



FPT POLYTECHNIC



android

www.poly.edu.vn

LẬP TRÌNH ANDROID VỚI RESTAPI

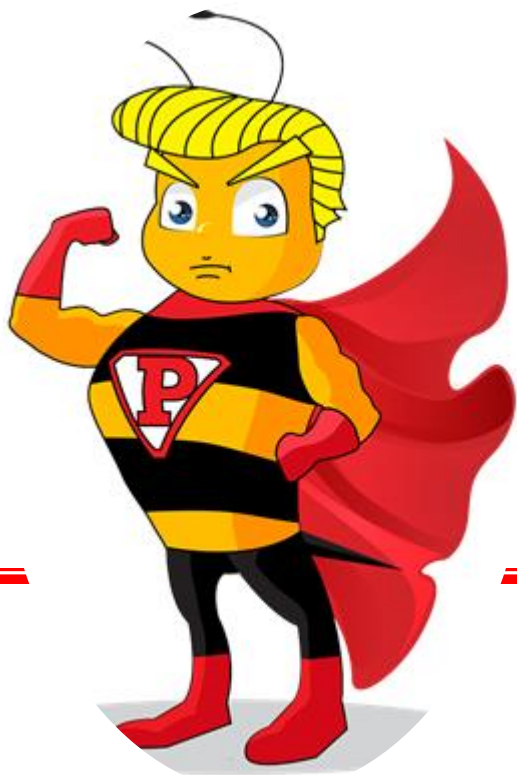
CALL API

- ☐ Call API đăng nhập – đăng ký
- ☐ Call API có kèm upload hình ảnh từ mobile

MỤC TIÊU

- ◎ CALL API ĐĂNG NHẬP — ĐĂNG KÝ
- ◎ CALL API CÓ KÈM UPLOAD HÌNH ẢNH TỪ MOBILE

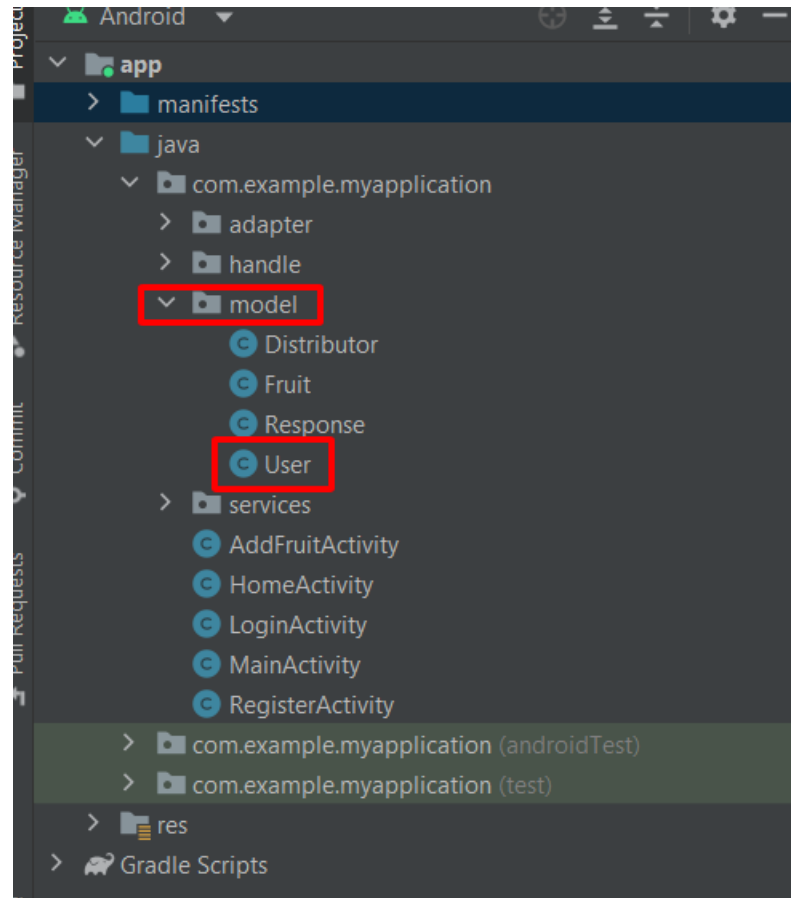




CALL API ĐĂNG KÝ KÈM UPLOAD HÌNH ẢNH

...

