

## Lab 8: Kết nối android với node server

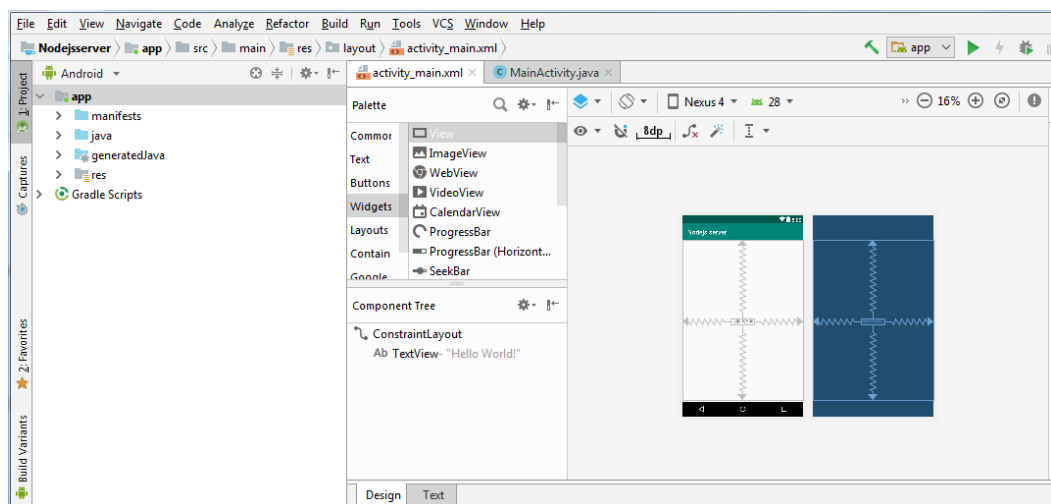
Mục tiêu:

Kết thúc bài thực hành này bạn có khả năng

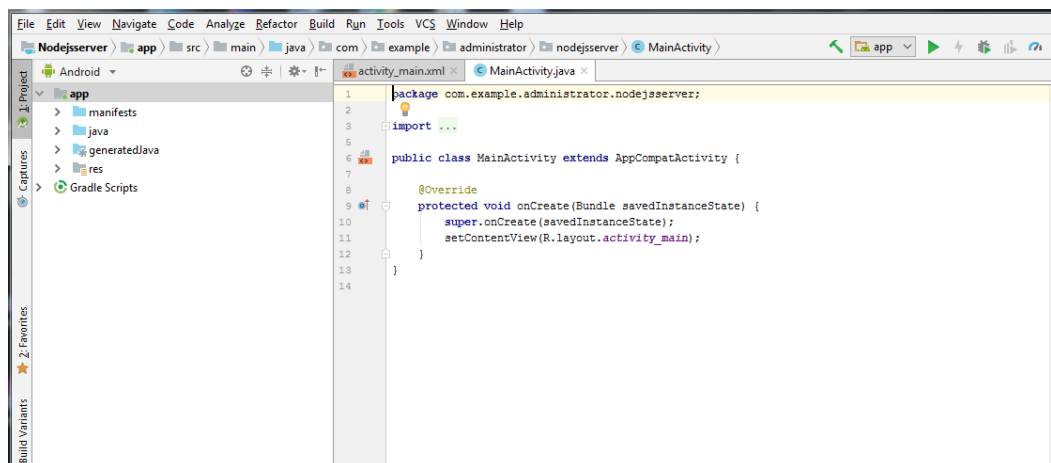
- + Tạo project và Cấu hình Node server cho Android trên Android studio
- + Tạo ứng dụng chat realtime Socket kết hợp Android

Bài 1 (3 điểm): Tạo project trong Android studio và cấu hình Node Js

Kết quả: Chúng ta sẽ được 1 file thiết kế giao diện là ‘activity\_main.xml’ và 1 file để code java là ‘MainActivity.java’

