توسعه برنامه های موبایل

جلسه چهاردهم مجازی

بخش اول

سحر صادقی

اتصال به اینترنت و دسترسی به منابع آن در اندروید

مروری بر اتصال به اینترنت و دسترسی به منابع از اینترنت در اندروید

آموزش دسترسی به اینترنت در اندروید

اندروید پکیج java.net **را شامل می شود. با وارد کردن این پکیج در پروژه می توانید به منابع و محتوای مورد**

ge which can be used to access network resources. The base class for HTTP network **نیاز از طریق شبکه دسترسی پیدا** access in the java.net package is the HttpURLConnection class.

کنید. کلاس پایه برای اتصال HTTP **در پکیج** java.net **، کلاس** HttpURLConnection **است.**

Android contains the standard Java network java.net packa

اجرای عملیات مربوط به شبکه با API **جاوا طاقت فرسا است. بدین معنی که توسعه دهنده می بایست**

connection**را باز کند/ببندد،** cache**را فعال نموده و مطمئن شود که عملیات مربوط به شبکه در**

(background thread **به صورت موازی و در پس زمینه) اجرا می شوند.**

تعداد زیادی کتابخانه ی کد باز (open source) **وجود دارد که در اختیار برنامه نویس قرار گرفته و انجام عملیات مزبور را آسان می سازد. پرکاربردترین این کتابخانه ها به شرح زیر می باشند:**

- access – OkHttp **برای اتصال و دسترسی بهینه** HTTP
- Retrofit REST based clients – Retrofit **برای سرویس گیرنده های مبتنی بر** REST
- Glide image processing – Glide **برای پردازش تصویر**

آموزش مجوز اتصال به اینترنت

به منظور اتصال به اینترنت، اپلیکیشن شما به مجوز android.permission.INTERNET **احتیاج دارد. در ورژن های جدید کتابخانه های اندروید** (API version **های نوین)، این مجوز به صورت خودکار به اپلیکیشن اعطا می شود.**

آموزش بررسی وضعیت اتصال به اینترنت

دستگاه های اندروید همیشه به اینترنت دسترسی ندارند. برای بررسی اینکه آیا دستگاه مورد نظر به شبکه و اینترنت دسترسی دارد یا خیر، اپلیکیشن شما بایستی مجوز android.permission.ACCESS_NETWORK_STATE **را داشته باشد.**

جهت بررسی وضعیت اتصال به اینترنت، می توانید از کد زیر استفاده کنید.

?

```
1   public boolean isNetworkAvailable() {
2       ConnectivityManager cm = (ConnectivityManager)
3               getSystemService(Context.CONNECTIVITY_SERVICE);
4       NetworkInfo networkInfo = cm.getActiveNetworkInfo();
5       // if no network is available networkInfo will be null
6       // otherwise check if we are connected
7       if (networkInfo != null && networkInfo.isConnected()) {
8           return true;
9       }
10      return false;
11  }
```

همان طور که در بالا گفته شد، برای بررسی وضعیت اتصال به شبکه، اپلیکیشن شما به مجوز ACCESS_NETWORK_STATE**احتیاج دارد.**

آموزش روش های بهینه برای اتصال و دسترسی به اینترنت در اندروید

در اپلیکیشن های اندرویدی، شما باید از اجرای عملیات طولانی و سنگین در UI thread **خودداری نمایید. از جمله ی این عملیات می توان به اتصال به اینترنت و دسترسی به فایل اشاره کرد. از ویرایش ۳.۰ به بعد اندروید، سیستم طوری تعبیه شده است که اگر دسترسی به اینترنت در** UI thread **رخ دهد، خطای** NetworkOnMainThreadException **صادر شده و به تبع آن سیستم به طور ناگهانی از کار می افتد.**

روش معمول و بهینه برای اتصال به اینترنت و دسترسی به منابع سطح وب (در یک اپلیکیشن با کیفیت)، استفاده از یک سرویس است. گفتنی است که می توان از یک activity **یا** fragment **نیز به اینترنت دسترسی را انجام داد، اما استفاده از سرویس برای نیل به این هدف طراحی بهینه تری را برای اپلیکیشن شما رقم می زند و همچنین به ساده نگه داشتن کد** activity **شما کمک شایانی می نماید.**

توجه:

جهت تست می توانید با درج تکه کد زیر در ابتدی متد onCreate() **کلاس** activity **، دسترسی به اینترنت را در** thread **اصلی اپلیکیشن فراهم کنید.**