❑ Trong folder **model** tạo file **User**



□ Nội dung file **User**

```
package com.example.myapplication.model;

18 usages
public class User {
    3 usages
    private String _id,username,password,email,name,avatar,available;
    5 usages
    private String createdAt,updatedAt;

    1 usage
    public User() {
    }

    no usages
    public User(String _id, String username, String password, String email, String name,
                String avatar, String available, String createdAt, String updatedAt) {
        this._id = _id;
        this.username = username;
        this.password = password;
        this.email = email;
        this.name = name;
        this.avatar = avatar;
        this.available = available;
        this.createdAt = createdAt;
        this.updatedAt = updatedAt;
    }

    1 usage
```

❑ Nội dung file **User** (tiếp theo) – Tạo đầy đủ getter, setter

```
1 usage
public String get_id() { return _id; }

no usages
public void set_id(String _id) { this._id = _id; }

no usages
public String getCreatedAt() { return createdAt; }

no usages
public void setCreatedAt(String createdAt) { this.createdAt = createdAt; }

no usages
public String getUsername() { return username; }

1 usage
public void setUsername(String username) { this.username = username; }

no usages
public String getPassword() { return password; }

1 usage
public void setPassword(String password) { this.password = password; }

no usages
public String getEmail() { return email; }

no usages
public void setEmail(String email) { this.email = email; }

public String getName() { return name; }
```

- ❑ Trong file ApiService thêm phương thức POST form đăng ký

```
1 usage
@Multipart
@POST("register-send-email")
Call<Response<User>> register(@Part("username") RequestBody username,
                             @Part("password") RequestBody password,
                             @Part("email") RequestBody email,
                             @Part("name") RequestBody name,
                             @Part MultipartBody.Part avatar);
```

*Lưu ý: Khi POST data có kèm hình ảnh phải sử dụng **Multipart***

❑ Sử dụng thư viện Glide để hiển thị hình ảnh từ bộ nhớ

```
//Glide  
implementation 'com.github.bumptech.glide:glide:4.15.1'  
annotationProcessor 'com.github.bumptech.glide:compiler:4.14.2'
```

- ❑ Glide là một thư viện open source hỗ trợ load ảnh trên Android. Dùng Glide sẽ đơn giản hóa các công việc mà bạn cần làm khi sử dụng một bức ảnh trong Android đi rất nhiều. Chúng ta không cần quan tâm đến việc decoding, memory and disk caching mà thay vào đó chỉ cần sử dụng interface rất đơn giản từ Glide.

