



FPT POLYTECHNIC



android

www.poly.edu.vn

LẬP TRÌNH ANDROID VỚI RESTAPI

LÀM VIỆC VỚI API BÊN THỨ BA

- ☐ Giới thiệu API bên thứ ba
- ☐ Làm việc với API bên thứ ba (GHN)

MỤC TIÊU

- ◎ GIỚI THIỆU API BÊN THỨ BA
- ◎ LÀM VIỆC VỚI API BÊN THỨ BA (GHN)

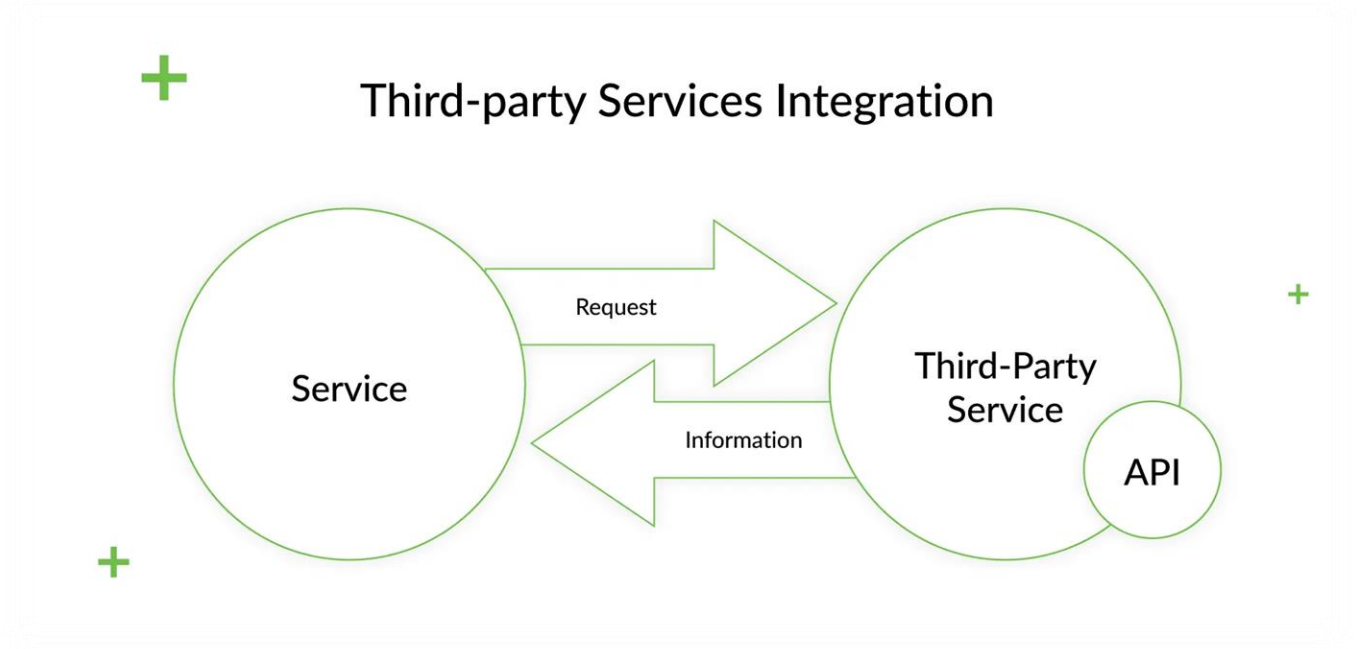




API BÊN THỨ BA

...