?

```
StrictMode.ThreadPolicy policy = new StrictMode.
ThreadPolicy.Builder().permitAll().build();
StrictMode.setThreadPolicy(policy);
Unresolved directive in 001_article.adoc - include::../JavaNetworking/010_overview.adoc[] == Web Sockets
```

Web Socket**یک استاندارد مبتنی بر** HTTP **برای تبادل ناهمزمان پیغام**-(asynchronous message based communication)**بین سرویس گیرنده و سرویس دهنده است. به منظور راه اندازی این ارتباط، یک درخواست** HTTP GET **با یک** HTTP header **خاص ایجاد نمایید.اگر سرویس دهنده این درخواست را پذیرفت، سرویس گیرنده و سرویس دهنده می توانند با هم پیغام (داده هایی را) رد و بدل کنند. این پیغام ها می توانند متن یا داده های** binary **باشند. لازم به ذکر است که داده های مورد تبادل می بایست کم حجم باشند چرا که پروتکل** web socket **اساسا به منظور انتقال** payload **و داده های کم حجم تعبیه و طراحی شده است.**

بهترین روش برای تبادل پیغام و داده بین کلاینت و سرور، قرار دادن آن در قالب JSON **می باشد.**

Messages can be text or binary data and should be relatively small, as the web socket protocol is intended to be used with small payloads in the data.
It is good practice to use JSON as data format for the messages.

چک کردن اتصال اپلیکیشن به اینترنت در اندروید

اندروید به اپلیکیشن های شما اجازه می دهد تا به اینترنت یا هر شبکه ی عمومی دیگر متصل شده و عملکردهای مختلف شبکه انجام دهید.

یک دستگاه اندروید می تواند انواع مختلفی از اتصال شبکه را داشته باشد. این مقاله بر روی wi-fi **و یا اتصال شبکه موبایل تمرکز می کند.**

چک کردن اتصال شبکه در اندروید:

قبل از اینکه شما عملکردی را در شبکه اجرا کنید، باید چک کنید که آیا به شبکه و یا اینترنت متصل هستید یا نه.

برای این کار اندروید کلاس ConnectivityManager **را ارائه می دهد.**

لازم است که یک آبجکت از این کلاس را با فراخوانی متد getSystemService() **به عنوان نمونه قرار دهید.**

سینتکس آن مانند زیر می باشد:

```
ConnectivityManager check = (ConnectivityManager) this.context.get
SystemService(Context.CONNECTIVITY_SERVICE);
```

زمانی که شما آبجکت کلاس ConnectivityManager **را به عنوان نمونه قرار دادید، می توانید از متد** getAllNetworkInfo**برای گرفتن اطلاعات همه ی شبکه ها استفاده کنید. این متد آرایه ای از اطلاعات شبکه** (Networkinfo)**را گزارش می دهد. بنابراین باید آن را به این شکل دریافت کنید:**

```
NetworkInfo[] info = check.getAllNetworkInfo();
```

آخرین کاری که باید انجام دهید این است که (Connected State **وضعیت اتصال) مربوط به شبکه را چک کنید، که سینتکس آن به این شکل می باشد:**

```
for (int i = 0; i<info.length; i++){    if (info[i].getState() == N
etworkInfo.State.CONNECTED){       Toast.makeText(context, "Internet
is connected       Toast.LENGTH_SHORT).show();    } }
```

انجام عملیات شبکه:

پس از اینکه وضعیت اتصال به اینترنت را بررسی کردید می توانید هر عملیات مربوط به شبکه را انجام دهید.

در اینجا کد html **وبسایت را از** URL **استخراج می کنیم . کلاس** HttpURLConnection **و** URL **امکانات لازم برای انجام این عملیات را فراهم می کند.**

شما باید با ایجاد لینک وبسایت ، یک شی از کلاس URL **ایجاد کنید . سینتکس آن به صورت زیر است:**

```
String link = "http://www.google.com"; URL url = new URL(link);
```

پس از آن ، باید از متد openConnection **کلاس** url **را فراخوانی کنید و آن را در یک شی** HttpURLConnection **دریافت کنید . پس از آن نیاز به فراخوانی متد** connect **از کلاس** HttpURLConnection **خواهید داشت.**

```
HttpURLConnection conn = (HttpURLConnection) url.openConnection();
conn.connect();
```