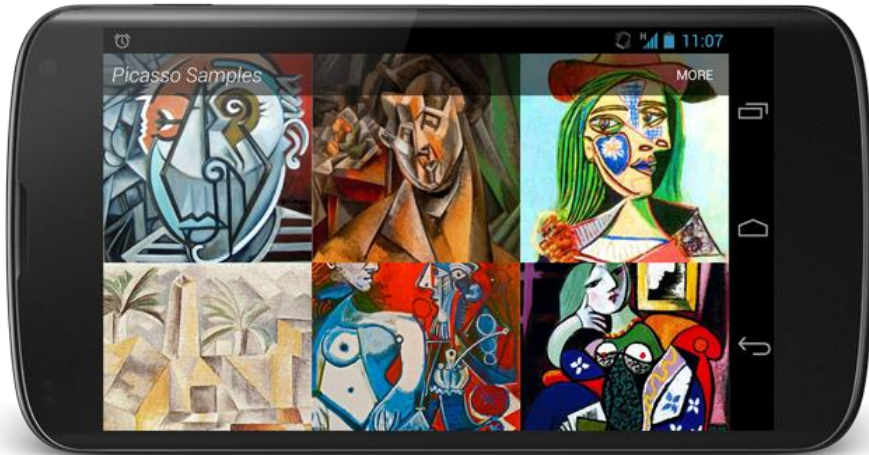


- ❑ Glide hỗ trợ **fetching, decoding** và hiển thị cả ảnh tĩnh hoặc ảnh động đó là điểm nổi bật của nó so với các thư viện load ảnh khác cho android. Glide có thể được dễ dàng đưa vào bất kỳ mô hình mạng nào. Mặc định Glide sử dụng mô hình custom của **HttpURLConnection** tuy nhiên chúng ta có thể dễ dàng cho nó hoạt động với **Volley** hoặc **OkHttp**
- ❑ Link thư viện: <https://github.com/bumptech/glide>

- ❑ Ngoài ra, chúng ta có thể tham khảo thêm một số thư viện tương tự Glide như:
- ❑ **Fresco** (<https://github.com/facebook/fresco>)



❑ Picasso (<https://square.github.io/picasso/>)