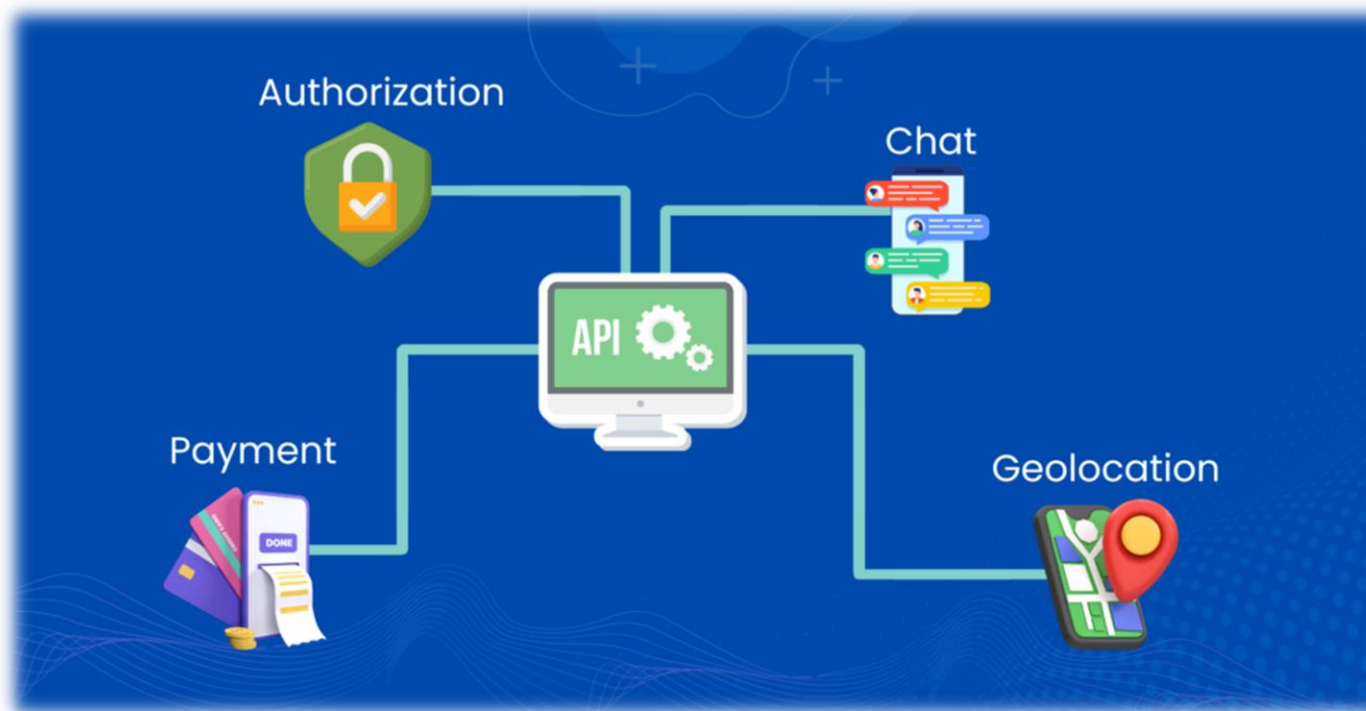
- API bên thứ ba (**Third-Party API**) là một loại giao diện lập trình ứng dụng (API) được cung cấp bởi một tổ chức hoặc cá nhân không liên quan đến tổ chức hoặc cá nhân sử dụng API đó. API này cho phép các nhà phát triển sử dụng các dịch vụ, chức năng hoặc dữ liệu của người cung cấp API bên thứ ba trong ứng dụng hoặc dự án của họ.

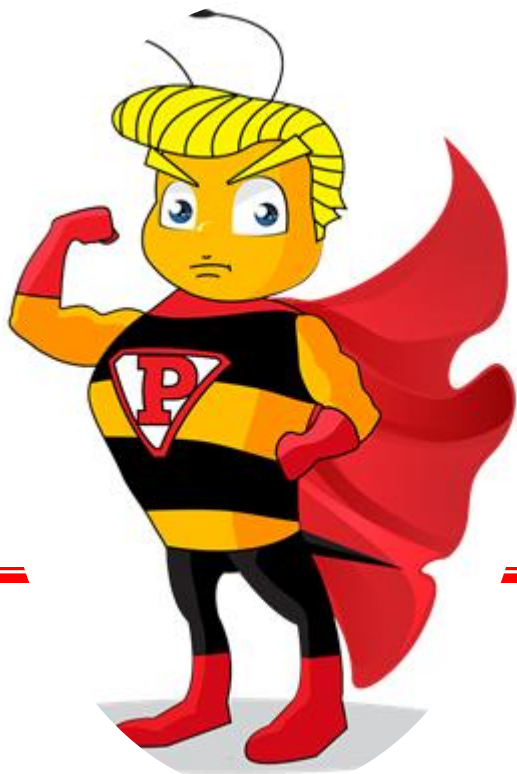


- Ví dụ phổ biến về API bên thứ ba bao gồm API của các công ty lớn như Facebook, Twitter, Google Maps, PayPal, và nhiều dịch vụ khác. Các nhà phát triển có thể sử dụng API này để tích hợp các tính năng hoặc dữ liệu từ các dịch vụ này vào ứng dụng của họ, giúp cung cấp các trải nghiệm mở rộng cho người dùng.



- ❑ API bên thứ ba có thể cung cấp các dịch vụ đa dạng, chẳng hạn như xác thực người dùng, truy cập dữ liệu, thanh toán trực tuyến, chia sẻ nội dung trên mạng xã hội, và nhiều tính năng khác.





LÀM VIỆC VỚI API BÊN THỨ 3 (GHN) LẤY THÔNG TIN ĐỊA CHỈ

...

- ❑ Truy cập trang web <https://5sao.ghn.dev/>
- ❑ Đăng ký tài khoản và xem hướng dẫn <https://api.ghn.vn/home/docs/detail?id=49>

The screenshot displays the GHN API documentation website. The top navigation bar includes 'Home', 'Document' (highlighted), 'FAQ', and 'Version'. A search bar labeled 'Search Document' is on the right. The left sidebar lists various API endpoints: 'Get Token', 'Create Account and Get Token, Client_ID, ShopID', 'Order', 'Calculate Fee', 'Store', 'Address', 'Webhook', 'Ticket', 'More Information', and 'AFFILIATE'. The main content area is titled 'Create Account and Get Token, Client_ID, ShopID' and contains two sections: 'I. Create Account in Staging and Production enviroment' and 'II. Get Token and Create Shop's address'. Below these sections is a browser window showing the login page of the GHN API. The login page has a header with the GHN logo and a navigation menu. The main content area is titled 'Đăng nhập' (Login) and includes a warning message: 'Buổi tối an lành bên gia đình bạn nhất gác công việc lại nào.' Below this is a login form with fields for 'Số điện thoại hoặc Email' (Phone number or Email) and 'Mật khẩu' (Password). A 'Đăng nhập' (Login) button is at the bottom of the form. A footer message says 'Bạn chưa có tài khoản? Đăng ký ngay!' (You don't have an account? Register now!).