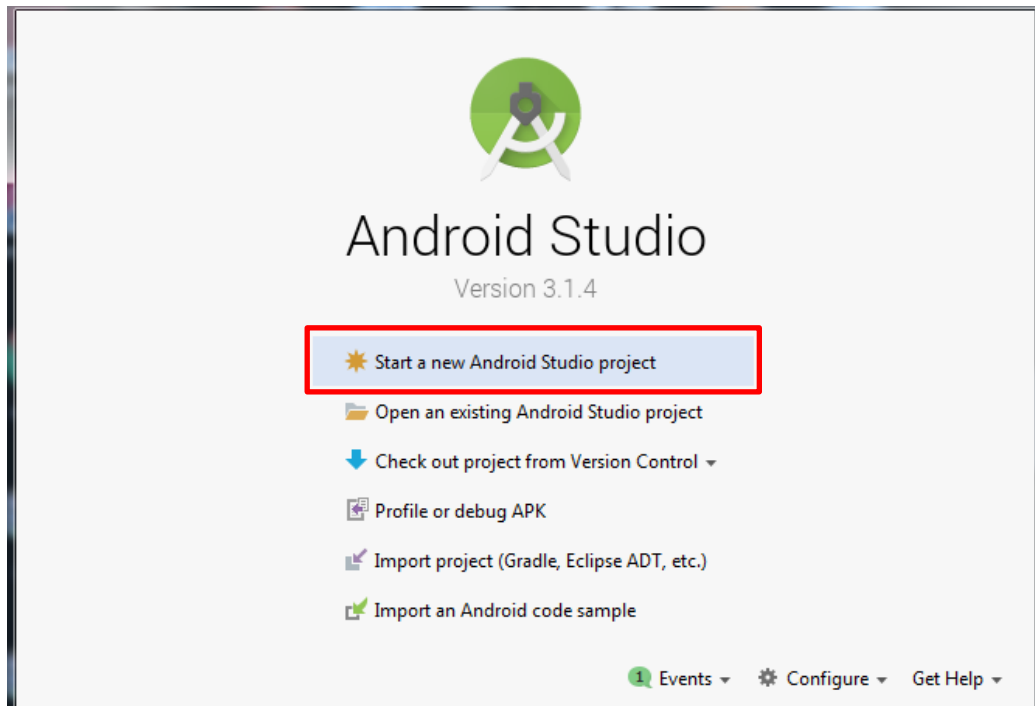


activity\_main.xml

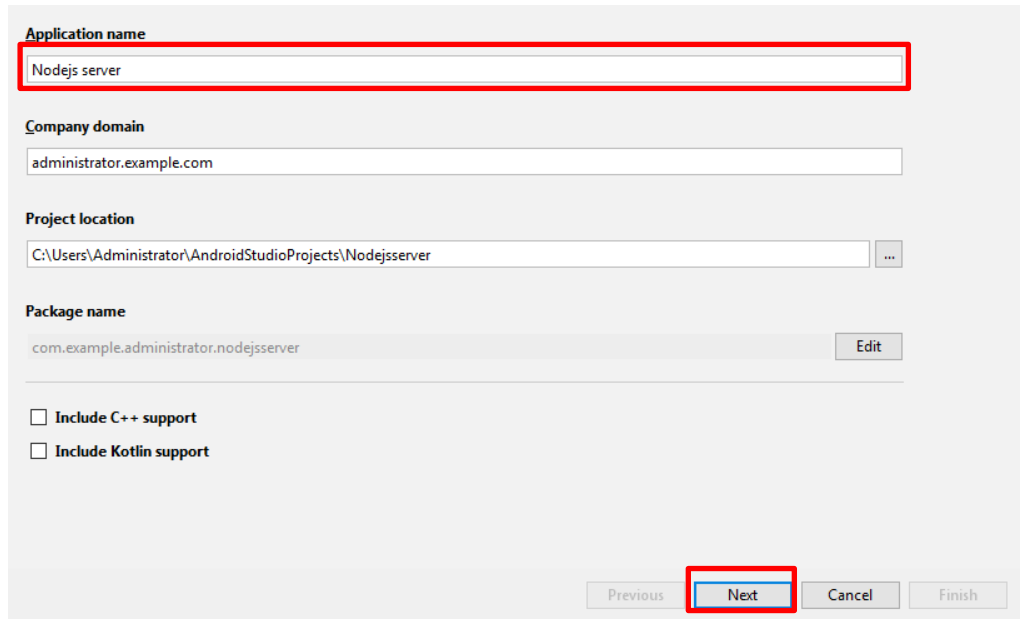


’, MainActivity.java’

Bước 1: mở Android studio lên, Click vào ‘Start a new Android Studio project’

