



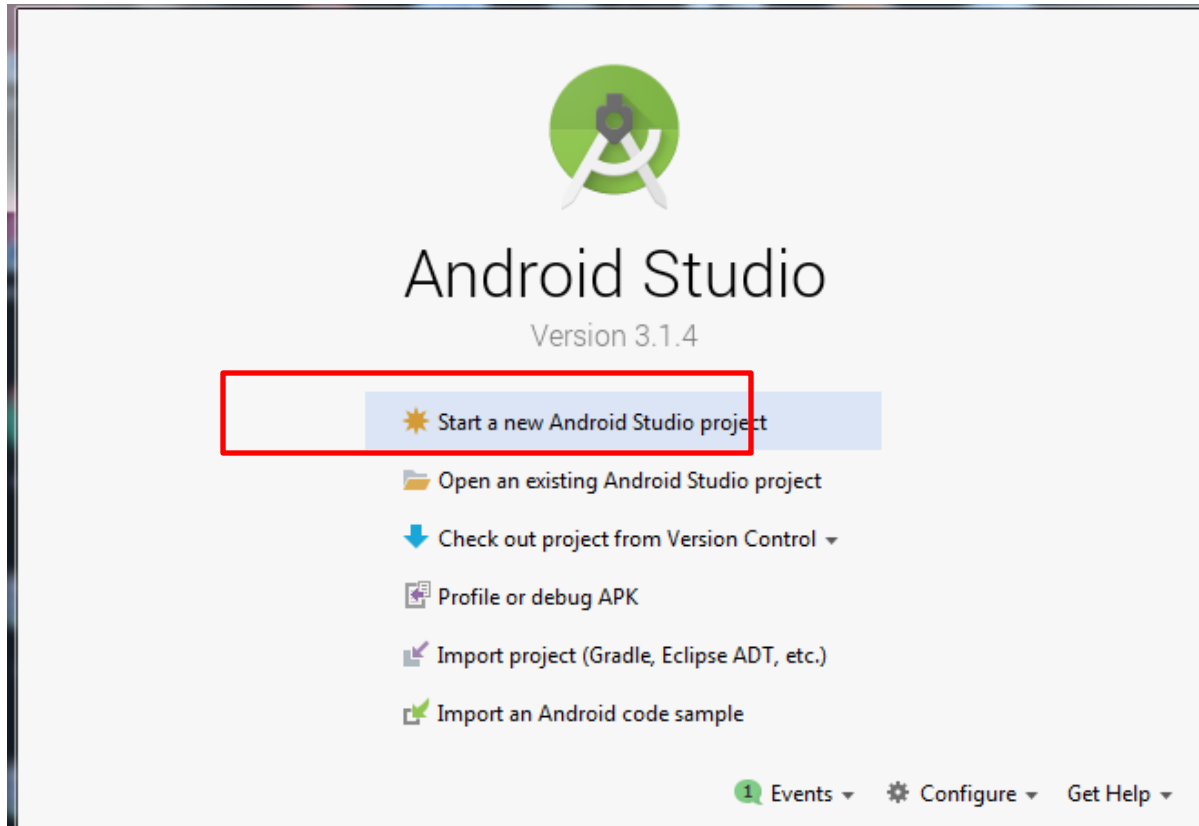
LẬP TRÌNH SERVER CHO ANDROID

BÀI 8: KẾT NỐI ANDROID VỚI NODE SERVER

- ⦿ Cài đặt Node Server với Android Studio
- ⦿ Cấu hình Node Server với Android Studio
- ⦿ Ứng dụng chat NodeJS Socket với Android



- ❑ Để tạo project tiến hành mở ứng dụng
- ❑ Click vào 'Start a new Android Studio project'



☐ Đặt tên ứng dụng sau đó click 'Next'

Application name

Nodejs server

Company domain

administrator.example.com

Project location

C:\Users\Administrator\AndroidStudioProjects\nodejsserver

Package name

com.example.administrator.nodejsserver Edit

☐ Include C++ support

☐ Include Kotlin support

Previous **Next** Cancel Finish

- ❑ Tới đây có rất nhiều thiết bị, chọn 'Phone and Tablet' và chọn phiên bản Android là 4.4(KitKat), sau đó click 'Next'
- ❑ Lưu ý: các bạn có thể chọn phiên bản cao hơn.

