <item>Danish (da)</item>
    <item>Dutch (nl)</item>
    <item>English (en)</item>
    <item>Filipino (tl)</item>
    <item>Finnish (fi)</item>
    <item>French (fr)</item>
    <item>German (de)</item>
    <item>Greek (el)</item>
    <item>Indonesian (id)</item>
    <item>Italian (it)</item>
    <item>Japanese (ja)</item>
    <item>Korean (ko)</item>
    <item>Latvian (lv)</item>
    <item>Lithuanian (lt)</item>
    <item>Norwegian (no)</item>
    <item>Polish (pl)</item>
    <item>Portuguese (pt-PT)</item>
    <item>Romanian (ro)</item>
    <item>Russian (ru)</item>
    <item>Spanish (es)</item>
    <item>Serbian (sr)</item>
    <item>Slovak (sk)</item>
    <item>Slovenian (sl)</item>
    <item>Swedish (sv)</item>
    <item>Ukrainian (uk)</item>
  </array>
</resources>
```

Translate.java (1/4)

```
public class Translate extends Activity {
    private Spinner fromSpinner;
    private Spinner toSpinner;
    private EditText origText;
    private TextView transText;
    private TextView retransText;

    private TextWatcher textWatcher;
    private OnItemSelectedListener itemListener;

    private Handler guiThread;
    private ExecutorService transThread;
    private Runnable updateTask;
    private Future transPending;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.main);
        initThreading();
        findViews();
        setAdapters();
        setListeners();
    }

    @Override
    protected void onDestroy() {
        // Terminate extra threads here
        transThread.shutdownNow();
        super.onDestroy();
    }
}
```

Translate.java (2/4)

```
/** Get a handle to all user interface elements */
private void findViews() {
    fromSpinner = (Spinner) findViewById(R.id.from_language);
    toSpinner = (Spinner) findViewById(R.id.to_language);
    origText = (EditText) findViewById(R.id.original_text);
    transText = (TextView) findViewById(R.id.translated_text);
    retransText = (TextView) findViewById(R.id.retranslated_text);
}

/** Define data source for the spinners */
private void setAdapters() {
    // Spinner list comes from a resource,
    // Spinner user interface uses standard layouts
    ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this, R.array.languages, android.R.layout.simple_spinner_item);
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    fromSpinner.setAdapter(adapter);
    toSpinner.setAdapter(adapter);
    // Automatically select two spinner items
    fromSpinner.setSelection(8); // English (en)
    toSpinner.setSelection(11); // French (fr)
}

/** Setup user interface event handlers */
private void setListeners() {
    // Define event listeners
    textWatcher = new TextWatcher() {
        public void beforeTextChanged(CharSequence s, int start, int count, int after) { /* Do nothing */ }
        public void onTextChanged(CharSequence s, int start, int before, int count) { queueUpdate(1000 /* milliseconds */); }
        public void afterTextChanged(Editable s) { /* Do nothing */ }
    };
    itemListener = new OnItemSelectedListener() {
        public void onItemSelected(AdapterView parent, View v, int position, long id) { queueUpdate(200 /* milliseconds */); }
        public void onNothingSelected(AdapterView parent) { /* Do nothing */ }
    };

    // Set listeners on graphical user interface widgets
    origText.addTextChangedListener(textWatcher);
    fromSpinner.setOnItemSelectedListener(itemListener);
    toSpinner.setOnItemSelectedListener(itemListener);
}
```

Translate.java (3/4)

```
/** Initialize multi-threading. There are two threads: 1) The main graphical user interface thread already started by Android, and 2) The translate thread, which we start using an executor. */
private void initThreading() {
    guiThread = new Handler();
    transThread = Executors.newSingleThreadExecutor();

    // This task does a translation and updates the screen
    updateTask = new Runnable() {
        public void run() {
            // Get text to translate
            String original = origText.getText().toString().trim();

            // Cancel previous translation if there was one
            if (transPending != null) transPending.cancel(true);

            // Take care of the easy case
            if (original.length() == 0) {
                transText.setText(R.string.empty);
                retransText.setText(R.string.empty);
            } else {
                // Let user know we're doing something
                transText.setText(R.string.translating);
                retransText.setText(R.string.translating);