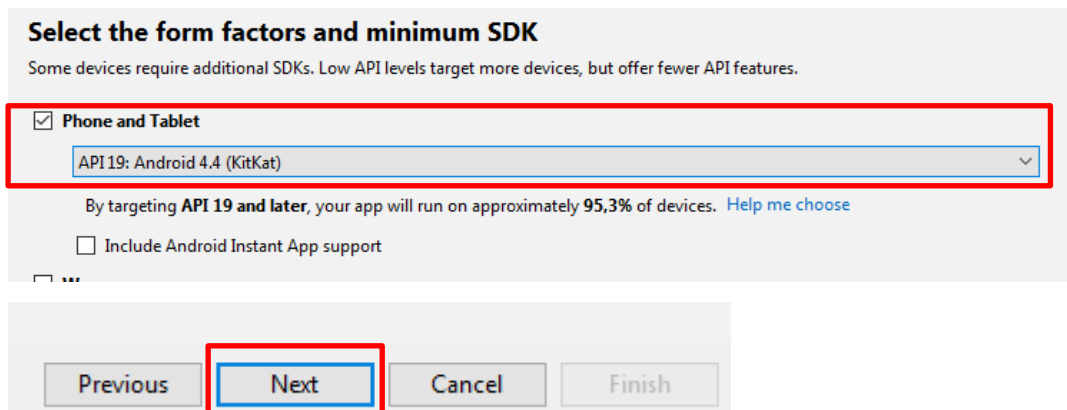


Bước 2: Đặt tên ứng dụng sau đó click ‘Next’

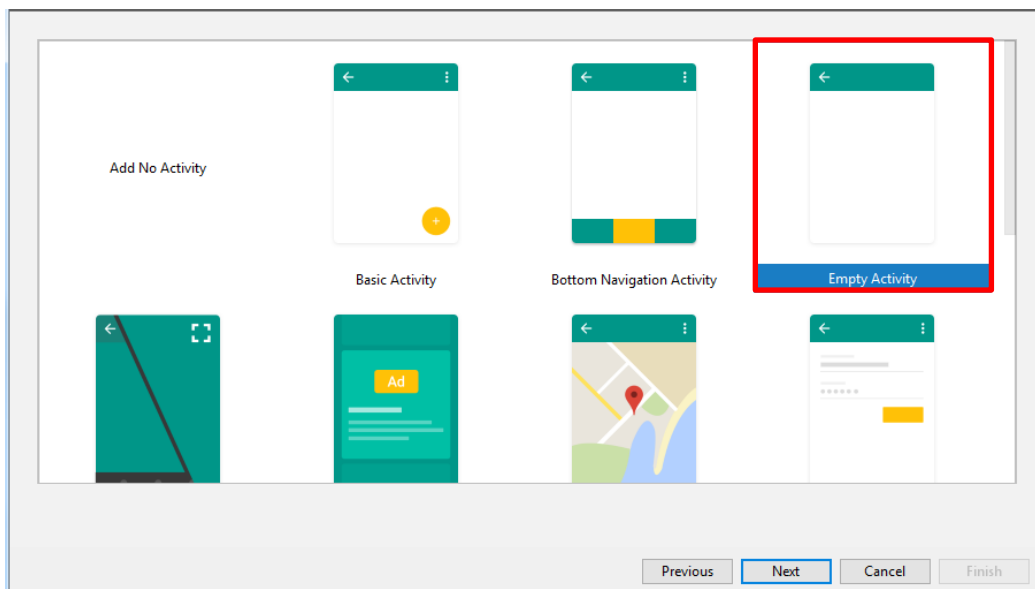


Bước 3: Chọn ‘Phone and Tablet’ và chọn phiên bản Android là 4.4(KitKat), sau đó click ‘Next’