❑ Tạo model dữ liệu thành phố / tỉnh

(<https://api.ghn.vn/home/docs/detail?id=60>)

```
public class Province {  
    private int ProvinceID;  
    private String ProvinceName;  
  
    public Province(int provinceID, String provinceName) {  
        ProvinceID = provinceID;  
        ProvinceName = provinceName;  
    }  
  
    public int getProvinceID() {  
        return ProvinceID;  
    }  
  
    public void setProvinceID(int provinceID) {  
        ProvinceID = provinceID;  
    }  
  
    public String getProvinceName() {  
        return ProvinceName;  
    }  
  
    public void setProvinceName(String provinceName) {  
        ProvinceName = provinceName;  
    }  
}
```

You, Today • Uncommitted changes

Tạo model dữ liệu Quận/Huyện (<https://api.ghn.vn/home/docs/detail?id=78>)

```
public class District {  
    private int DistrictID;    You, Today · Uncommitted changes  
    private int ProvinceID;  
    private String DistrictName;  
  
    public District(int districtID, int provinceID, String districtName) {  
        DistrictID = districtID;  
        ProvinceID = provinceID;  
        DistrictName = districtName;  
    }  
  
    public int getDistrictID() {  
        return DistrictID;  
    }  
  
    public void setDistrictID(int districtID) {  
        DistrictID = districtID;  
    }  
  
    public int getProvinceID() {  
        return ProvinceID;  
    }  
  
    public void setProvinceID(int provinceID) {  
        ProvinceID = provinceID;  
    }  
  
    public String getDistrictName() {  
        return DistrictName;  
    }  
  
    public void setDistrictName(String districtName) {  
        DistrictName = districtName;  
    }  
}
```

❑ Tạo model Request lấy danh sách Quận Huyện

```
public class DistrictRequest{  
    private int province_id;  
  
    public DistrictRequest(int province_id) {  
        this.province_id = province_id;  
    }  
  
    public int getProvince_id() {  
        return province_id;  
    }  
  
    public void setProvince_id(int province_id) {  
        this.province_id = province_id;  
    }  
}
```

Tạo model dữ liệu Phường/Xã (<https://api.ghn.vn/home/docs/detail?id=61>)

```
public class Ward {  
    private String WardCode;  
    private int DistrictID;  
    private String WardName;  
  
    public Ward(String wardCode, int districtID, String wardName) {  
        WardCode = wardCode;  
        DistrictID = districtID;  
        WardName = wardName;  
    }  
  
    public String getWardCode() {  
        return WardCode;  
    }  
  
    public void setWardCode(String wardCode) {  
        WardCode = wardCode;  
    }  
  
    public int getDistrictID() {  
        return DistrictID;  
    }  
  
    public void setDistrictID(int districtID) {  
        DistrictID = districtID;  
    }  
  
    public String getWardName() {  
        return WardName;  
    }  
  
    public void setWardName(String wardName) {  
        WardName = wardName;  
    }  
}
```

Model Response GHN

```
public class ResponseGHN<T> {  
    3 usages  
    private int code;  
    3 usages  
    private String message;  
    3 usages  
    private T data;  
  
    no usages  
    public ResponseGHN(int code, String message, T data) {  
        this.code = code;  
        this.message = message;  
        this.data = data;  
    }  
  
    no usages  
    public int getCode() { return code; }  
  
    no usages  
    public void setCode(int code) { this.code = code; }  
  
    no usages  
    public String getMessage() { return message; }  
  
    no usages  
    public void setMessage(String message) { this.message = message; }  
  
    public T getData() { return data; }
```

❑ Tạo GHN Services và GHN Request (tại folder **Services**)

```
public interface GHNServices {  
    public static String GHN_URL = "https://dev-online-gateway.ghn.vn/";  
  
    //Get danh sách TP/Tỉnh  
    @GET("shipp/public-api/master-data/province")  
    Call<ResponseGHN<ArrayList<Province>>> getListProvince();  
    //Get danh sách Quận/Huyện  
    @POST("shipp/public-api/master-data/district")  
    Call<ResponseGHN<ArrayList<District>>> getListDistrict(@Body DistrictRequest districtRequest);  
    //Get danh sách Phường/Xã  
    @GET("shipp/public-api/master-data/ward")  
    Call<ResponseGHN<ArrayList<Ward>>> getListWard(@Query("district_id") int district_id);  
}
```

