

## HƯỚNG DẪN TẠO DỰ ÁN VỚI RESTful API TRONG ANDROID

```
example.json
                                                                                                 💗 📶 🖺 6:17 ra
                                                                              Json Example
  "fish_img":"1.jpg",
                                                                              Indian Mackerel
                                                                                    Size: Medium
Category: Marine Fish
   "fish_name": "Indian Mackerel",
   "cat_name": "Marine Fish",
                                                                               Manthal Repti
                                                                                                     Rs 200/Kg
   "size_name":"Medium",
                                                                                    Size: Small
Category: Marine Fish
   "price": "100"
},
                                                                              Baby Sole Fish
                                                                                                    8s. 600\Kg
                                                                                    Size, Small
Category: Marine Fish
  "fish_img": "2.jpg",
                                                                              Silver Pomfret
                                                                                                    Ra. 300\Kg
   "fish_name": "Manthal Repti",
                                                                                    Size: Large
Category: Manne Fish
   "cat_name": "Marine Fish",
   "size_name": "Small",
                                                                                                    Rs. 800\Kģ
   "price":"200"
                                                                                    Size: Small
Category: Shell Fish
},
                                                                                            0
                                                                                                       1
```

## **Resource Software Solution**

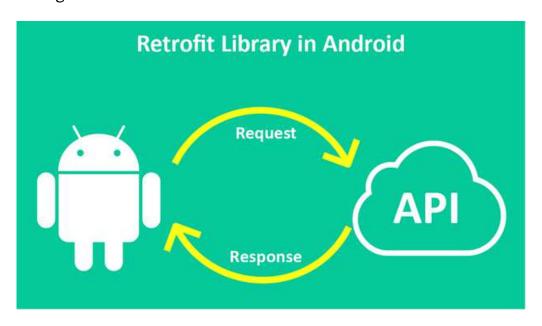


## Mục lục

1.	Giới thiệu	3
2.	Các bước thực hiện tương tác Restful API trong Android	3
3.	Ví dụ mẫu	<b>.</b> .3
4.	Tự làm	. 11

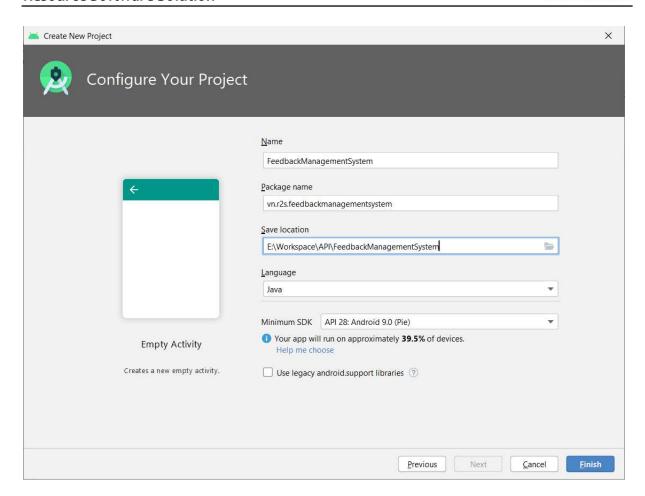


- 1. Giới thiệu
- Retrofit là thư viên hỗ trợ lập trình viên Android tạo ra những ứng dụng tương tác với API



- 2. Các bước thực hiện tương tác Restful API trong Android
  - Thêm dependencies
  - Tạo danh sách Model tương ứng với số lượng Entity
  - Tạo interface cung cấp các annotations cho từng phương thức GET, POST,
     PUT và DELETE
  - Tạo thể hiện của Retrofit để gửi yêu cầu (request) đến API
  - Thêm quyền truy cập Internet cho ứng dụng trong AndroidManifest.xml
  - Tương tác API
- 3. Ví dụ mẫu
  - a) Mở Android Studio và tạo một Android Project
  - Tên Project tên FeedbackManagementSystem