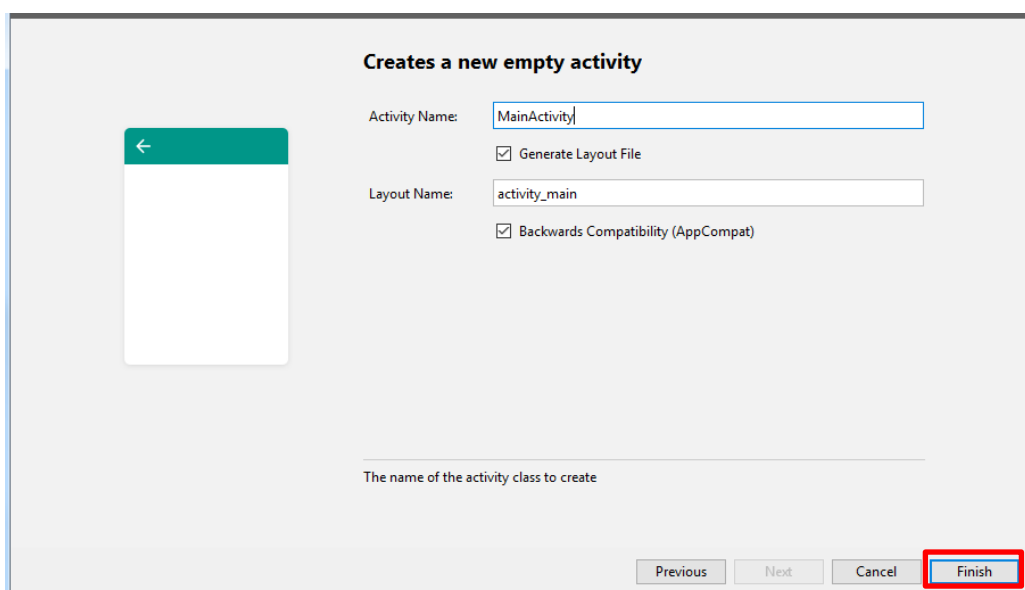
Lưu ý: có thể chọn phiên bản khác tùy vào máy chủ ảo hoặc máy thực tương thích



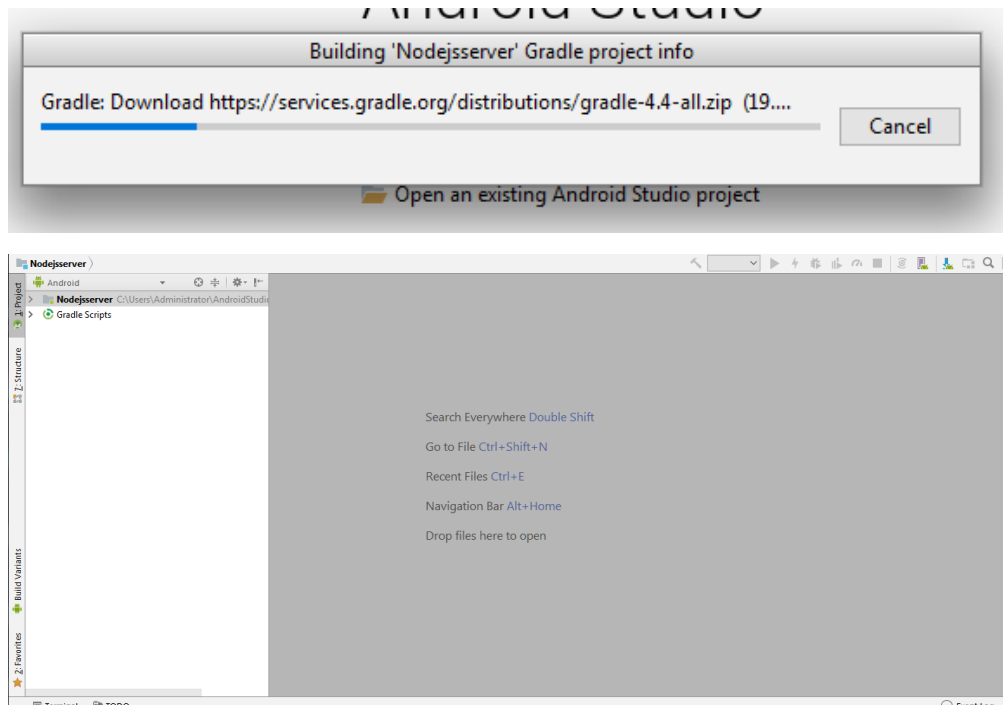
Bước 4: chọn ‘Add No Activity’ sau đó click ‘Next’