You 14 minutes ago • Uncommitted changes

❑ Tạo GHN Services và GHN Request (tại folder **Services**) *(tiếp theo)*

```
public class GHNRequest {  
    // Shop ID của tài khoản GHN  
    public final static String SHOPID = "188698";  
    //Token của tài khoản GHN  
    public final static String TokenGHN = "39fa33f2-4b4b-11ee-a6e6-e60958111f48";  
    private GHNServices ghnRequestInterface;  
  
    public GHNRequest() {  
        //Tạo intercept  
        OkHttpClient.Builder httpClient = new OkHttpClient.Builder();  
        httpClient.addInterceptor(new Interceptor() {  
            @NonNull  
            @Override  
            public Response intercept(Chain chain) throws IOException {  
                Request request = chain.request().newBuilder()  
                    .addHeader( name: "ShopId", SHOPID)  
                    .addHeader( name: "Token", TokenGHN)  
                    .build();  
                return chain.proceed(request);  
            }  
        });  
        //Create Retrofit  
        ghnRequestInterface = new Retrofit.Builder()  
            .baseUrl(GHN_URL)  
            .addConverterFactory(GsonConverterFactory.create())  
            .client(httpClient.build())  
            .build().create(GHNServices.class);  
    }  
  
    public GHNServices callAPI() {  
        //Get Retrofit  
        return ghnRequestInterface;  
    }  
}
```


- ❑ Mua hàng (Tại giao diện sản phẩm thêm nút mua, khi nhấn mua sẽ truyền sang giao diện chọn vị trí giao hàng)

```
        .into(holder.image);
holder.buy.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Bundle bundle = new Bundle();
        bundle.putSerializable("item", ds.get(holder.getAdapterPosition()));
        Intent intent = new Intent(context, LocationActivity.class);
        intent.putExtras(bundle);
        (context).startActivity(intent);
    }
});
}
```

Chọn vị trí giao hàng

```
public class LocationActivity extends AppCompatActivity {  
    GHNRequest ghnRequest;  
    Spinner sp_province, sp_district, sp_ward;  
    EditText edt_location;  
    Button btn_next;  
  
    //Lưu giữ liều chọn You, 19 minutes ago • Uncommitted changes  
    private String WardCode;  
    private int DistrictID;  
    private int ProvinceID;  
    //Adapter Spinner  
    private Adapter_Item_Ward_Select_GHN adapter_item_ward_select_ghn;  
    private Adapter_Item_District_Select_GHN adapter_item_district_select_ghn;  
    private Adapter_Item_Province_Select_GHN adapter_item_province_select_ghn;  
}
```

❑ Chọn vị trí giao hàng *(tiếp theo)*

```
private Adapter_Item_Province_Select_GHN adapter_item_province_select_ghn;  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_location);  
    ghnRequest = new GHNRequest();  
    sp_province = findViewById(R.id.sp_province);  
    sp_district = findViewById(R.id.sp_district);  
    sp_ward = findViewById(R.id.sp_ward);  
    edt_location = findViewById(R.id.edt_location);  
    btn_next = findViewById(R.id.btn_next);  
    //Gọi API lấy danh sách TP/Tỉnh đầu tiên  
    ghnRequest.callAPI().getListProvince().enqueue(responseProvince);  
    //Lắng nghe sự kiện chọn You, Moments ago • Uncommitted changes  
    sp_province.setOnItemSelectedListener(onItemSelectedListener);  
    sp_district.setOnItemSelectedListener(onItemSelectedListener);  
    sp_ward.setOnItemSelectedListener(onItemSelectedListener);  
    sp_province.setSelection(0);  
}
```

Chọn vị trí giao hàng (*tiếp theo*)

```
AdapterView.OnItemSelectedListener onItemSelectedListener = new AdapterView.OnItemSelectedListener() {  
    @Override  
    public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {  
  
        switch (adapterView.getId())  
        {  
            case R.id.sp_province:  
                ProvinceID = ((Province) adapterView.getAdapter().getItem(i)).getProvinceID();  
                DistrictRequest districtRequest = new DistrictRequest(ProvinceID);  
                //Sau khi có API TP/Tỉnh lấy gọi API lấy danh sách Quận/Huyện  
                ghnRequest.callAPI().getListDistrict(districtRequest).enqueue(responseDistrict);  
                break;  
            case R.id.sp_district:  
                DistrictID = ((District) adapterView.getAdapter().getItem(i)).getDistrictID();  
                //Sau khi có API Quận/Huyện lấy gọi API lấy danh sách Phường/Xã  
                ghnRequest.callAPI().getListWard(DistrictID).enqueue(responseWard);  
                break;  
            case R.id.sp_ward:  
                WardCode = ((Ward) adapterView.getAdapter().getItem(i)).getWardCode();  
                break;  
            default:  
                break;  
        }  
    }  
}  
  
@Override  
public void onNothingSelected(AdapterView<?> adapterView) {  
  
}  
};
```