Select the form factors and minimum SDK

Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

☒ **Phone and Tablet**

API 19: Android 4.4 (KitKat) ▼

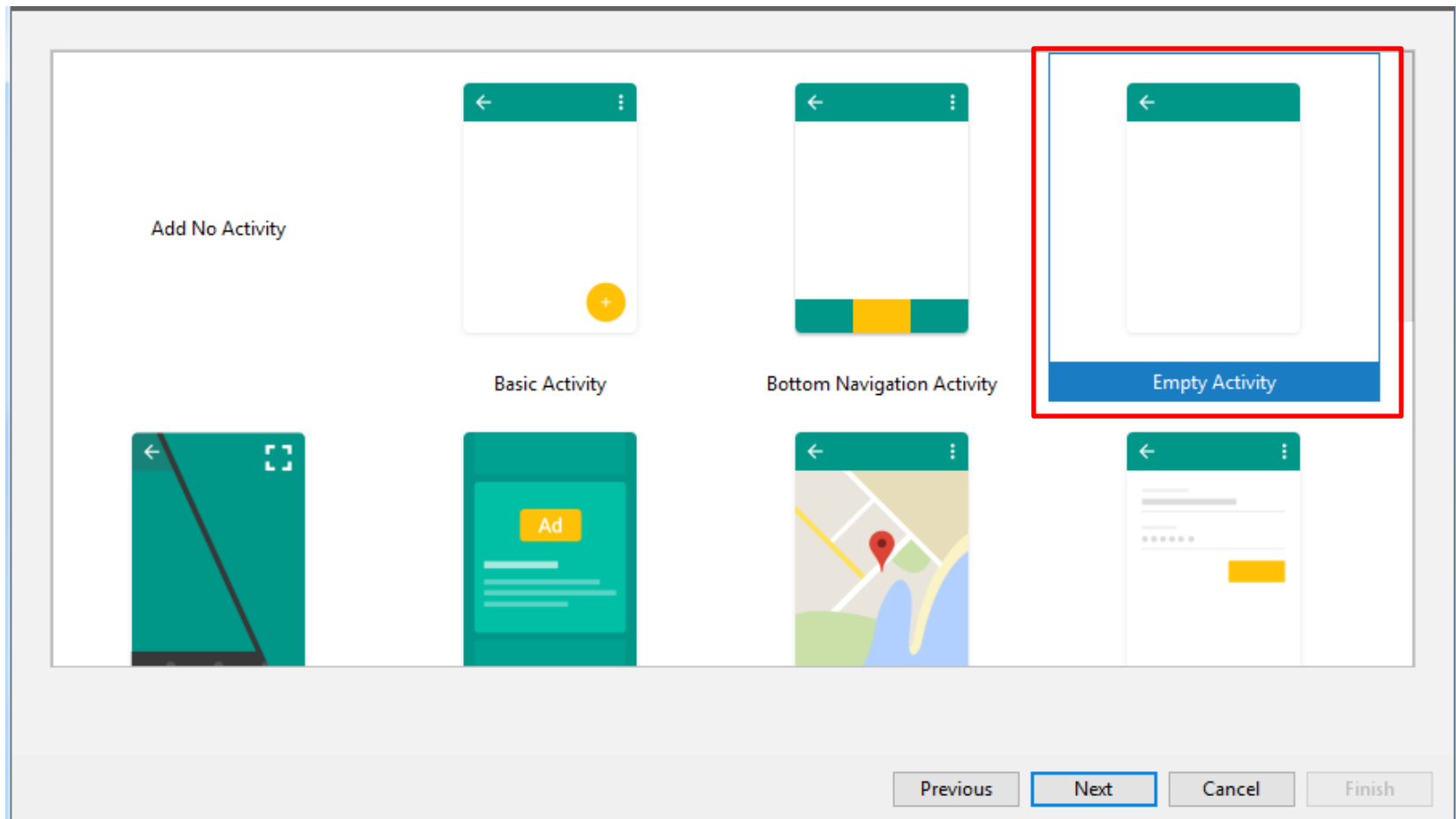
By targeting **API 19 and later**, your app will run on approximately **95,3%** of devices. [Help me choose](#)

☐ Include Android Instant App support

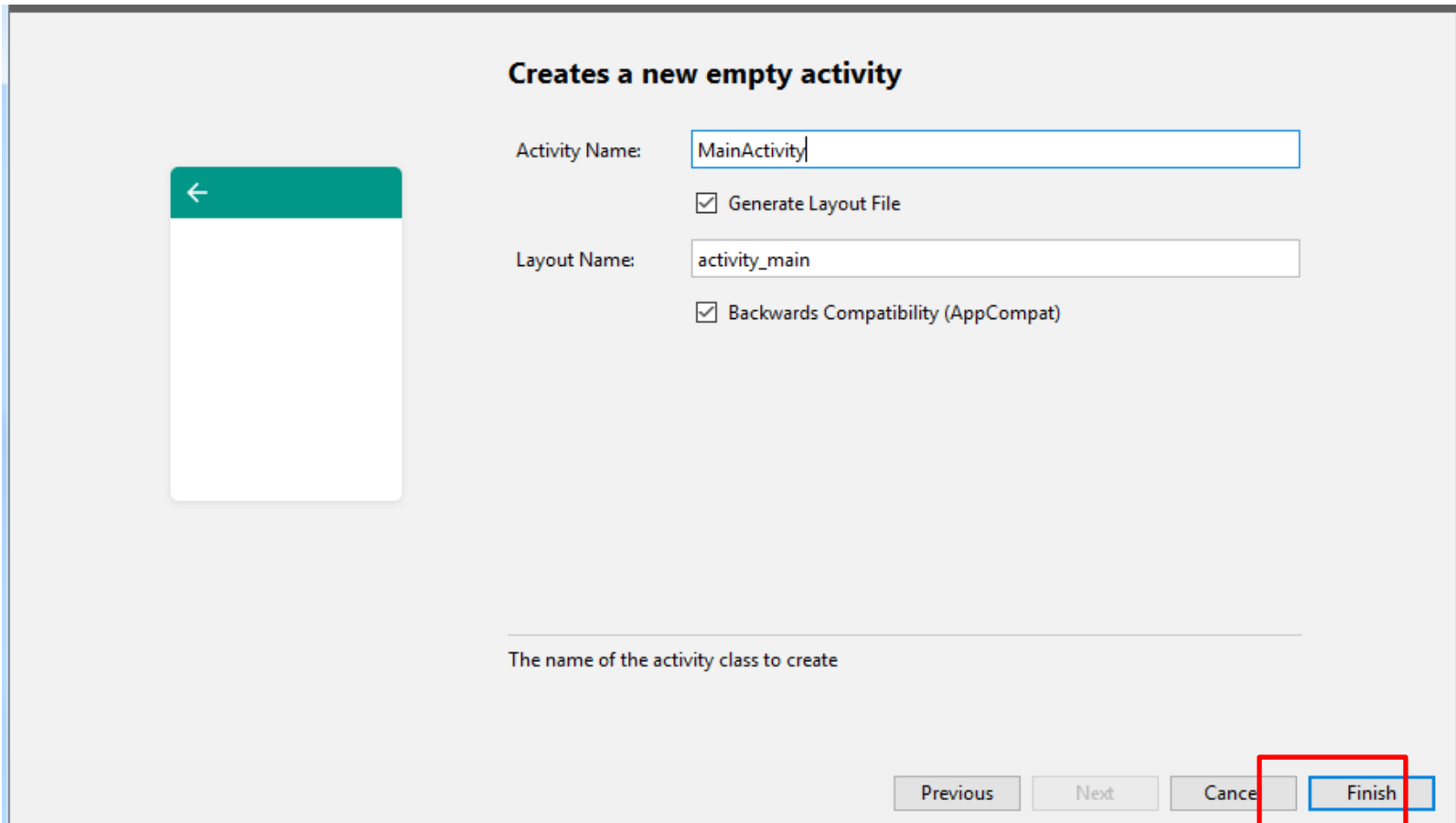
☐ ...