آخرین چیزی که نیاز است انجام دهید HTML **را از سایت بگیرید .برای تحقق آن کلاس های** InputStream **و** BufferedReader**خواهید داشت.**

```
InputStream is = conn.getInputStream(); BufferedReader reader = new
BufferedReader(new InputStreamReader(is, "UTF-8")); String webPage =
"",data="";  while ((data = reader.readLine()) != null){    webPage
+= data + "\n"; }
```

علاوه بر این متد ها , متد های دیگری نیز برای کلاس HttpURLConnection **در دسترس است که به شرح زیر است:**

disconnect() : **این متد اتصال را آزاد می کند تا منابع بتوانند دوباره استفاده شوند یا اینکه در نهایت بسته شوند.**

getRequestMethod() : **این متد ، متد درخواستی را که برای اتصال به سرور** remote HTTP **مورد استفاده می سازد مشخص می کند.**

getResponseCode() : **این متد پاسخی که توسط سرور** HTTP **ارسال می شود را** return **می کند.**

setRequestMethod(String method) : **این متد دستور** (command) **درخواست ، که به سرور** HTTP **راه دور ارسال میشود را تنظیم می کند.**

usingProxy() : **این متد نشان می دهد که آیا اتصال از یک پروکسی سرور استفاده می کند یا خیر.**

مثال:

مثال زیر نحوه ی استفاده از کلاس HttpURLConnection **را تبیین می کند. این اپلیکیشن به شما اجازه می دهد کد** HTML **را از وبسایت استخراج کنید.**

این مثال را باید روی یک دستگاه واقعی که به اینترنت متصل است تست کنید.

محتوای فایل: src/MainActivity.java

```
package com.tutorialspoint.myapplication;  import android.app.Progr
essDialog; import android.graphics.Bitmap; import android.graphics.B
itmapFactory;  import android.net.ConnectivityManager; import androi
d.os.Bundle; import android.os.Handler; import android.os.Message; i
mport android.support.v7.app.ActionBarActivity; import android.view.
View;  import android.widget.Button; import android.widget.ImageView
; import android.widget.Toast;  import java.io.IOException; import j
ava.io.InputStream;  import java.net.HttpURLConnection; import java.
net.MalformedURLException; import java.net.URL; import java.net.URLC
onnection;  public class MainActivity extends ActionBarActivity {
private ProgressDialog progressDialog;    private Bitmap bitmap = nu
ll;    Button b1;     @Override    protected void onCreate(Bundle sa
vedInstanceState) {       super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);       b1 = (Button) findView
ById(R.id.button);        b1.setOnClickListener(new View.OnClickList
ener() {          @Override          public void onClick(View v) {
checkInternetConenction();             downloadImage("http://www.tut
orialspoint.com/green/images/logo.png");          }       });    }
private void downloadImage(String urlStr) {       progressDialog = P
rogressDialog.show(this, "", "Downloading Image from " + urlStr);
final String url = urlStr;        new Thread() {          public voi
d run() {             InputStream in = null;              Message ms
g = Message.obtain();             msg.what = 1;              try {
in = openHttpConnection(url);                bitmap = BitmapFactory.
decodeStream(in);                Bundle b = new Bundle();
b.putParcelable("bitmap", bitmap);                msg.setData(b);
in.close();             }catch (IOException e1) {                e1.
printStackTrace();             }             messageHandler.sendMess
age(msg);          }       }.start();    }     private InputStream o
penHttpConnection(String urlStr) {       InputStream in = null;
int resCode = -1;        try {          URL url = new URL(urlStr);
URLConnection urlConn = url.openConnection();           if (!(urlCon
n instanceof HttpURLConnection)) {             throw new IOException
("URL is not an Http URL");          }                           Ht
tpURLConnection httpConn = (HttpURLConnection) urlConn;          htt
pConn.setAllowUserInteraction(false);          httpConn.setInstanceF
ollowRedirects(true);          httpConn.setRequestMethod("GET");
httpConn.connect();          resCode = httpConn.getResponseCode();
if (resCode == HttpURLConnection.HTTP_OK) {             in = httpCon
n.getInputStream();          }       }catch (MalformedURLException e
) {          e.printStackTrace();       }catch (IOException e) {
e.printStackTrace();       }       return in;    }     private Handl
er messageHandler = new Handler() {       public void handleMessage(
Message msg) {          super.handleMessage(msg);          ImageView
```

```
img = (ImageView) findViewById(R.id.imageView);          img.setImag
eBitmap((Bitmap) (msg.getData().getParcelable("bitmap")));
progressDialog.dismiss();       }    };     private boolean checkInt
ernetConenction() {       // get Connectivity Manager object to chec
k connection       ConnectivityManager connec          =(Connectivit
yManager)getSystemService(getBaseContext().CONNECTIVITY_SERVICE);
// Check for network connections       if ( connec.getNetworkInfo(0)
.getState() ==           android.net.NetworkInfo.State.CONNECTED ||
connec.getNetworkInfo(0).getState() ==           android.net.Network
Info.State.CONNECTING ||          connec.getNetworkInfo(1).getState(
) ==           android.net.NetworkInfo.State.CONNECTING ||
connec.getNetworkInfo(1).getState() == android.net.NetworkInfo.State
.CONNECTED ) {             Toast.makeText(this, " Connected ", Toast
.LENGTH_LONG).show();             return true;          }else if (
connec.getNetworkInfo(0).getState() ==              android.net.Netw
orkInfo.State.DISCONNECTED ||             connec.getNetworkInfo(1).g
etState() ==              android.net.NetworkInfo.State.DISCONNECTED
) {                Toast.makeText(this, " Not Connected ", Toast.LEN
GTH_LONG).show();                return false;             }
return false;    }  }
```