Network



Disk



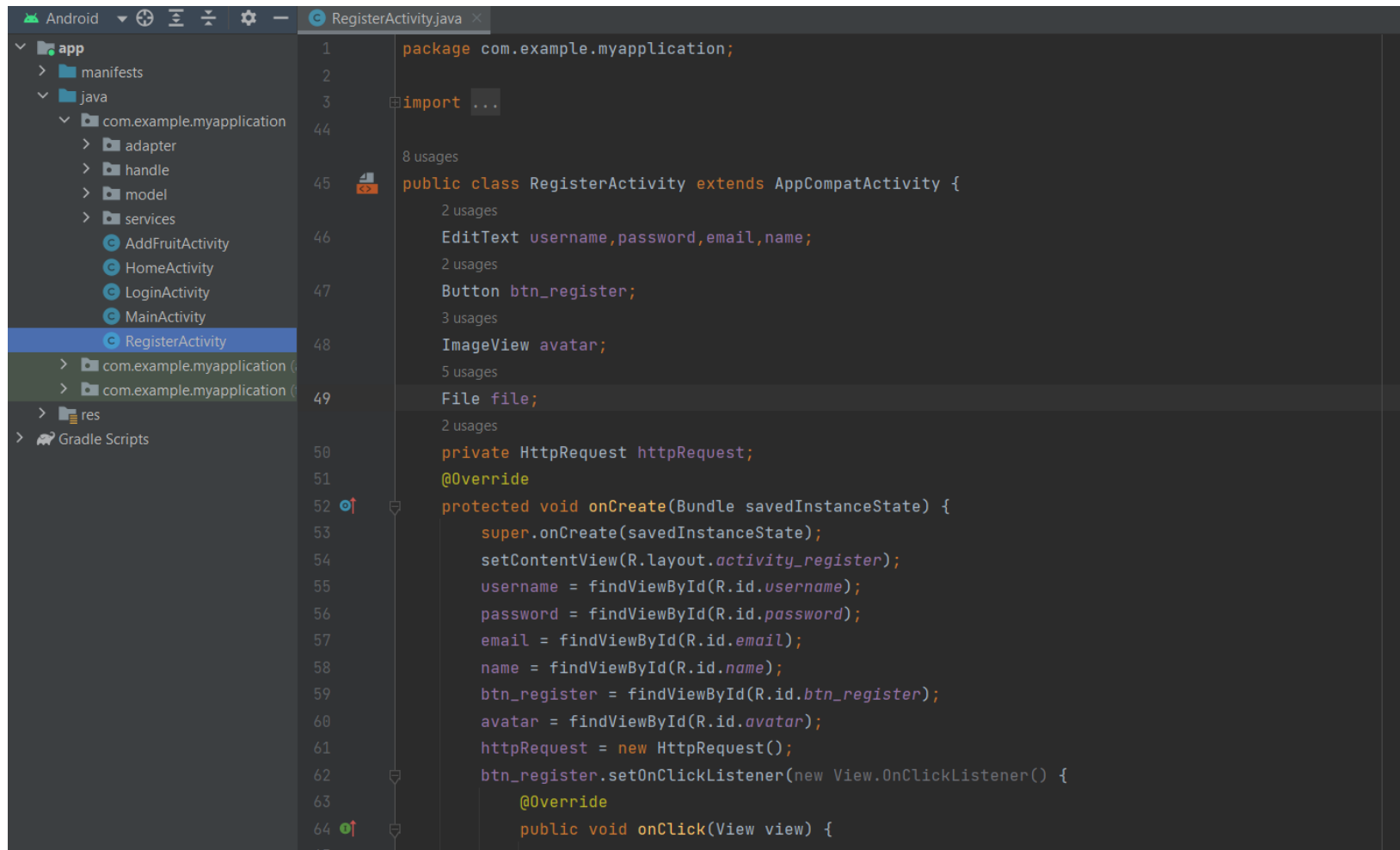
Memory

❑ Cấp quyền đọc, ghi dữ liệu trong bộ nhớ

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
</manifest>
```

□ Tạo Activity Register



```
1 package com.example.myapplication;
2
3 import ...
4
44 8 usages
45 public class RegisterActivity extends AppCompatActivity {
46     EditText username,password,email,name;
47     Button btn_register;
48     ImageView avatar;
49     File file;
50     private HttpRequest httpRequest;
51     @Override
52     protected void onCreate(Bundle savedInstanceState) {
53         super.onCreate(savedInstanceState);
54         setContentView(R.layout.activity_register);
55         username = findViewById(R.id.username);
56         password = findViewById(R.id.password);
57         email = findViewById(R.id.email);
58         name = findViewById(R.id.name);
59         btn_register = findViewById(R.id.btn_register);
60         avatar = findViewById(R.id.avatar);
61         httpRequest = new HttpRequest();
62         btn_register.setOnClickListener(new View.OnClickListener() {
63             @Override
64             public void onClick(View view) {
```

❑ Bắt sự kiện khi chọn hình



```
74     requestBody = RequestBody.create(MediaType.parse("image/*"), file);
75     multipartBody = MultipartBody.Part.createFormData("avatar", file.getName(), requestBody);
76     // "avatar" là cùng tên với key trong mutipart
77 } else {
78     multipartBody = null;
79 }
80 httpRequest.callAPI().register(_username, _password, _email, _name, multipartBody).enqueue(responseUser);
81 }
82 });
83
84 avatar.setOnClickListener(new View.OnClickListener() {
85     @Override
86     public void onClick(View view) { chooseImage(); }
87 });
88
89
90 }
```


❑ Xử lý chọn hình

```
RegisterActivity.java X
112 //Hàm chọn hình
113 1 usage
114 private void chooseImage() {
115     if (ContextCompat.checkSelfPermission(context: this,
116         android.Manifest.permission.READ_EXTERNAL_STORAGE) == PackageManager.PERMISSION_GRANTED) {
117         Intent intent = new Intent();
118         intent.setType("image/*"); // if you want to you can use pdf/gif/video
119         intent.setAction(Intent.ACTION_GET_CONTENT);
120         getImage.launch(intent);
121     } else {
122         ActivityCompat.requestPermissions(activity: this, new String[]{android.Manifest.permission.READ_EXTERNAL_STORAGE}, requestCode: 1);
123     }
124 }
125 //Activity result sau khi lấy hình
126 1 usage
127 ActivityResultLauncher<Intent> getImage = registerForActivityResult(new ActivityResultContracts.StartActivityForResult(),
128     new ActivityResultCallback<ActivityResult>() {
129         @Override
130         public void onActivityResult(ActivityResult result) {...}
131     });
132 }
```

❑ Xử lý chọn hình (tiếp theo)