Previous **Next** Cancel Finish

❑ Tại đây chọn 'Add No Activity' sau đó click 'Next'



❑ Các bạn click vào nút finish như hình dưới



Creates a new empty activity

Activity Name:

☒ Generate Layout File

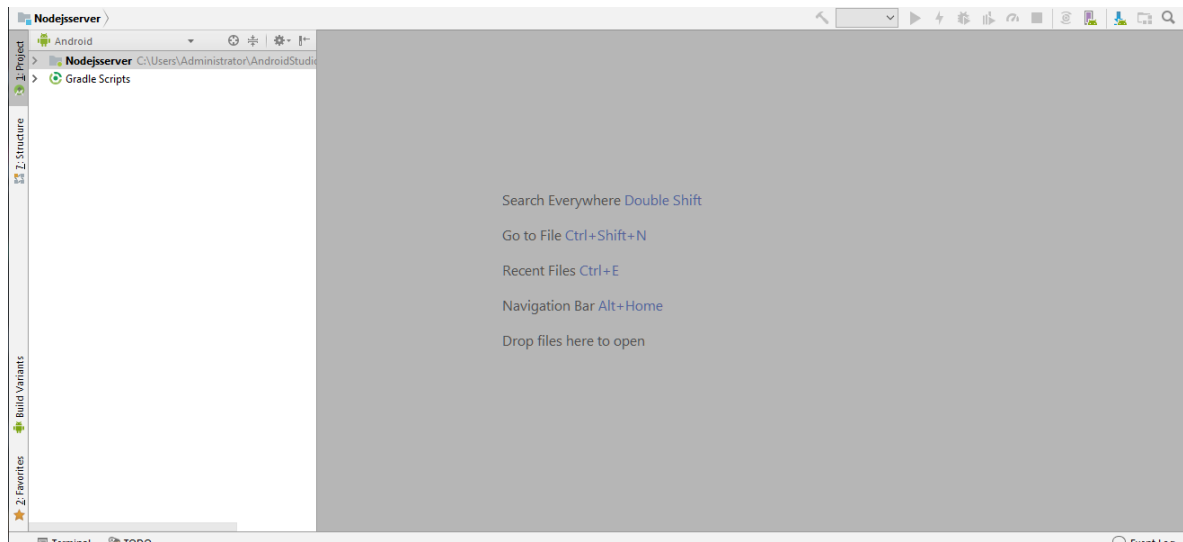
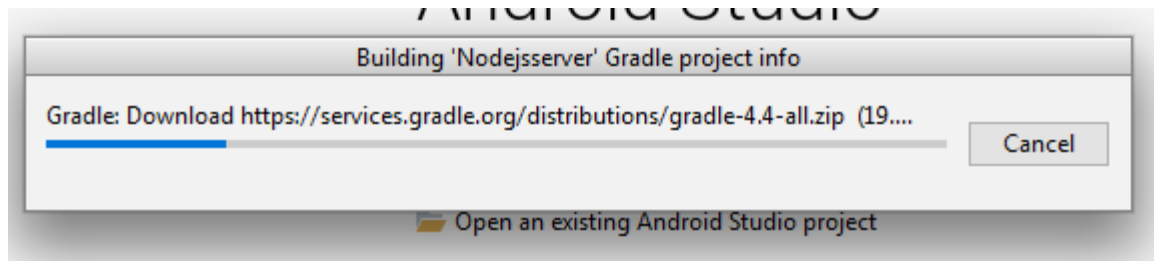
Layout Name:

☒ Backwards Compatibility (AppCompat)

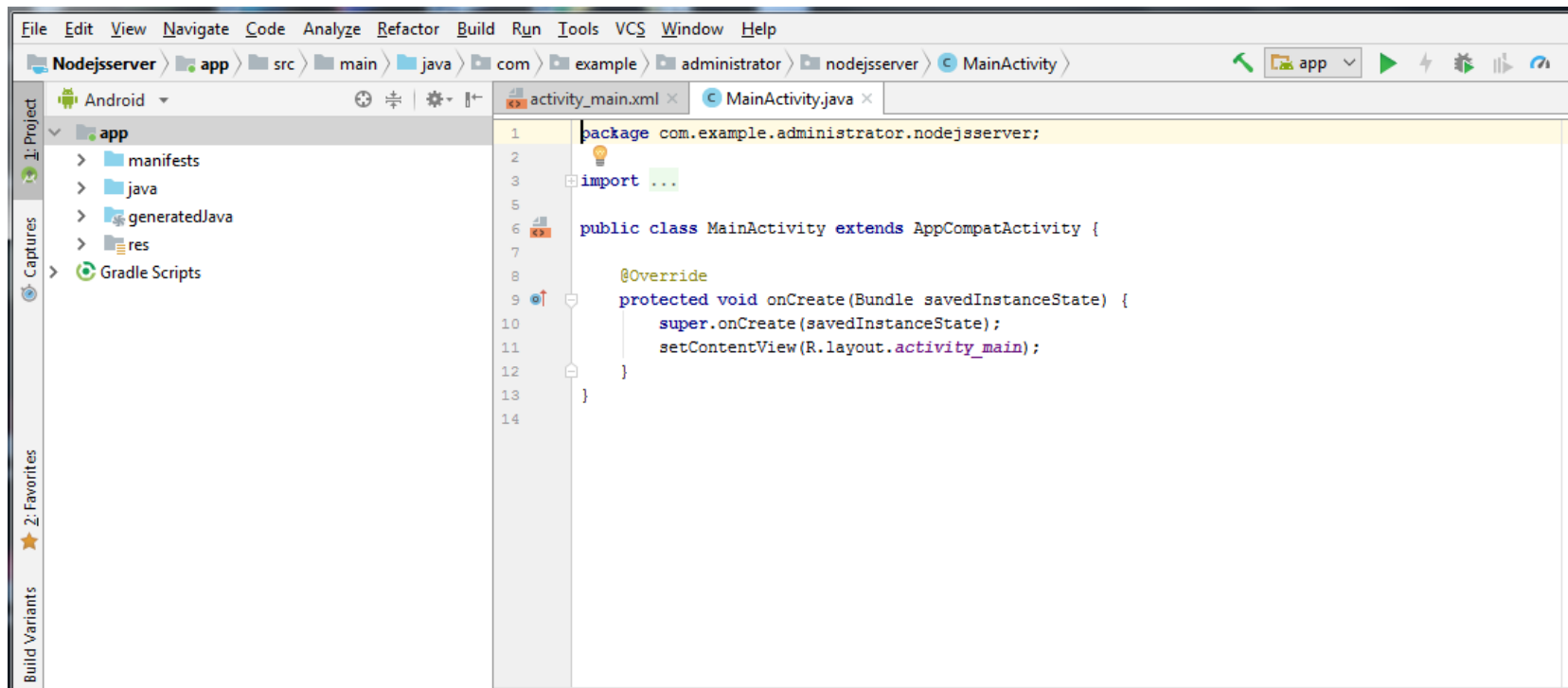
The name of the activity class to create