محتوای فایل: activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?> <RelativeLayout xmlns:androi
d="http://schemas.android.com/apk/res/android"    xmlns:tools="http:
//schemas.android.com/tools" android:layout_width="match_parent"
android:layout_height="match_parent" android:paddingLeft="@dimen/act
ivity_horizontal_margin"    android:paddingRight="@dimen/activity_ho
rizontal_margin"    android:paddingTop="@dimen/activity_vertical_mar
gin"    android:paddingBottom="@dimen/activity_vertical_margin" tool
s:context=".MainActivity">        <TextView       android:layout_wid
th="wrap_content"       android:layout_height="wrap_content"       a
ndroid:text="UI Animator Viewer"       android:id="@+id/textView"
android:textSize="25sp"       android:layout_centerHorizontal="true"
/>            <TextView       android:layout_width="wrap_content"
android:layout_height="wrap_content"       android:text="Tutorials p
oint"       android:id="@+id/textView2"       android:layout_below="
@+id/textView"       android:layout_alignRight="@+id/textView"
android:layout_alignEnd="@+id/textView"       android:textColor="#ff
36ff15"       android:textIsSelectable="false"       android:textSiz
e="35dp" />            <ImageView       android:layout_width="wrap_co
ntent"       android:layout_height="wrap_content"       android:id="
@+id/imageView"       android:layout_below="@+id/textView2"       an
droid:layout_centerHorizontal="true" />            <Button       andr
oid:layout_width="wrap_content"       android:layout_height="wrap_co
ntent"       android:text="Button"       android:id="@+id/button"
```