Chọn vị trí giao hàng *(tiếp theo)*

```
//Set view lên giao diện Spinner You, Moments ago • Uncommitted changes
private void SetDataSpinProvince(ArrayList<Province> ds){
    adapter_item_province_select_ghn = new Adapter_Item_Province_Select_GHN( context: this,ds);
    sp_province.setAdapter(adapter_item_province_select_ghn);
}
private void SetDataSpinDistrict(ArrayList<District> ds){
    adapter_item_district_select_ghn = new Adapter_Item_District_Select_GHN( context: this,ds);
    sp_district.setAdapter(adapter_item_district_select_ghn);
}
private void SetDataSpinWard(ArrayList<Ward> ds){
    adapter_item_ward_select_ghn = new Adapter_Item_Ward_Select_GHN( context: this,ds);
    sp_ward.setAdapter(adapter_item_ward_select_ghn );
}
}
```

Chọn vị trí giao hàng (tiếp theo)

```
Callback<ResponseGHN-ArrayList<Province>>> responseProvince = new Callback<ResponseGHN-ArrayList<Province>>>() {
    @Override
    public void onResponse(Call<ResponseGHN-ArrayList<Province>>> call, Response<ResponseGHN-ArrayList<Province>>> response) {
        if(response.isSuccessful()){
            if(response.body().getCode() == 200)
            {
                ArrayList<Province> ds = new ArrayList<> (response.body().getData());
                setDataSpinProvince(ds);
            }
        }
    }

    @Override
    public void onFailure(Call<ResponseGHN-ArrayList<Province>>> call, Throwable t) {

    }
};

Callback<ResponseGHN-ArrayList<District>>> responseDistrict = new Callback<ResponseGHN-ArrayList<District>>>() {
    @Override
    public void onResponse(Call<ResponseGHN-ArrayList<District>>> call, Response<ResponseGHN-ArrayList<District>>> response) {
        if(response.isSuccessful()){
            if(response.body().getCode() == 200)
            {
                ArrayList<District> ds = new ArrayList<> (response.body().getData());
                setDataSpinDistrict(ds);
            }
        }
    }

    @Override
    public void onFailure(Call<ResponseGHN-ArrayList<District>>> call, Throwable t) {

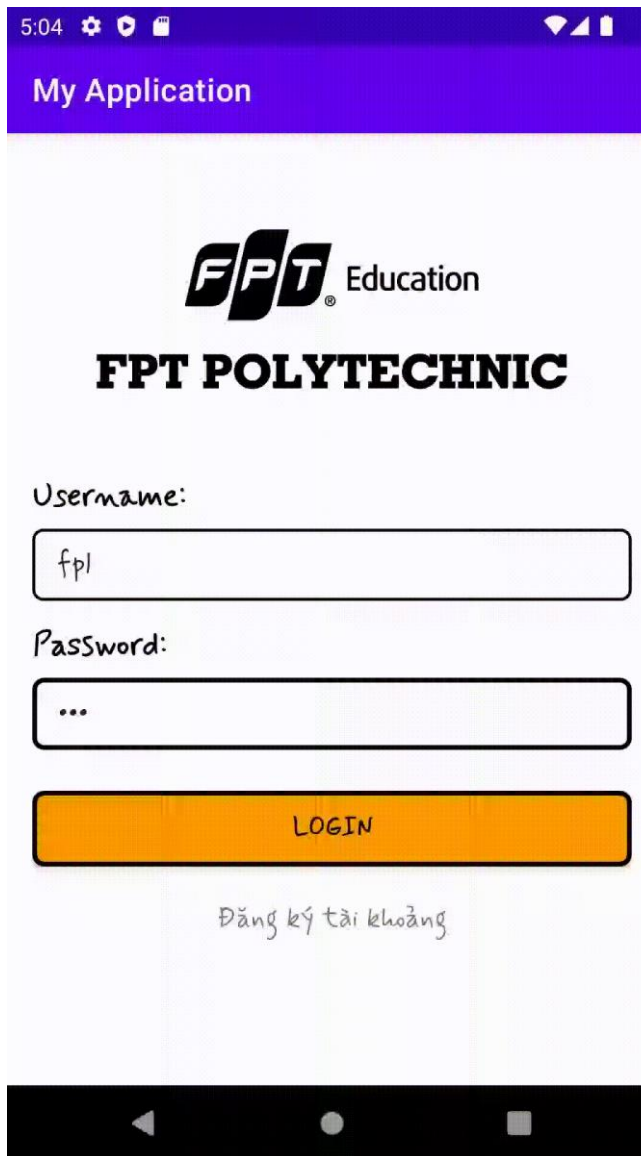
    }
};

Callback<ResponseGHN-ArrayList<Ward>>> responseWard = new Callback<ResponseGHN-ArrayList<Ward>>>() {
    @Override
    public void onResponse(Call<ResponseGHN-ArrayList<Ward>>> call, Response<ResponseGHN-ArrayList<Ward>>> response) {
        if(response.isSuccessful()){
            if(response.body().getCode() == 200)
            {
                ArrayList<Ward> ds = new ArrayList<> ();
                if(response.body().getData() == null) return;
                ds.addAll(response.body().getData());
                setDataSpinWard(ds);
            }
        }
    }

    @Override
    public void onFailure(Call<ResponseGHN-ArrayList<Ward>>> call, Throwable t) {

    }
};
```