Previous Next Cancel **Finish**

- ❑ Chờ Download và tiến hành click vào finish
- ❑ Sau khi cài đặt xong sẽ được giao diện dưới



❑ Sau khi tải hoàn chỉnh: MainActivity.java



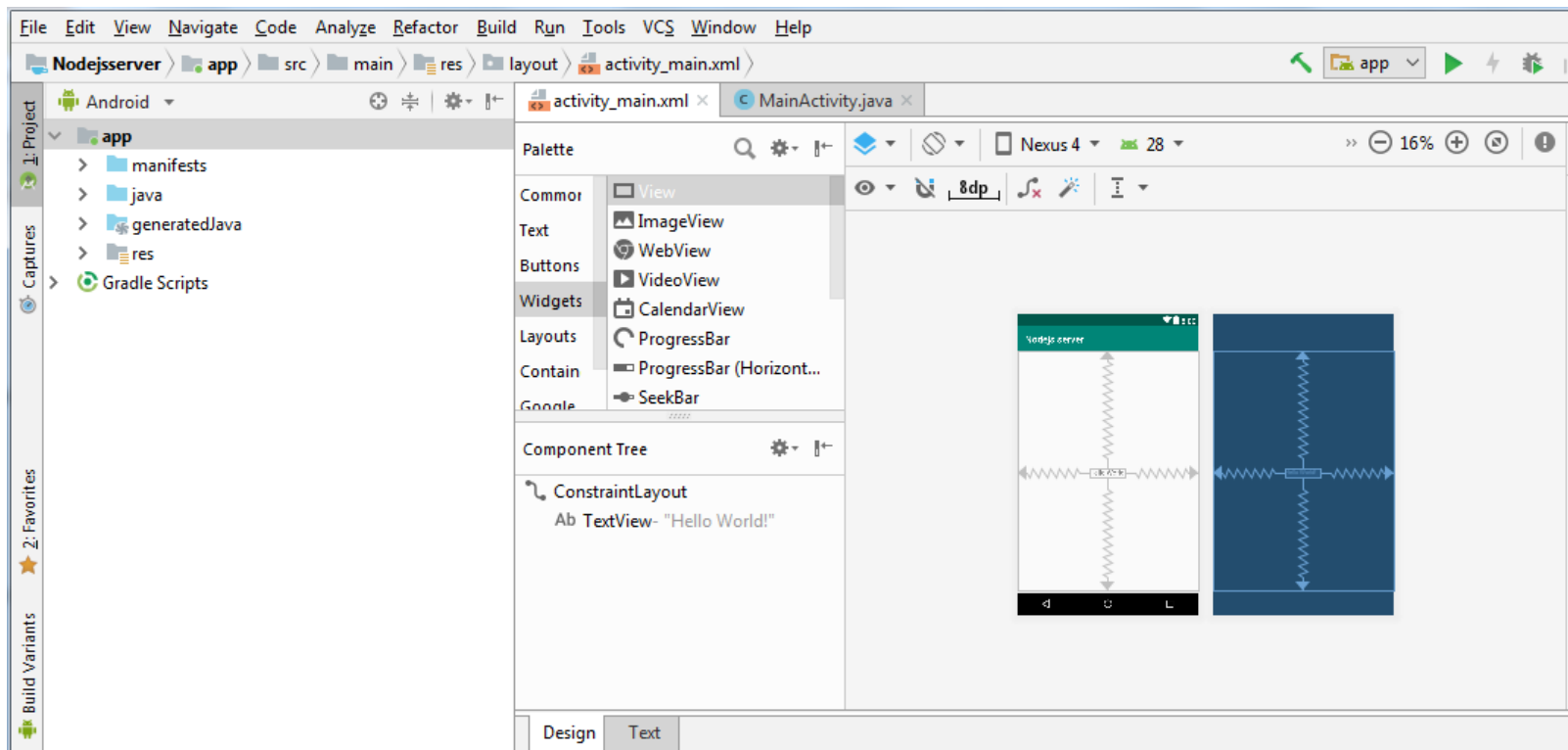
The screenshot shows the Android Studio interface. The top menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The breadcrumb navigation at the top indicates the current file path: Nodejsserver > app > src > main > java > com > example > administrator > nodejsserver > MainActivity. The left sidebar shows the Project view with the following structure:

- 1- Project
 - app
 - manifests
 - java
 - generatedJava
 - res
 - Gradle Scripts
- 2- Favorites
- Build Variants

The main editor displays the MainActivity.java file with the following code:

```
1 package com.example.administrator.nodejsserver;
2
3 import ...
4
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13 }
14
```