```
android:layout_below="@+id/imageView"        android:layout_centerHor
izontal="true"        android:layout_marginTop="76dp" />  </RelativeL
ayout>
```

محتوای فایل: Strings.xml

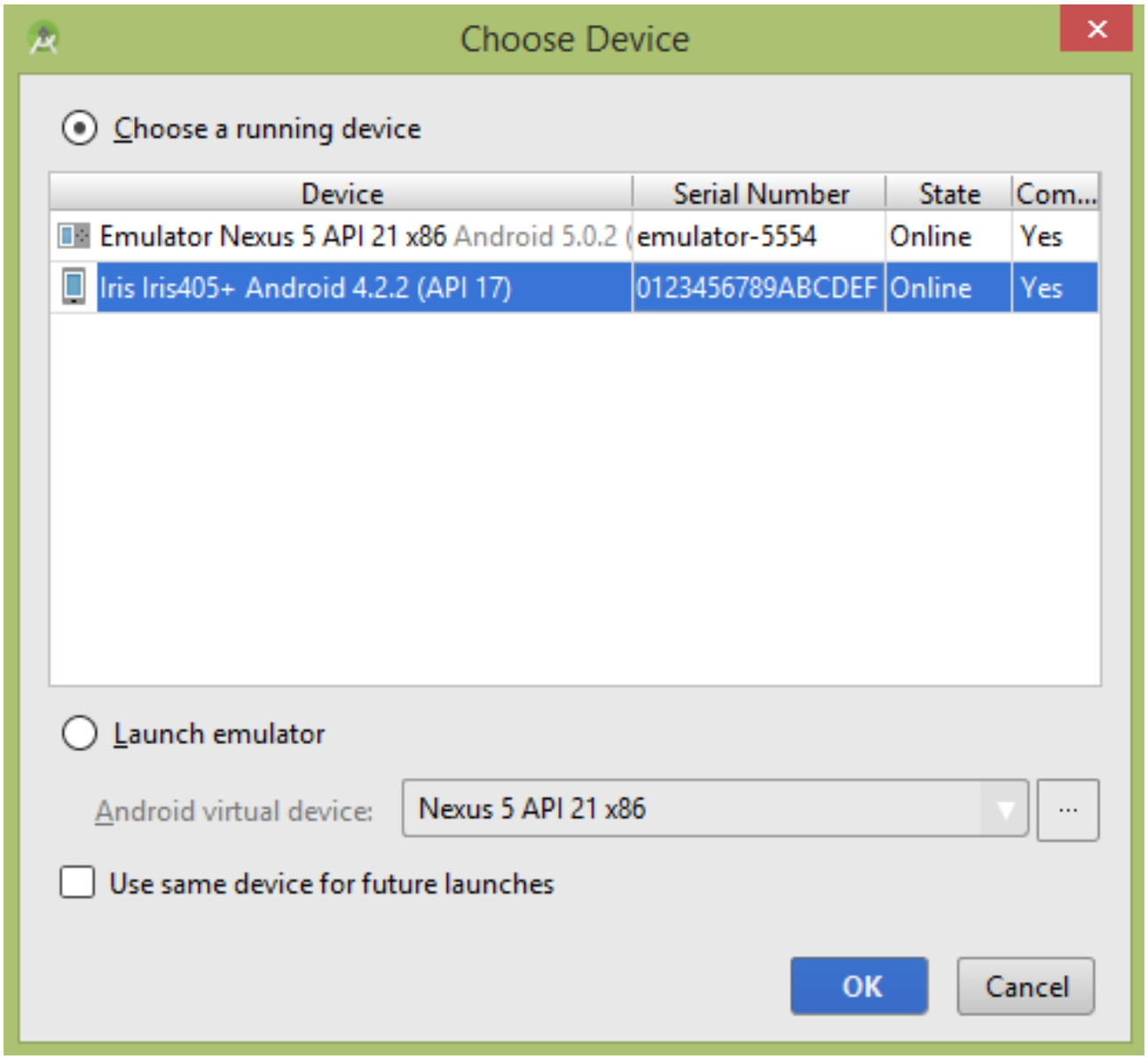
```
 <resources>    <string name="app_name">My Application</string> </re
sources>
```

محتوای فایل: AndroidManifest.xml

```
 <?xml version="1.0" encoding="utf-8"?> <manifest xmlns:android="htt
p://schemas.android.com/apk/res/android"    package="com.tutorialspo
int.myapplication" >    <uses-permission android:name="android.permi
ssion.INTERNET"></uses-permission>    <uses-permission android:name=
"android.permission.ACCESS_NETWORK_STATE"></uses-permission>
<application        android:allowBackup="true"        android:icon="@m
ipmap/ic_launcher"        android:label="@string/app_name"        andr
oid:theme="@style/AppTheme" >              <activity          androi
d:name=".MainActivity"          android:label="@string/app_name" >
<intent-filter>             <action android:name="android.intent.act
ion.MAIN" />             <category android:name="android.intent.cate
gory.LAUNCHER" />          </intent-filter>                </activi
ty>             </application> </manifest>
```

می خواهیم اپلیکیشن را اجرا کنیم. فرض می کنیم که دستگاه شما به کامپیوتر متصل است. برای اجرای اپلیکیشن از اندروید استودیو استفاده کنید و پروژه ی اصلی اکتیویتی را باز کنید. روی آیکون Run **از نوار ابزار کلیک کنید.**

قبل از شروع اپلیکیشن در اندروید استودیو ، پنجره ی زیر برای تعیین محل اجرای اپلیکیشن نمایش داده خواهد شد:


Choose Device
Choose a running device
Device
Serial Number
State
Com...
Emulator Nexus 5 API 21 x86 Android 5.0.2 (
emulator-5554
Online
Yes
Iris Iris405+ Android 4.2.2 (API 17)
0123456789ABCDEF
Online
Yes
Launch emulator
Android virtual device:
Nexus 5 API 21 x86
...
Use same device for future launches
OK
Cancel