                // Begin translation now but don't wait for it
                try {
                    TranslateTask translateTask = new TranslateTask(
                        Translate.this, // reference to activity
                        original, // original text
                        getLang(fromSpinner), // from language
                        getLang(toSpinner) // to language
                    );
                    transPending = transThread.submit(translateTask);
                } catch (RejectedExecutionException e) {
                    // Unable to start new task
                    transText.setText(R.string.translation_error);
                    retransText.setText(R.string.translation_error);
                }
            }
        }
    };
}
```


Translate.java (4/4)

```
/** Extract the language code from the current spinner item */
private String getLang(Spinner spinner) {
    String result = spinner.getSelectedItem().toString();
    int lparen = result.indexOf('(');
    int rparen = result.indexOf(')');
    result = result.substring(lparen + 1, rparen);
    return result;
}

/** Request an update to start after a short delay */
private void queueUpdate(long delayMillis) {
    // Cancel previous update if it hasn't started yet
    guiThread.removeCallbacks(updateTask);
    // Start an update if nothing happens after a few milliseconds
    guiThread.postDelayed(updateTask, delayMillis);
}

/** Modify text on the screen (called from another thread) */
public void setTranslated(String text) { guiSetText(transText, text); }

/** Modify text on the screen (called from another thread) */
public void setRetranslated(String text) { guiSetText(retransText, text);}

/** All changes to the GUI must be done in the GUI thread */
private void guiSetText(final TextView view, final String text) {
    guiThread.post(new Runnable() {
        public void run() { view.setText(text); }
    });
}
}
```

strings.xml

```
<resources>  
  <string name="app_name">Translate</string>  
  <string name="from_text">From:</string>  
  <string name="to_text">To:</string>  
  <string name="back_text">And back again:</string>  
  <string name="original_hint">Enter text to  
  translate</string>  
  <string name="empty"></string>  
  <string name="translating">Translating...</string>  
  <string name="translation_error">(Translation  
  error)</string>  
  <string name="translation_interrupted">(Translation  
  interrupted)</string>  
</resources>
```

요약

- 인텐트로 다른 액티비티나 프로그램을 실행시켜 보았다.
- 웹뷰를 통해 웹 화면을 액티비티 화면의 일부로 구성하였다.
- 자바스크립트를 통해 안드로이드 프로그램의 메서드들을 호출하였다.
- 안드로이드 응용 프로그램에서 웹 서비스를 이용하였다.

퀴즈

- 인텐트로 브라우징 하는 것과 웹 뷰와는 어떤 차이가 있는가?
- 웹 뷰로 브라우징 하는 경우, AndroidManifest에서 퍼미션을 지정해야 하는 이유는 무엇인가?
- 여기서 소개된 Toast 는 어떤 일을 하는가?
- 자바스크립트와 자바 간의 호출 프로그램에서 소개된 크롬이란 무엇을 의미하는가?
- 자바스크립트와 자바 간의 호출 프로그램에서 index.html은 왜 asset 디렉토리에 들어가는가?
- 번역 프로그램에서 ExecutorService 클래스가 하는 일은 무엇이고, Future 클래스가 하는 일은 무엇인가?
- JSON 은 무엇인가? 안드로이드에서 어떻게 프로그래밍하는가?
- AJAX는 무엇인가?

연습문제

- 하나의 액티비티 화면에 두 개의 웹뷰를 띄워서 브라우징 하는 프로그램을 구현하라.
- 본 강의에서 구현된 브라우저들은 <http://www.dongseo.ac.kr/> 과 같이 URL를 제대로 써줘야 동작한다. www.dongseo.ac.kr 처럼 불완전한 URL에도 작동하도록 프로그램을 수정하라.
- 계산기처럼 두 숫자의 더하기, 빼기, 곱하기, 나누기 값을 반환하는 웹 서비스를 만들고, 이를 안드로이드 응용 프로그램에서 호출하여 사용자가 입력한 두 수의 연산 결과를 보이는 프로그램을 작성하라.
- JSON 형식을 사용하여, 간단한 1 대 1 채팅 프로그램을 구현하라.
- 뉴스와 구글 지도(또는 다음 지도, 또는 네이버 지도)를 결합한 매시업 서비스를 고려해 보라. 이 서비스는 뉴스 문서를 해석하여 관련된 지역의 지도를 보여준다.