❑ Sau khi tải hoàn chỉnh: activity_main.xml



- ❑ Đầu tiên các bạn tạo file mới tên 'package.json'
- ❑ Tiến hành chèn đoạn code như hình dưới





```
{
  "name": "android",
  "version": "1.0.0",
  "private": "true",
  "dependencies": {
    "express": "*",
    "socket.io": "*",
    "socketio-file-upload": "^0.4.4"
  }
}
```

- ❑ Bây giờ các bạn tiến hành mở terminal lên và chạy dòng lệnh: `npm install`
- ❑ (Node project management): dòng lệnh trên sẽ giúp cho Node.JS hiểu và đọc file `package.json` cấu hình này để tiến hành Download mình cần những thư viện nào cho project

```
Administrator@YZFAXCMZING35CL MINGW64 /e/Hoc Tap/Hoc lap trinh Android/Android
$ npm install
npm notice created a lockfile as package-lock.json. You should commit this file.
added 87 packages in 27.498s

Administrator@YZFAXCMZING35CL MINGW64 /e/Hoc Tap/Hoc lap trinh Android/Android
$ |
```

- ❑ Bây giờ chúng ta tiến hành tạo file mới tên: app.js
- ❑ Để tiến hành xử lý dữ liệu phía Server

Name	Date modified	Type	Size
 node_modules	06/10/2018 2:54 CH	File folder	
 app	06/10/2018 3:00 CH	JScript Script File	0 KB
 package.json	06/10/2018 2:54 CH	JSON File	1 KB
 package-lock.json	06/10/2018 2:54 CH	JSON File	24 KB

- ❑ Tại file app.js mở lên và tiến hành thêm dòng lệnh sau

```
var express = require("express");
var app = express();
var server = require("http").createServer(app);
var io = require("socket.io").listen(server);
var fs = require("fs");
server.listen(process.env.PORT || 3434);

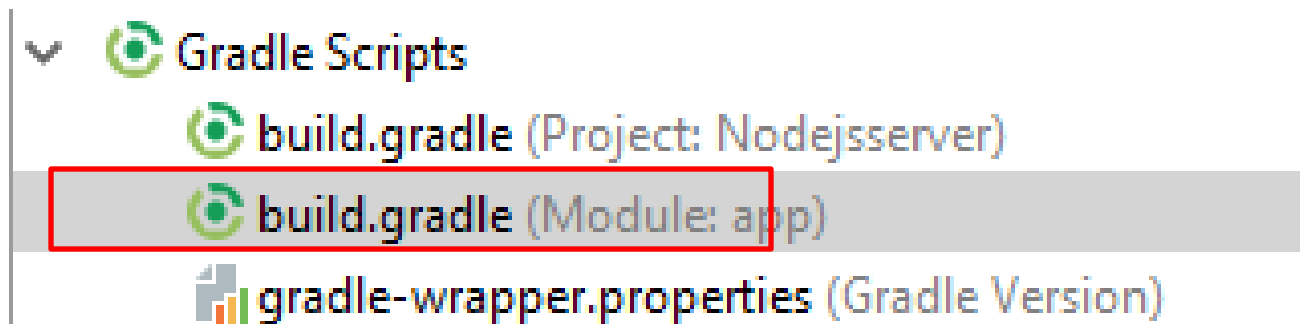
console.log('server is running now');

//Mở kết nối giữa Android và Server
io.sockets.on('connection', function (socket) {

    console.log("Có người kết nối");

});
```

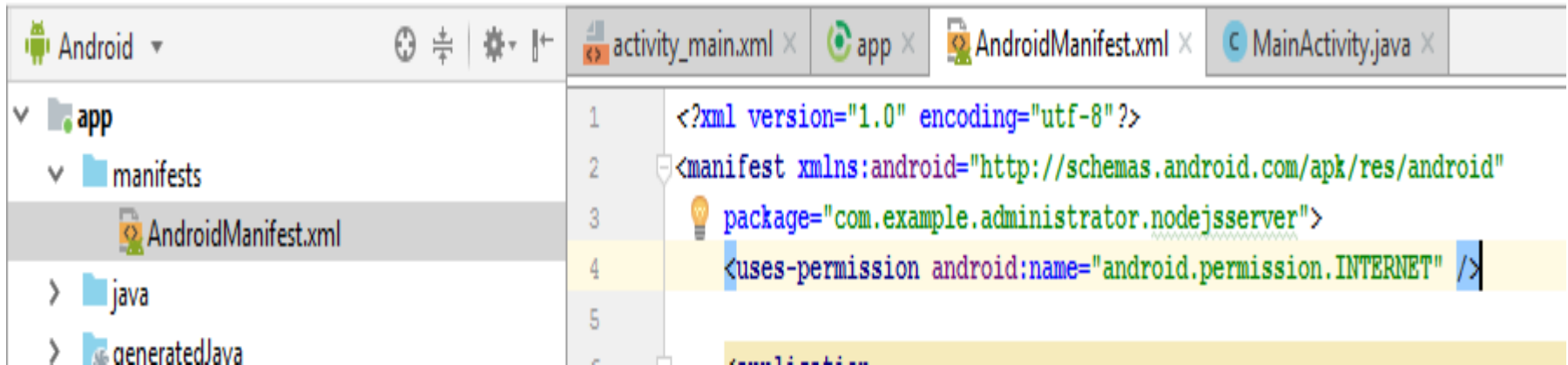
- ❑ Bây giờ là bước quan trọng, các bạn sẽ đưa Node server vào Android như sau.
- ❑ Chi tiết: <https://socket.io/blog/native-socket-io-and-android/>
- ❑ Tiến hành vào lại Android studio project của các bạn
- ❑ Gradle Scripts → build.gradle



- ❑ Tại phần dependencies, tiến hành thêm dòng lệnh như bên dưới
- ❑ Sau khi thêm click vào 'Sync now' để tiến hành thêm thư viện của Socket.io
- ❑ Lệnh: implementation 'com.github.nkzawa:socket.io-client:0.3.0'

```
dependencies {  
    implementation fileTree(dir: 'libs', include: ['*.jar'])  
    implementation 'com.android.support:appcompat-v7:28.0.0'  
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'com.android.support.test:runner:1.0.2'  
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'  
    implementation 'com.github.nkzawa:socket.io-client:0.3.0'  
}
```