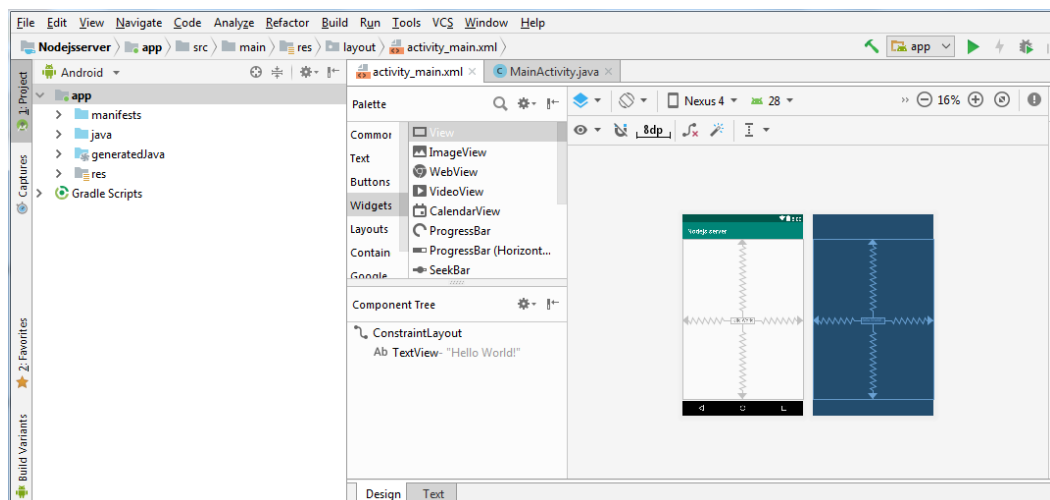
Bước 5: Các bạn click vào nút finish như hình dưới



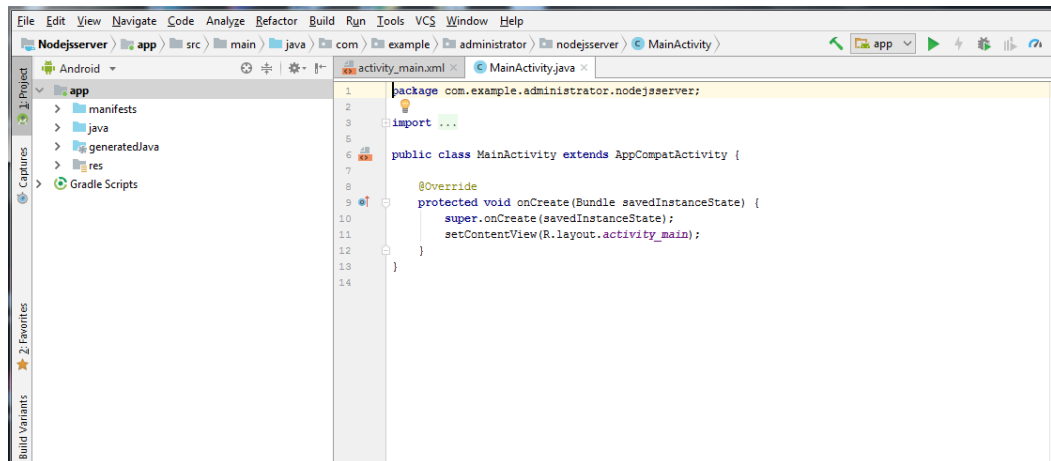
Bước 6: Chờ Download và tiến hành click vào finish  
Sau khi cài đặt xong sẽ được giao diện dưới



Bước 7: Chúng ta sẽ được 1 file thiết kế giao diện là  
'activity\_main.xml' và 1 file để code java là 'MainActivity.java'



## activity\_main.xml



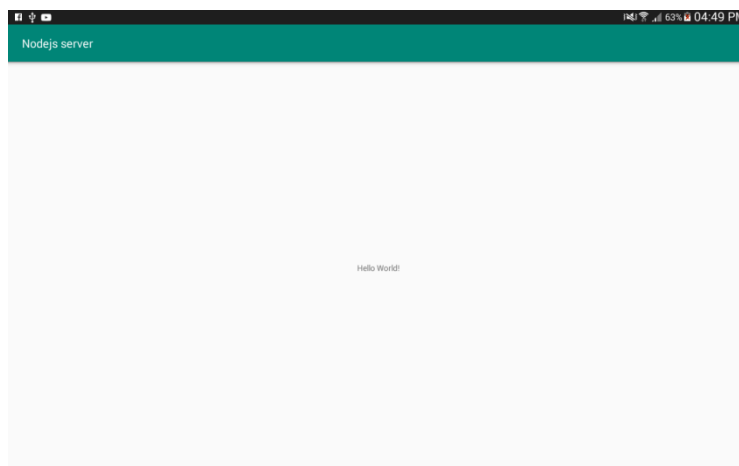
’ MainActivity.java’