## b) Thêm dependencies

- Mở build.gradle (Module: app)

```
You can use the Project Structure dialog to view and edit your project configuration
                                                                                                                                                                     Open (Ctrl+Alt
 📭 арр
   manifests
                                                              versionName "1.0"
   ▼ Image vn.r2s feedbackmanagementsystem
                                                              {\tt testInstrumentationRunner~"androidx.test.runner.AndroidJUnitRunner"}
    vn.r2s.feedbackmanagementsystem (androidTest 14
   res
                                                         buildTypes {
                                                              release {
    build.gradle (Project: FeedbackManagementSyst
                                                                   minifyEnabled false
    w build.gradle (Module: app)
                                                                   proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
    gradle-wrapper.properties (Gradle Version)
    proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
                                            21
                                                         }
    settings.gradle (Project Settings)
                                                     1
    all local.properties (SDK Location)
                                           24
                                            25
                                                          implementation fileTree(dir: "libs", include: ["*.jar"])
                                           26
27
                                                         implementation 'androidx.appcompat:appcompat:1.1.0'
                                                         implementation \ 'and roidx. constraint layout: constraint layout: {\tt 1.1.3'}
                                           28
29
                                                         testImplementation 'junit:junit:4.12'
                                                         androidTestImplementation 'androidx.test.ext:junit:1.1.1'
                                                         androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
```

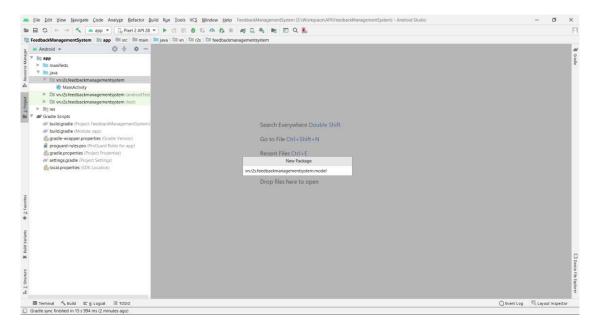
- Thêm

```
//Retrofit
implementation 'com.squareup.retrofit2:retrofit:2.5.0'
implementation 'com.squareup.retrofit2:converter-gson:2.4.0'
```





- Chọn Sync Now
- c) Tạo Model
- Tạo mới package tên vn.r2s.feedbackmanagementsystem.model



- Chuột phải model -> chọn New -> chọn Java Class



- Nhập User -> nhấn Enter
- Nhập code cho User



```
© User.java ×
        * User model
        * @author kyle
 5
 6
       public class User {
 8
           private Long id;
 9
           private String firstName;
10
           private String lastName;
11
           private String email;
12
13
            * Constructor
14
            * @param id
15
             * @param firstName
16
17
            * @param LastName
18
            * @param email
19
20
           public User(Long id, String firstName, String lastName, String email) {
21
               this.id = id;
22
               this.firstName = firstName;
               this.lastName = lastName;
23
               this.email = email;
24
25
           }
26
27
           public Long getId() {
28
                return id;
29
```

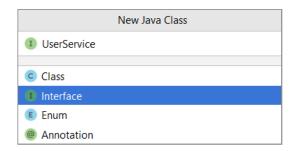
```
30
            public void setId(Long id) {
               this.id = id;
32
33
34
35
            public String getFirstName() {
               return firstName;
36
37
38
            public void setFirstName(String firstName) {
39
               this.firstName = firstName;
40
41
            }
42
43
            public String getLastName() {
               return lastName;
44
45
46
            public void setLastName(String lastName) {
47
               this.lastName = lastName;
48
49
            public String getEmail() {
               return email;
            }
54
            public void setEmail(String email) {
56
               this.email = email;
57
```



- d) Tạo interface cung cấp các annotations cho từng phương thức GET, POST, PUT và DELETE
- Tạo mới package tên vn.r2s.feedbackmanagementsystem.api



- Chuột phải **api** -> chọn New -> chọn Java Class -> nhập **UserService** -> chọn Interface -> nhấn Enter



- Nhập code cho UserService

```
14
        * User service
15
        * @author kyle
16
       */
17
       public interface UserService {
18
19
           @GET("api/v1/users")
20
           Call<ArrayList<User>> getUsers();
21
22
           @GET("api/v1/users")
23
           Call<User> getUserById(@Query("id") long id);
24
25
           @POST("api/v1/users")
26
           Call<User> postUser(@Body User user);
27
28
           @PUT("api/v1/users")
29
           Call<Void> updateUser(@Query("id") long id, @Body User user);
30
31
           @DELETE("api/v1/user")
32
           Call<Void> delete(@Query("id") long id);
33
34
35
```



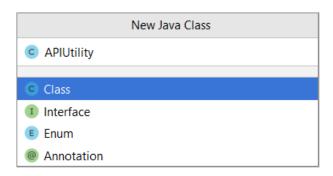
- e) Tạo thể hiện của Retrofit để gửi yêu cầu (request) đến API
- Chuột phải api -> chọn New -> chọn Java Class -> nhập APIClient