- ❑ Về lại manifests → AndroidManifest.xml
- ❑ Tiến hành thêm code như hình dưới
- ❑ Lệnh: `<uses-permission android:name="android.permission.INTERNET" />`



- ❑ Vào MainActivity.java, tiến hành thêm dòng lệnh sau để kết nối đến Node server

```
import com.github.nkzawa.socketio.client.IO;
import com.github.nkzawa.socketio.client.Socket;

private Socket mSocket;
{
    try {
        mSocket = IO.socket("http://chat.socket.io");
    } catch (URISyntaxException e) {}
}

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    mSocket.connect();
}
```

- ❑ Các bạn vào cmd gõ: ipconfig
- ❑ Để xem địa chỉ IP để cấu hình server cho Android nhé
- ❑ Lưu ý: các bạn lấy Port từ máy chủ ảo, lan, wifi đều được nhé

```
Ethernet adapter VirtualBox Host-Only Network:
```

```
Connection-specific DNS Suffix . :  
Link-local IPv6 Address . . . . . : fe80::bc33:a726:9ec8:2b0%14  
IPv4 Address. . . . . : 192.168.56.1  
Subnet Mask . . . . . : 255.255.255.0  
Default Gateway . . . . . :
```

```
Ethernet adapter VirtualBox Host-Only Network #2:
```

- ❑ Sau khi gộp các bước lại, chúng ta sẽ được như hình dưới

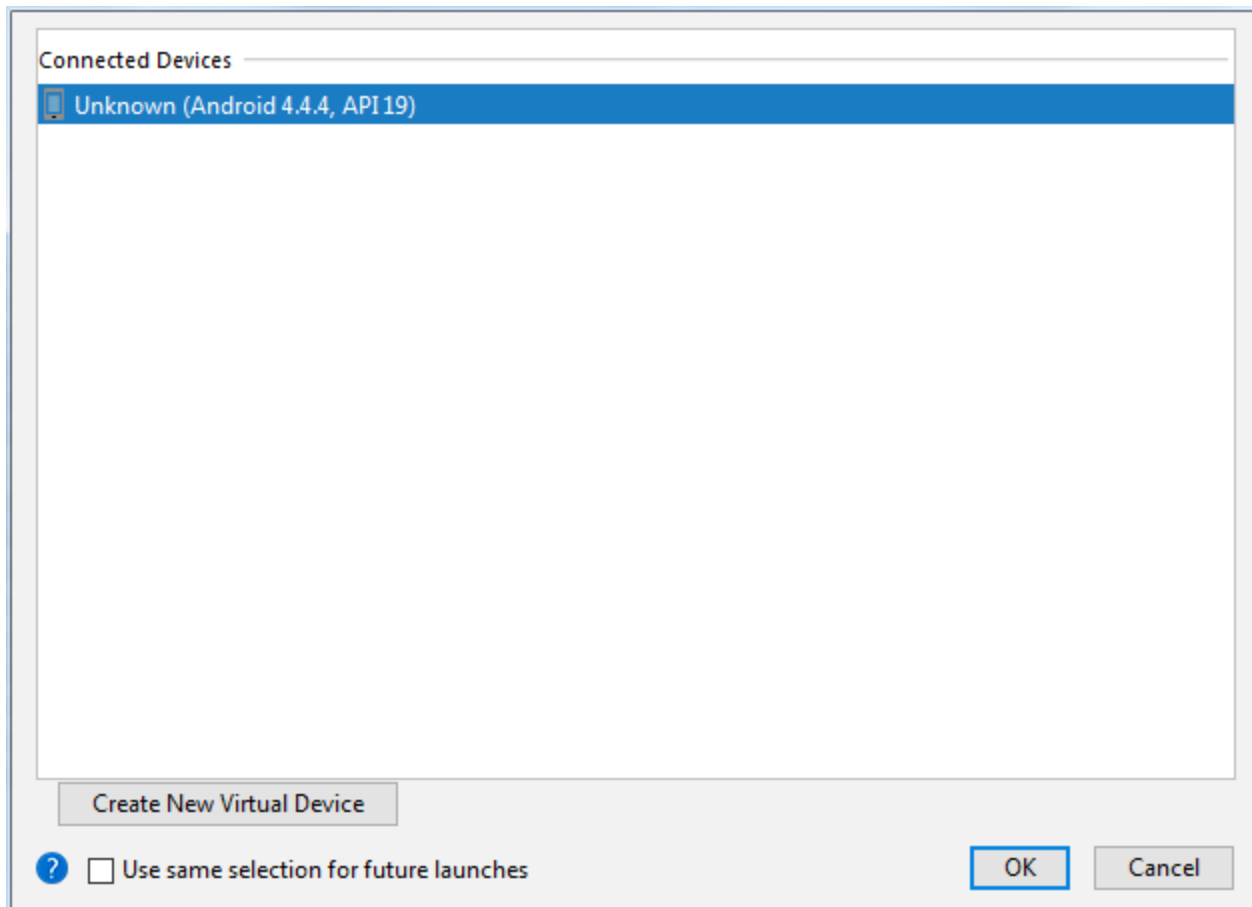
```
//Import SocketIO
import com.github.nkzawa.socketio.client.IO;
import com.github.nkzawa.socketio.client.Socket;

import java.net.URISyntaxException;

public class MainActivity extends AppCompatActivity {

    private Socket mSocket;
    {
        try {
            //Gọi port từ IP Lan hoặc Wifi
            //Lưu ý đây là port Wifi do máy mình phát ra
            mSocket = IO.socket( uri: "http://192.168.210.1:3000");
        } catch (URISyntaxException e) {}
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //Kết nối tới SocketIO
        mSocket.connect();
    }
}
```