 Xem video demo



5:04

My Application

FPT Education
FPT POLYTECHNIC

Username:

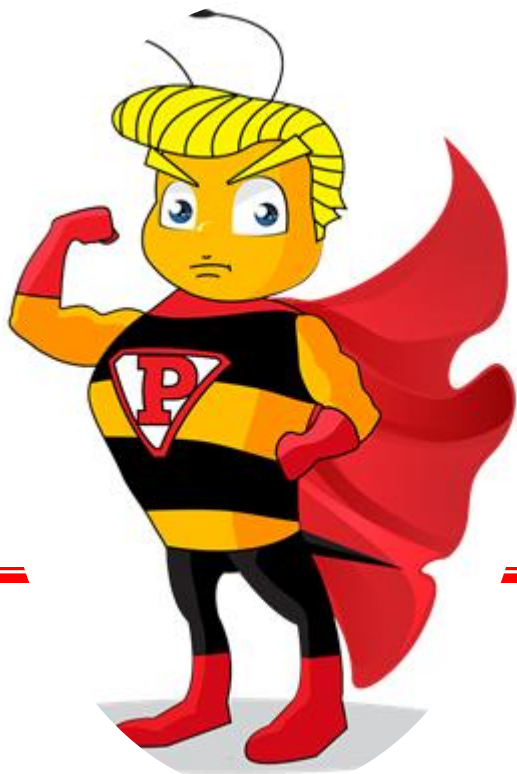
fpl

Password:

...

LOGIN

Đăng ký tài khoản



LÀM VIỆC VỚI API BÊN THỨ 3 (GHN) TẠO ĐƠN HÀNG

...

❑ Tạo model database

```
els > Js order.js > Orders > Id_user > ref  
const mongoose = require('mongoose');  
const Schema = mongoose.Schema;  
  
const Orders = new Schema({  
  order_code: {type: String},  
  id_user : {type: Schema.Types.ObjectId, ref: 'user'}  
},{  
  timestamps: true  
})  
  
module.exports = mongoose.model('Order', Orders)
```

❑ Viết API lưu thông tin đơn hàng vào database (**api.js**)

```
const Orders = require('../models/order')

router.post('/add-order', async (req, res) => {
  try {
    const data = req.body;
    const newOrder = new Orders({
      order_code: data.order_code,
      id_user : data.id_user
    })
    const result = await newOrder.save(); //Thêm vào database
    if(result)
    {
      // Nếu thêm thành công result !null trả về dữ liệu
      res.json({
        "status" : 200,
        "messenger" : "Thêm thành công",
        "data" : result
      })
    }
    else
    {
      // Nếu thêm không thành công result null, thông báo không thành công
      res.json({
        "status" : 400 ,
        "messenger" : "Lỗi, thêm không thành công",
        "data" : null
      })
    }
  } catch (error) {
    console.log(error);
  }
})
```

Tạo Model JSON GHN Order

(<https://api.ghn.vn/home/docs/detail?id=123>)

```
1  {
2    "payment_type_id": 2,
3    "note": "Tintest 123",
4    "required_note": "KHONGCHOXEMHANG",
5    "return_phone": "0332190444", // số điện thoại trả hàng
6    "return_address": "39 NTT", // địa chỉ trả hàng
7    "to_name": "TinTest124", // tên người nhận
8    "to_phone": "0987654321", // Số điện thoại người nhận
9    "to_address": "72 Thành Thái, Phường 14, Quận 10, Hồ Chí Minh, Vietnam", // địa chỉ người nhận
10   "to_ward_code": "90768", // mã phường/xã
11   "to_district_id": 3695, // mã huyện/quận
12   "cod_amount": 200000, // giá
13   "content": "Theo New York Times", // nội dung đặt hàng
14   "weight": 200, // trọng lượng
15   "length": 1, // chiều dài hàng (theo đóng gói)
16   "width": 19, // chiều rộng hàng (theo đóng gói)
17   "height": 10, // chiều cao hàng (theo đóng gói)
18   "insurance_value": 10000000, // khai báo giá trị lô hàng
19   "service_id": 0, // loại dịch vụ mặc định
20   "service_type_id": 2, // loại giao hàng 2
21   "items": [ // sản phẩm
22     {
23       "name": "Áo Polo",
24       "code": "Polo123",
25       "quantity": 1, // Yêu cầu phải có
26       "price": 200000,
27       "weight": 1200 // Yêu cầu phải có
28     }
29   ]
30 }
31
32 }
```