```
RegisterActivity.java
114         if (ContextCompat.checkSelfPermission( context: this,
115             android.Manifest.permission.READ_EXTERNAL_STORAGE) == PackageManager.PERMISSION_GRANTED) {...} else {
121             ActivityCompat.requestPermissions( activity: this, new String[]{android.Manifest.permission.READ_EXTERNAL_STORAGE}, requestCode: 1);
122         }
123     }
124     //Activity result sau khi lấy hình
125     1 usage
126    ActivityResultLauncher<Intent> getImage = registerForActivityResult(new ActivityResultContracts.StartActivityForResult(),
127         new ActivityResultCallback<ActivityResult>() {
128         @Override
129         public void onActivityResult(ActivityResult result) {
130             if (result.getResultCode() == Activity.RESULT_OK) {
131                 // There are no request codes
132                 Intent data = result.getData();
133                 Uri imagePath = data.getData();
134                 //File avatar biến global
135                 file = createFileFromUri(imagePath, name: "avatar");
136                 //Glide để load hình
137                 Glide.with( activity: RegisterActivity.this) RequestManager
138                     .load(file) // load file hình
139                     .thumbnail(Glide.with( activity: RegisterActivity.this).load(R.mipmap.loading)) RequestBuilder<Drawable>
140                     .centerCrop() //center cắt ảnh
141                     .circleCrop() //bo tròn hình
142                     .diskCacheStrategy(DiskCacheStrategy.NONE) //clean cache
143                     .skipMemoryCache(true)
144                     .into(avatar);
145             }
146         }
147     };
```

❑ Tạo file hình từ uri

```
RegisterActivity.java x
146     });
147
148     //Hàm tạo file hình từ Uri
149     @ 1 usage
150     private File createFileFromUri(Uri path,String name)
151     {
152         File _file = new File(RegisterActivity.this.getCacheDir(), child: name+".png");
153         try {
154             InputStream in = RegisterActivity.this.getContentResolver().openInputStream(path);
155             OutputStream out = new FileOutputStream(_file);
156             byte[] buf = new byte[1024];
157             int len;
158             while((len=in.read(buf))>0){
159                 out.write(buf, off: 0,len);
160             }
161             out.close();
162             in.close();
163             return _file;
164         } catch (FileNotFoundException e) {
165             e.printStackTrace();
166         } catch (IOException e) {
167             e.printStackTrace();
168         }
169         return null;
170     }
```

□ Gọi API

```
RegisterActivity.java x
61  httprequest = new httprequest(),
62  btn_register.setOnClickListener(new View.OnClickListener() {
63      @Override
64      public void onClick(View view) {
65
66          //Sử dụng RequestBody
67          RequestBody _username = RequestBody.create(MediaType.parse("multipart/form-data"),username.getText().toString());
68          RequestBody _password = RequestBody.create(MediaType.parse("multipart/form-data"),password.getText().toString());
69          RequestBody _email = RequestBody.create(MediaType.parse("multipart/form-data"),email.getText().toString());
70          RequestBody _name = RequestBody.create(MediaType.parse("multipart/form-data"),name.getText().toString());
71          MultipartBody.Part multipartBody;
72          if(file != null)
73          {
74              RequestBody requestFile = RequestBody.create(MediaType.parse("image/*"), file);
75              multipartBody = MultipartBody.Part.createFormData("avatar",file.getName(),requestFile);
76              // "avatar" là cùng tên với key trong mutipart
77          }else {
78              multipartBody = null;
79          }
80          httpRequest.callAPI().register(_username,_password,_email,_name,multipartBody).enqueue(responseUser);
81      }
82  });
83
```

□ Tạo Callback