- ❑ Tiến hành chạy máy chủ ảo hoặc các bạn chạy trên máy thực bằng USB Debugging

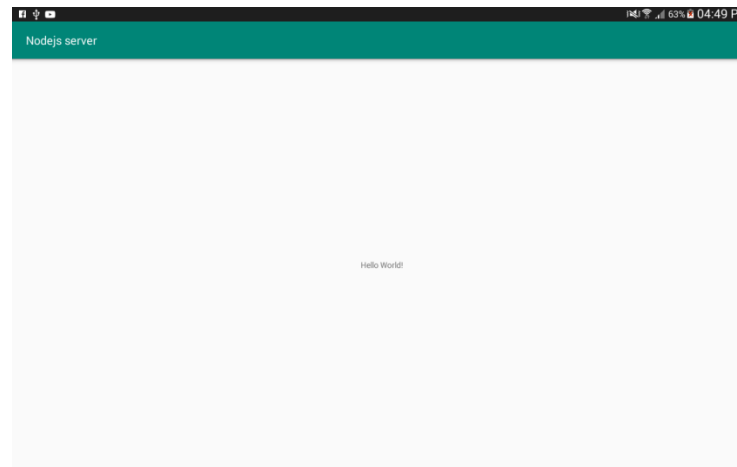


CẤU HÌNH NODE SERVER CHO ANDROID VÀ CHẠY ỨNG DỤNG ĐẦU TIÊN

- ❑ Giờ các bạn vào terminal của Node gõ: node app.js
- ❑ Tiến hành chạy Android để xem kết quả nhé

```
Administrator@YZFAXCMZING35CL MINGW64 /e/Hoc Tap/Hoc lap trinh Android/Android  
$ node app.js  
server is running now  
Có người kết nối
```

Từ Node server



Từ phía Android



DEMO

- Cấu hình Node Js
Với android



- ❑ Để tạo được ứng dụng chat cơ bản, tiến hành mở lại file **app.js** và tiến hành cấu hình cho Node.JS như hình dưới

```
var express = require("express");
var app = express();
var server = require("http").createServer(app);
var io = require("socket.io").listen(server);
var fs = require("fs");
server.listen(process.env.PORT || 3434);

console.log('server is running now');

var listUser = [];
//Mở kết nối giữa Android và Server
io.sockets.on('connection', function (socket) {
  console.log("Có người kết nối");

  socket.on('user_login', function(user_name){
    if(listUser.indexOf(user_name) > 1){
      return;
    }
    listUser.push(user_name);
    socket.user = user_name;
  });
  socket.on('send_message', function(message){
    io.sockets.emit('receiver_message', {data: socket.user+": "+message});
  });
});
```


TẠO ỨNG DỤNG CHAT NODEJS SOCKET VỚI ANDROID – NODE SERVER

```

var express = require("express");
var app = express();
var server = require("http").createServer(app);
var io = require("socket.io").listen(server);
var fs = require("fs");
server.listen(process.env.PORT || 3434);

console.log('server is running now');

var listUser = [];
//Mở kết nối giữa Android và Server
io.sockets.on('connection', function (socket) {
  console.log("Có người kết nối");

  socket.on('user_login', function(user_name){
    if(listUser.indexOf(user_name) > 1){
      return;
    }
    listUser.push(user_name);
    socket.user = user_name;
  });
  socket.on('send message', function(message){
    io.sockets.emit('receiver_message', {data: socket.user+"": "+message});
  });
});
  
```

`io.sockets.on` dùng để lắng nghe các sự kiện kết nối lên server

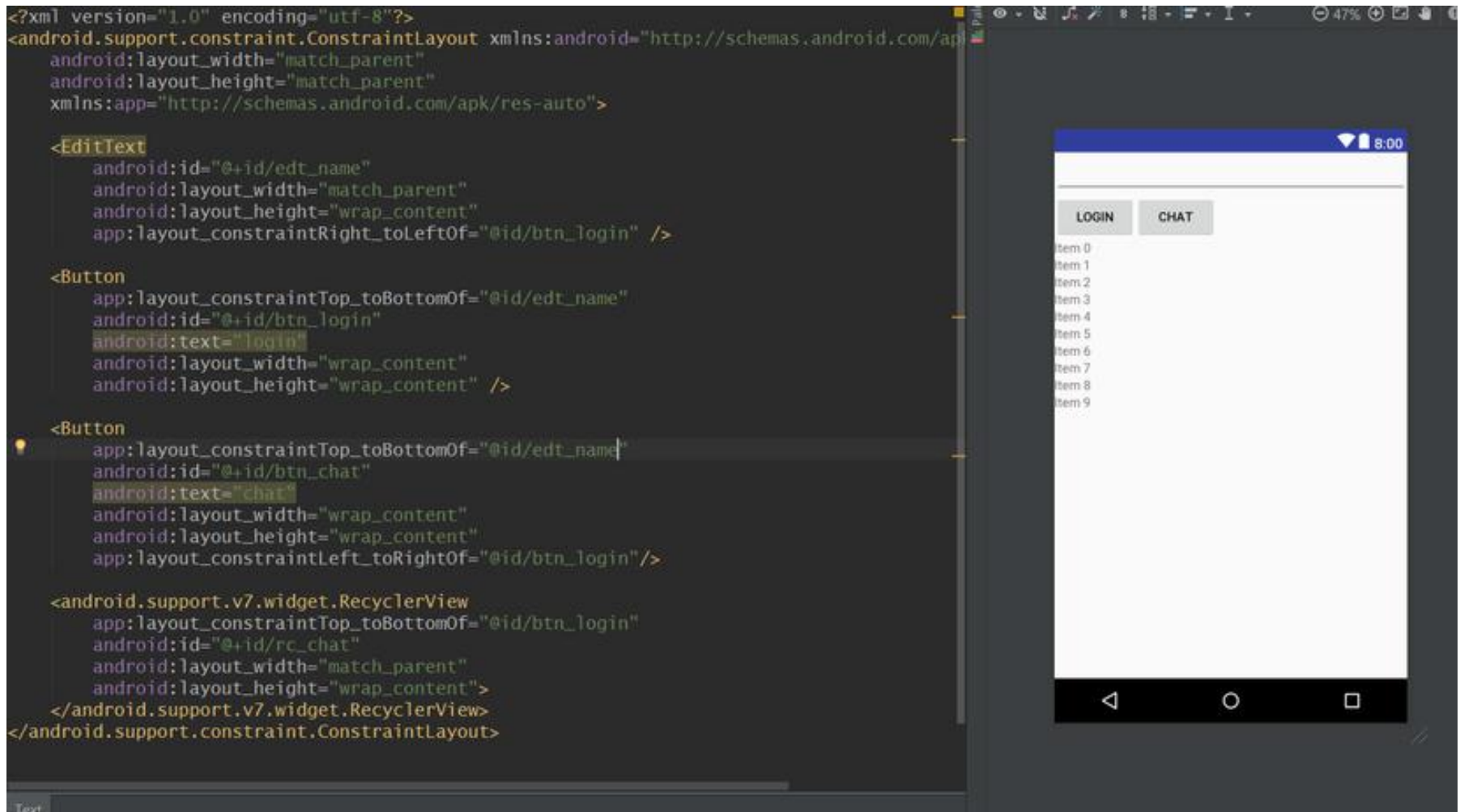
Lắng nghe các sự kiện client gửi lên server với data là message