Tiếp theo là Cấu hình Node server cho Android

Kết quả: Phía Node server sẽ thông báo khi có người dùng truy cập,  
phía Android sẽ hiển thị layout

```
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 Android

Bước 1: Đầu tiên các bạn tạo file mới tên ‘package.json’

Tiến hành chèn đoạn code như hình dưới

*MOB402 - Lập trình server cho Android*

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

Bước 2: Bây giờ các bạn tiến hành mở terminal lên và chạy dòng lệnh:  
npm install

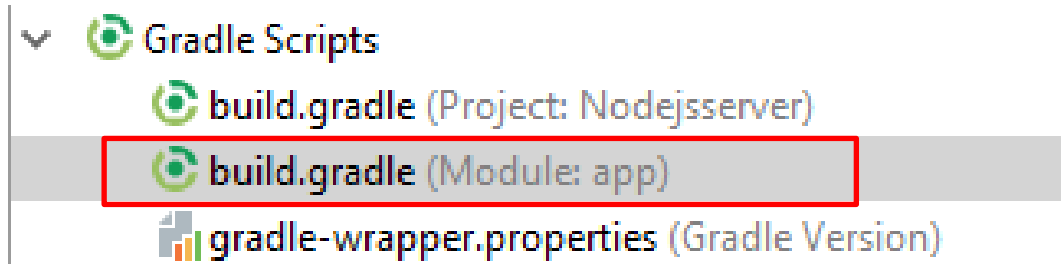
```
Administrator@YZFAXCMZING35CL MINGW64 /e/Hoc Tap/Hoc lap trinh Android/Android
$ npm install
npm notice created a lockfile as package-lock.json. You should commit this file.
added 87 packages in 27.498s

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

Bước 3: 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	
<b>app</b>	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

Bước 4: Tiến hành vào lại Android studio project của các bạn  
Gradle Scripts → build.gradle



Bước 4: Tại phần dependencies, thêm dòng lệnh dưới

Lệnh: `implementation 'com.github.nkzawa:socket.io-client:0.3.0'`

Sau khi thêm click vào ‘Sync now ’ để tiến hành thêm thư viện của Socket.io

```
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'
}
```

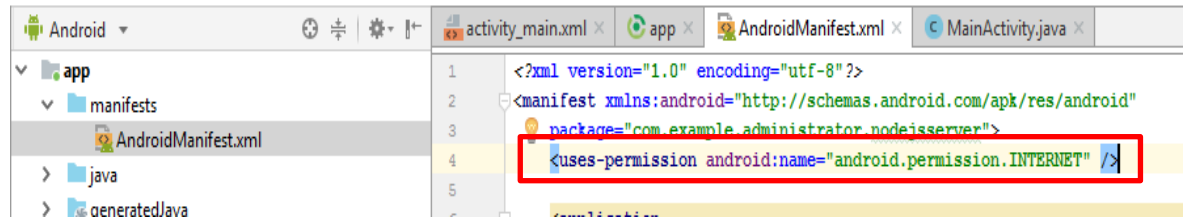
Bước 5: Về lại manifests → AndroidManifest.xml

Tiến hành thêm code như hình dưới

Lệnh: `<uses-permission`

`android:name="android.permission.INTERNET" />`





Bước 6 : 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 VMware Host Interface #2:
```

Bước 7 : Tiến hành chèn IP và dòng lệnh như hình dưới trong  
'MainActivity.java'

```
//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;