```
RegisterActivity.java x
91 Callback<Response<User>> responseUser = new Callback<Response<User>>() {
92     @Override
93     public void onResponse(Call<Response<User>> call, retrofit2.Response<Response<User>> response) {
94         if(response.isSuccessful())
95         {
96             //check status code
97             if(response.body().getStatus() == 200)
98             {
99                 Toast.makeText(context: RegisterActivity.this, text: "Đăng ký thành công", Toast.LENGTH_SHORT).show();
100                 //Sau khi đăng ký thành công trở về trang đăng nhập
101                 //finish();
102             }
103         }
104     }
105     @Override
106     public void onFailure(Call<Response<User>> call, Throwable t) {
107         Log.d(tag: ">>> GetListDistributor", msg: "onFailure: " + t.getMessage());
108     }
109 };
110
```

 Xem kết quả demo



6:19 3G

My Application

Đăng ký



Username:

fpl

Password:

...

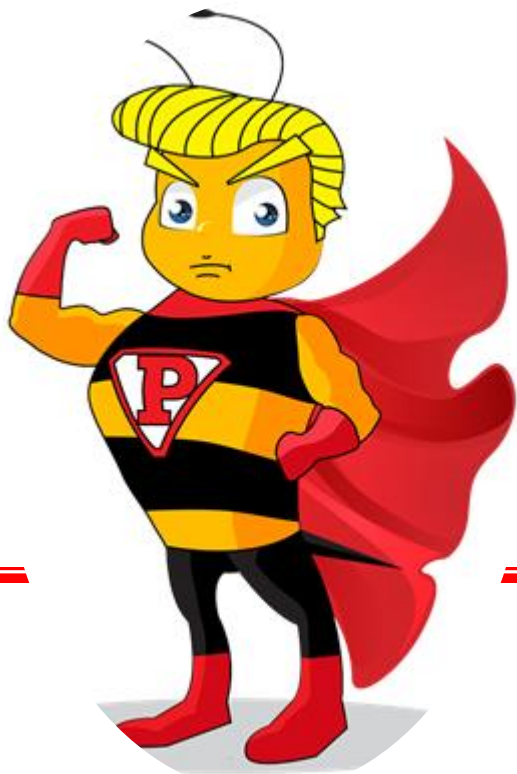
Email:

myemail@gmail.com

Name:

Android

REGISTER



CALL API ĐĂNG NHẬP

...

❑ Trong ApiService thêm phương thức POST - Login

```
1 usage  
@POST("login")  
Call<Response<User>> login(@Body User user);
```


❑ Trong model Reponse them 2 trường token và refreshToken

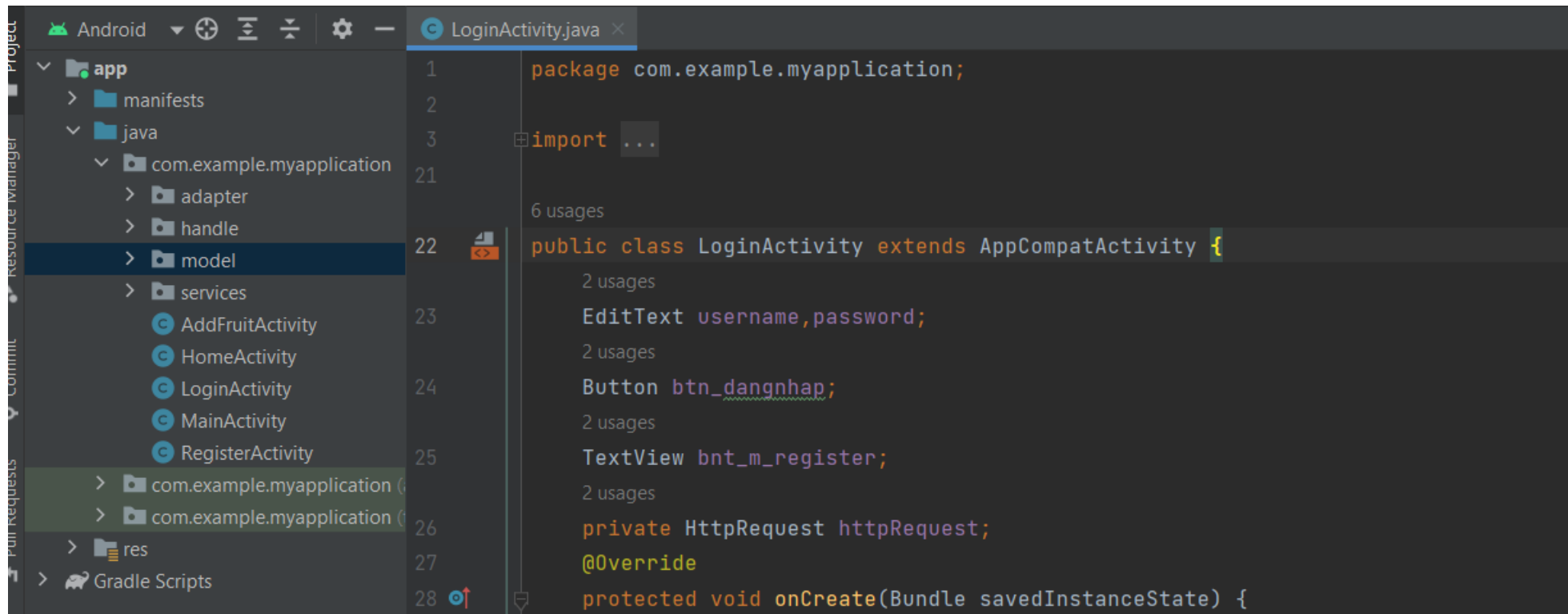
```
50 usages
public class Response<T> {
    3 usages
    private int status;
    3 usages
    private String messenger;
    //T là kiểu Generic
    3 usages
    private T data;
    3 usages
    private String token;
    3 usages
    private String refreshToken;