Gửi message đến client

Để lưu lại user login khi đăng nhập lên server

TẠO ỨNG DỤNG CHAT NODEJS SOCKET VỚI ANDROID – CLIENT SERVER

- ❑ Tại file AndroidManifest.xml tiến hành thêm dòng lệnh sau cho giao diện Android



```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <EditText
        android:id="@+id/edt_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintRight_toLeftOf="@id/btn_login" />

    <Button
        app:layout_constraintTop_toBottomOf="@id/edt_name"
        android:id="@+id/btn_login"
        android:text="login"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <Button
        app:layout_constraintTop_toBottomOf="@id/edt_name"
        android:id="@+id/btn_chat"
        android:text="chat"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintLeft_toRightOf="@id/btn_login"/>

    <android.support.v7.widget.RecyclerView
        app:layout_constraintTop_toBottomOf="@id/btn_login"
        android:id="@+id/rc_chat"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    </android.support.v7.widget.RecyclerView>
</android.support.constraint.ConstraintLayout>
```

The image shows the XML code for an Android app interface. The code defines a `ConstraintLayout` containing an `EditText` field, two `Button` elements (one labeled "login" and one labeled "chat"), and a `RecyclerView` element. The `login` button is positioned below the `EditText`, and the `chat` button is positioned to the right of the `login` button. The `RecyclerView` is positioned below the `login` button. To the right of the code editor, a preview of the app's UI is shown, featuring the same layout: a text input field, a "LOGIN" button, a "CHAT" button, and a list of items labeled "Item 0" through "Item 9".

- ❑ Tiến hành vào cmd, gõ: ipconfig
- ❑ Để xem địa chỉ IP để cấu hình server cho Android nhé
- ❑ Lưu ý: Lấy Port từ máy chủ ảo, lan, wifi đều được nhé

```
Ethernet adapter VirtualBox Host-Only Network:  
Connection-specific DNS Suffix . :  
Link-local IPv6 Address . . . . . : fe80::bc33:a726:9ec8:2b0%14  
IPv4 Address. . . . . : 192.168.56.1  
Subnet Mask . . . . . : 255.255.255.0  
Default Gateway . . . . . :
```

```
Ethernet adapter VirtualBox Host-Only Network #2:
```

TẠO ỨNG DỤNG CHAT NODEJS SOCKET VỚI ANDROID – CLIENT SERVER

- ❑ Mở file MainActivity.java, tiến hành thêm dòng lệnh sau

```
public class ChatActivity extends AppCompatActivity {  
    private RecyclerView mRecyclerViewChat;  
    private Button mButtonLogin;  
    private Button mButtonChat;  
    private EditText mEditTextName;  
    private List<String> mListMessages;  
    private ChatAdapter mChatAdapter;  
    private final String URL_SERVER = "http://192.168.210.1:3434";  
    private Socket mSocket;  
  
    {  
        try {  
            mSocket = IO.socket(URL_SERVER);  
        } catch (URISyntaxException e) {  
            e.printStackTrace();  
        }  
    }  
  
    @Override  
    protected void onCreate(@Nullable Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_chat);  
        mSocket.connect();  
        mSocket.on("receiver_message", onNewMessage);  
        mButtonLogin = findViewById(R.id.btn_login);  
        mButtonChat = findViewById(R.id.btn_chat);  
        mEditTextName = findViewById(R.id.edt_name);  
        mListMessages = new ArrayList<>();  
        mRecyclerViewChat = findViewById(R.id.rc_chat);  
        RecyclerView.LayoutManager layoutManager = new LinearLayoutManager(context, this);  
        mRecyclerViewChat.setLayoutManager(layoutManager);  
        mChatAdapter = new ChatAdapter(mListMessages);  
        mRecyclerViewChat.setAdapter(mChatAdapter);  
    }  
}
```

- ❑ Tiếp tục tại file MainActivity.java thêm dòng lệnh như hình dưới

```
mButtonLogin.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        mSocket.emit( event: "user_login", mEditTextName.getText().toString());  
    }  
});  
  
mButtonChat.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        mSocket.emit( event: "send_message", mEditTextName.getText().toString());  
    }  
});  
}  
  
private Emitter.Listener onNewMessage = new Emitter.Listener() {  
    @Override  
    public void call(final Object... args) {  
        runOnUiThread(new Runnable() {  
            @Override  
            public void run() {  
                JSONObject data = (JSONObject) args[0];  
                String message;  
                message = data.optString( name: "data");  
                mChatAdapter.addMessage(message);  
            }  
        });  
    }  
};  
};  
}
```



DEMO

- Tạo ứng dụng chat

Lập trình Server cho Android

- ❑ Dựa vào kiến thức bạn có được tiến hành tạo cho mình một server node kết nối với android từ đó phát sinh thêm:
 - ❖ 1 người dùng chỉ thông báo duy nhất 1 lượt truy cập
 - ❖ Ip người dùng
 - ❖ Người dùng ngắt kết nối sẽ thông báo



KẾT THÚC