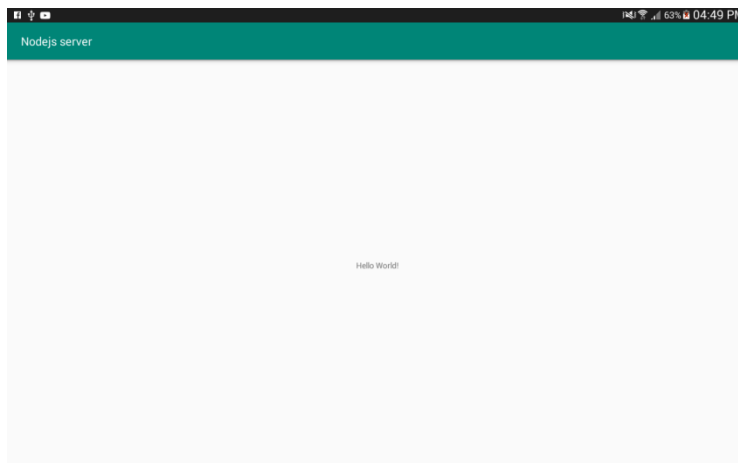
    {
        try {
            //Gọi port từ IP Lan hoặc Wifi
            //Lưu ý đây là port Wifi do máy mình phát ra
            mSocket = IO.socket( uri: "http://192.168.210.1:3000");
        } catch (URISyntaxException e) {}
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //Kết nối tới SocketIO
        mSocket.connect();
    }
}
```

Bước 8 : Phía Node server sẽ thông báo khi có người dùng truy cập, phía Android sẽ hiển thị layout

```
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 Android

## Bài 2 (3 điểm): Tạo ứng dụng chat realtime Socket kết hợp Android

Bước 1(Server): Tại file app.js, tiến hành thêm dòng lệnh sau

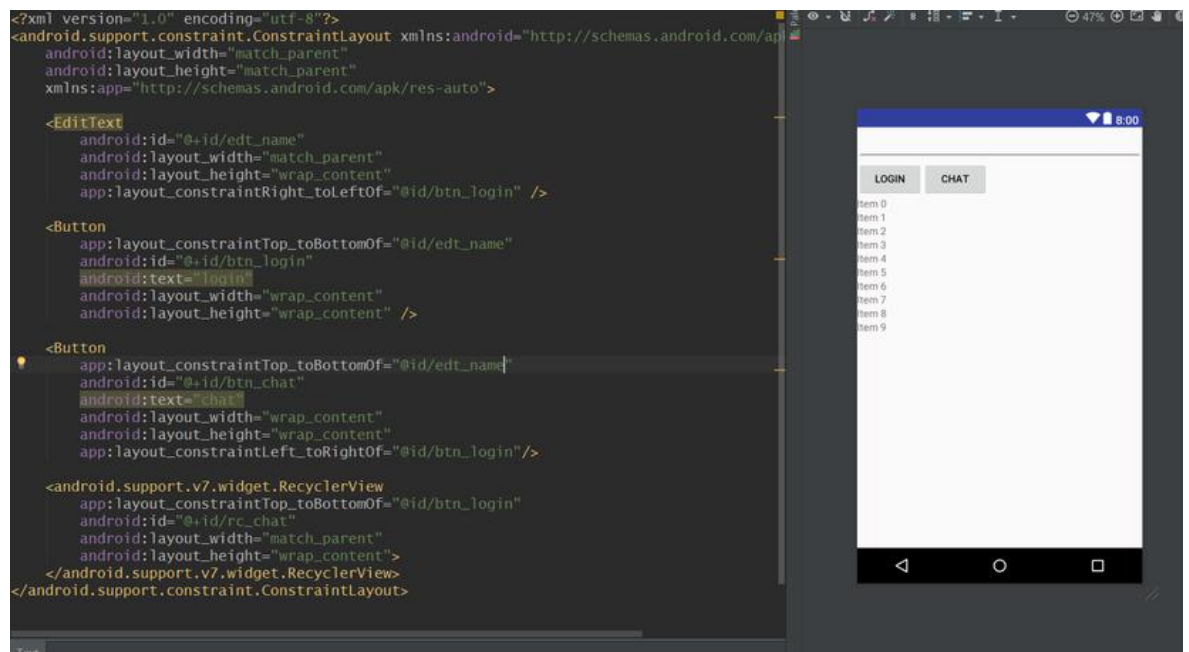
```
var express = require("express");
var app = express();
var server = require("http").createServer(app);
var io = require("socket.io").listen(server);
var fs = require("fs");
server.listen(process.env.PORT || 3434);

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

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});
    });
});
```

Bước 2(Client) : Tại file AndroidManifest.xml tiến hành thêm dòng lệnh sau cho giao diện Android



Bước 3: Tiến hành vào cmd, gõ: ipconfig

```
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 . . . . . :
```

Bước 4: 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);  
    }  
}
```

Bước 5: 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);
            }
        });
    }
};
```

### Bài 3 (4 điểm) : Bài tập nâng cao

- 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