- Nhập code cho **APIClient** 

```
* API Client class
7
          @author kyle
8
9
       public class APIClient {
10
11
           private static Retrofit retrofit = null;
12
13
           public static Retrofit getClient(String baseURL) {
14
                retrofit = new Retrofit.Builder()
15
                        .baseUrl(baseURL)
16
                        .addConverterFactory(GsonConverterFactory.create())
17
                        .build();
18
                return retrofit;
19
           }
20
21
22
```

- Chuột phải api -> chọn New -> chọn Java Class -> nhập APIUtility



- Nhập code cho APIUtility



```
3
        * API utility class
4
        * @author kyle
        */
6
       public class APIUtility {
8
           private static String baseURL="http://10.0.2.2:8080/Feedback-Restful-API/";
9
10
11
            * Create an user service instance
            * @return UserService
13
14
15 @
           public static UserService getUserService() {
16
               return APIClient.getClient(baseURL).create(UserService.class);
17
18
19
```

- f) Thêm quyền truy cập Internet cho ứng dụng trong AndroidManifest.xml
- Mở AndroidManifest.xml

```
package="vn.r2s.feedbackmanagementsystem">
4
5
           <uses-permission android:name="android.permission.INTERNET" />
7
           <application</a>
                android:allowBackup="true"
8
                android:icon="@mipmap/ic_launcher"
                android:label="FeedbackManagementSystem"
                android:roundIcon="@mipmap/ic_launcher_round"
11
               android:supportsRtl="true"
12
13
                android:usesCleartextTraffic="true"
                android:theme="@style/AppTheme">
14
                <activity</a>
15
                    android: name=".HomeActivity"
16
                    android:label="HomeActivity"
17
                    android:theme="@style/AppTheme.NoActionBar">
18
19
                    <intent-filter>
                        <action android:name="android.intent.action.MAIN" />
20
21
                        <category android:name="android.intent.category.LAUNCHER" />
22
23
                    </intent-filter>
24
                </activity>
                <activity android:name=".MainActivity">
25
26
27
                </activity>
           </application>
28
```



- Thêm quyền truy cập Internet

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

- Fix Cleartext Traffic

```
android:usesCleartextTraffic="true"
```

```
package="vn.r2s.feedbackmanagementsystem">
5
           <uses-permission android:name="android.permission.INTERNET" />
7
           <application</a>
               android:allowBackup="true"
8
               android:icon="@mipmap/ic_launcher"
               android:label="FeedbackManagementSystem"
               android:roundIcon="@mipmap/ic launcher round"
11
               android:supportsRtl="true"
12
               android:usesCleartextTraffic="true"
               android:theme="@style/AppTheme">
14
               <activity
                   android:name=".HomeActivity"
                   android:label="HomeActivity"
17
                   android:theme="@style/AppTheme.NoActionBar">
18
                    <intent-filter>
19
                        <action android:name="android.intent.action.MAIN" />
20
21
                        <category android:name="android.intent.category.LAUNCHER" />
22
                    </intent-filter>
23
               </activity>
24
               <activity android:name=".MainActivity">
25
26
               </activity>
27
           </application>
28
```

- g) Tương tác API
- Tao thể hiện của UserService
- Gọi các phương thức đã khai báo trong interface



```
18
       public class UserViewModel extends AndroidViewModel {
19
20
           // This is the data that we will fetch asynchronously
           private MutableLiveData<ArrayList<User>> userList;
           private UserService userService;
           public UserViewModel(Application application) {
24
               super(application);
25
                userService = APIUtility.getUserService();
29
30
           // We will call this method to get the data
           public MutableLiveData<ArrayList<User>>> getUserList() {
31
                if (userList == null) {
                    userList = new MutableLiveData<ArrayList<User>>();
                    loadUsers();
34
36
                // Finally we will return the list
                return userList;
38
39
40
41
             * Get all users
42
43
44
              @return list of user
45
            private void loadUsers() {
                Call<ArrayList<User>> call = userService.getUsers();
                call.enqueue(new Callback<ArrayList<User>>() {
48
19
                    @Override
                    public void onResponse(Call<ArrayList<User>> call, Response<ArrayList<User>> response
50 0
                        // Finally we are setting the list to our MutableLiveData
51
                        userList.setValue(response.body());
52
                    }
53
54
56 🐠
                    public void onFailure(Call<ArrayList<User>> call, Throwable t) {
                        Toast.makeText(getApplication().getBaseContext()
58
                                , text: "UserViewModel: " + t.getMessage(), Toast.LENGTH_LONG).show();
59
60
                });
     }
62
```

- h) Build và run
- 4. Tự làm

Bổ sung tính năng thêm, xóa và cập nhật User