    no usages
    public Response(int status, String messenger, T data, String token, String refreshToken) {
        this.status = status;
        this.messenger = messenger;
        this.data = data;
        this.token = token;
        this.refreshToken = refreshToken;
    }

    1 usage
    public String getToken() { return token; }

    no usages
    public void setToken(String token) { this.token = token; }
```

❑ Tạo Activity Login



```
1 package com.example.myapplication;
2
3 import ...
4
5 6 usages
6 public class LoginActivity extends AppCompatActivity {
7     2 usages
8     EditText username,password;
9     2 usages
10    Button btn_dangnhap;
11    2 usages
12    TextView bnt_m_register;
13    2 usages
14    private HttpRequest httpRequest;
15
16    @Override
17    protected void onCreate(Bundle savedInstanceState) {
```

Call API login

```
com / example / myapplication / LoginActivity / ResponseUser
LoginActivity.java x
40      }
41    });
42    btn_dangnhap.setOnClickListener(new View.OnClickListener() {
43      @Override
44      public void onClick(View view) {
45        User user = new User();
46        String _username = username.getText().toString();
47        String _password = password.getText().toString();
48        user.setUsername(_username);
49        user.setPassword(_password);
50        httpRequest.callAPI().login(user).enqueue(responseUser);
51      }
52    });
53  }
```

□ Tạo Callback

```
LoginActivity.java
52     });
53     }
54     Callback<Response<User>> responseUser = new Callback<Response<User>>() {
55         @Override
56         public void onResponse(Call<Response<User>> call, retrofit2.Response<Response<User>> response) {
57             if(response.isSuccessful())
58             {
59                 //check status code
60                 if(response.body().getStatus() == 200)
61                 {
62                     Toast.makeText(context: LoginActivity.this, text: "Đăng nhập thành công", Toast.LENGTH_SHORT).show();
63                     //Lưu token, lưu device token, id
64                     SharedPreferences sharedPreferences = getSharedPreferences( name: "INFO",MODE_PRIVATE);
65                     SharedPreferences.Editor editor = sharedPreferences.edit();
66                     editor.putString( s: "token",response.body().getToken());
67                     editor.putString( s: "refreshToken",response.body().getRefreshToken());
68                     editor.putString( s: "id",response.body().getData().get_id());
69                     editor.apply();
70                     //Sau khi chuyển sang màn hình chính
71                     startActivity(new Intent( packageContext: LoginActivity.this,HomeActivity.class));
72                 }
73             }
74         }
75         @Override
76         public void onFailure(Call<Response<User>> call, Throwable t) {
77             Log.d( tag: ">>> GetListDistributor", msg: "onFailure: " + t.getMessage());
78         }
79     };
```

DEMO



FPT Education

FPT POLYTECHNIC

Thank you