Tạo Model JSON GHN Order

(<https://api.ghn.vn/home/docs/detail?id=123>) (tiếp theo)

```
public class GHNItem {  
    private String name,code;  
    private int quantity,price,weight;  
  
    public GHNItem() {  
  
    }  
  
    public GHNItem(String name, String code, int quantity, int price, int weight) {  
        this.name = name;  
        this.code = code;  
        this.quantity = quantity;  
        this.price = price;  
        this.weight = weight;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public String getCode() {  
        return code;  
    }  
}
```

Tạo Model JSON GHN Order

(<https://api.ghn.vn/home/docs/detail?id=123>) (*tiếp theo*)

```
public class GHNOrderRequest {
    private int payment_type_id;
    private String note, required_note, return_phone, return_address;
    private String to_name, to_phone, to_address, to_ward_code;
    private int to_district_id, cod_amount;
    private String content;
    private int weight, length, width, height, insurance_value, service_id, service_type_id;
    private ArrayList<GHNItem> items;

    public GHNOrderRequest() {
    }

    public GHNOrderRequest(int payment_type_id, String note, String required_note, String return_phone, String return_address, String to_name, String to_phone, String to_address, String to_ward_code, int to_district_id, int cod_amount, String content, int weight, int length, int width, int height, int insurance_value, int service_id, int service_type_id, ArrayList<GHNItem> items) {
        this.payment_type_id = payment_type_id;
        this.note = note;
        this.required_note = required_note;
        this.return_phone = return_phone;
        this.return_address = return_address;
        this.to_name = to_name;
        this.to_phone = to_phone;
        this.to_address = to_address;
        this.to_ward_code = to_ward_code;
        this.to_district_id = to_district_id;
        this.cod_amount = cod_amount;
        this.content = content;
        this.weight = weight;
        this.length = length;
        this.width = width;
        this.height = height;
        this.insurance_value = insurance_value;
        this.service_id = service_id;
        this.service_type_id = service_type_id;
        this.items = items;
    }
}
```

Tạo Model JSON GHN Order

(<https://api.ghn.vn/home/docs/detail?id=123>) (tiếp theo)

```
//Hàm tạo cho đơn hàng mặc định
public GHNOrderRequest(String to_name, String to_phone, String to_address, String to_ward_code, int to_district_id, ArrayList<GHNItem> items) {
    this.to_name = to_name;
    this.to_phone = to_phone;
    this.to_address = to_address;
    this.to_ward_code = to_ward_code;
    this.to_district_id = to_district_id;
    this.items = items;
    payment_type_id = 2;
    note = "Vui lòng gọi khách trước khi giao";
    required_note = "KHONGCHOXEMHANG";
    return_phone = "0332190444";
    return_address = "39 NTT";
    content = "";
    cod_amount = 0;
    this.items.forEach(item -> {
        cod_amount += item.getPrice();
        weight += item.getWeight();
    });
    length = this.items.size();
    width = 19;
    height = 10;
    insurance_value = cod_amount;
    service_id = 0;
    service_type_id = 2;
}
```

Tạo Model JSON GHN Order

(<https://api.ghn.vn/home/docs/detail?id=123>) (tiếp theo)

```
public class GHNOrderResponse {  
    private String order_code;  
  
    public GHNOrderResponse(String order_code) {  
        this.order_code = order_code;  
    }  
  
    public String getOrder_code() {  
        return order_code;  
    }  
  
    public void setOrder_code(String order_code) {  
        this.order_code = order_code;  
    }  
}
```

Tạo Model JSON GHN Order

(<https://api.ghn.vn/home/docs/detail?id=123>) (tiếp theo)

```
public class Order {  
    private String _id;  
    private String order_code;  
    private String id_user;  
  
    public Order(String _id, String order_code, String id_user) {  
        this._id = _id;  
        this.order_code = order_code;  
        this.id_user = id_user;  
    }  
  
    public String get_id() {  
        return _id;  
    }  
  
    public void set_id(String _id) {  
        this._id = _id;  
    }  
  
    public String getOrder_code() {  
        return order_code;  
    }  
  
    public void setOrder_code(String order_code) {  
        this.order_code = order_code;  
    }  
  
    public String getId_user() {  
        return id_user;  
    }  
  
    public void setId_user(String id_user) {  
        this.id_user = id_user;  
    }  
}
```


❑ Thêm Interface GHNServices và interface ApiService

```
public interface GHNServices {  
    public static String GHN_URL = "https://dev-online-gateway.ghn.vn/";  
    // You, Moments ago • Uncommitted changes  
    @POST("ship/public-api/v2/shipping-order/create")  
    Call<ResponseGHN<GHNOrderResponse>> GHNOrder(@Body GHNOrderRequest ghnOrderRequest);  
}
```

```
public interface ApiService {  
    // Sử dụng máy ảo android studio thì localhost thay thành ip 10.0.0.2  
    // Đối với máy thật ta sử dụng ip của máy  
    // Base_URL là url của api  
    public static String BASE_URL = "http://10.0.2.2:3000/api/";  
  
    @POST("add-order")  
    Call<Response<Response<Order>>> order(@Body Order order);  
    // You, Moments ago • Uncommitted changes  
}
```

□ Tại file **LocationActivity**

```
btn_order.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        if(WardCode.equals("")) return;  
        Fruit fruit = (Fruit) getIntent().getExtras().getSerializable(key: "item");  
        GHNItem ghnItem = new GHNItem();  
        ghnItem.setName(fruit.getName());  
        ghnItem.setPrice(Integer.parseInt(fruit.getPrice()));  
        ghnItem.setCode(fruit.get_id());  
        ghnItem.setQuantity(1);  
        ghnItem.setWeight(50);  
        ArrayList<GHNItem> items = new ArrayList<>();  
        items.add(ghnItem);  
        GHNOrderRequest ghnOrderRequest = new GHNOrderRequest(  
            edt_name.getText().toString(),  
            edt_phone.getText().toString(),  
            edt_location.getText().toString(),  
            WardCode,  
            DistrictID,  
            items  
        );  
        ghnRequest.callAPI().GHNOrder(ghnOrderRequest).enqueue(responseOrder);  
    }  
});
```

Tại file **LocationActivity** (tiếp theo)

```
Callback<ResponseGHN<GHNOrderResponse>> responseOrder = new Callback<ResponseGHN<GHNOrderResponse>>() {
    @Override
    public void onResponse(Call<ResponseGHN<GHNOrderResponse>> call, retrofit2.Response<ResponseGHN<GHNOrderResponse>> response) {
        if(response.isSuccessful()){
            if(response.body().getCode() == 200)
            {
                Toast.makeText(context: LocationActivity.this, text: "Đặt hàng thành công", Toast.LENGTH_SHORT).show();
                Order order = new Order();
                order.setOrder_code(response.body().getData().getOrder_code());
                order.setId_user(getSharedPreferences( name: "INFO",MODE_PRIVATE).getString( s: "id", s1: ""));
                httpRequest.callAPI().order(order).enqueue(responseOrderDatabase);
            }
        }
    }

    @Override
    public void onFailure(Call<ResponseGHN<GHNOrderResponse>> call, Throwable t) {

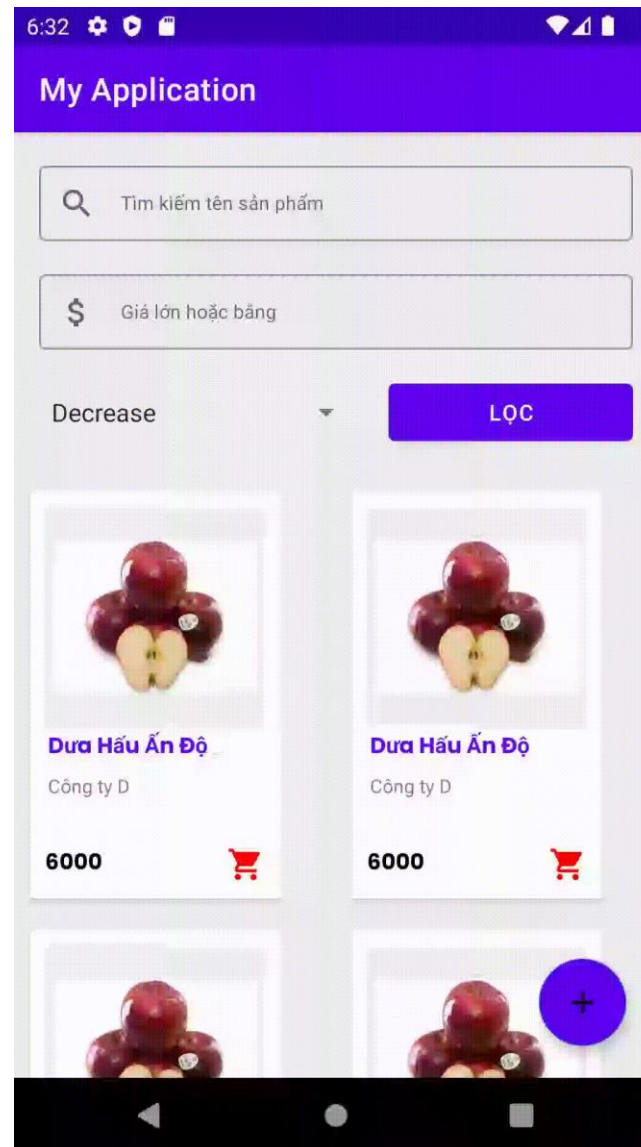
    }
};

Callback<Response<Order>> responseOrderDatabase = new Callback<Response<Order>>() {
    @Override
    public void onResponse(Call<Response<Order>> call, retrofit2.Response<Response<Order>> response) {
        if(response.isSuccessful()){
            if(response.body().getStatus() == 200)
            {
                Toast.makeText(context: LocationActivity.this, text: "Cảm ơn đã mua hàng", Toast.LENGTH_SHORT).show();
                finish();
            }
        }
    }

    @Override
    public void onFailure(Call<Response<Order>> call, Throwable t) {

    }
};
```

 Xem video demo







FPT Education

FPT POLYTECHNIC

Thank you