Lập trình trên thiết bị di động

API

GV: Nguyễn Huy Cường

Email: nh.cuong@hutech.edu.vn

Nội dung

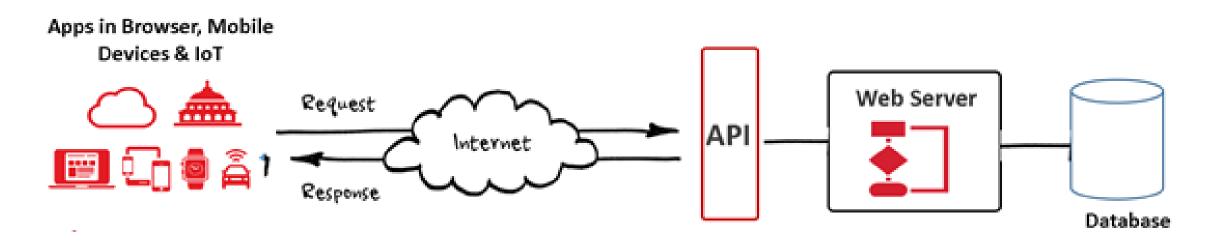
1. API

- API Là gì?
- Tạo Web API
- Kết nối ứng dụng với API

2. Dịch vụ API

- ☐ FireBase API
- Google MAP API
- Google Drive API, Youtube API, Facebook API...

API là gì?



- API (Application Programming Interface): phương thức trung gian kết nối các ứng dụng và thư viện khác nhau.
- API chỉ là các hàm hay thủ tục thông thường. Các hàm này được viết trên nhiều ngôn ngữ lập trình khác nhau: PHP, Python, Ruby, .NET (C#), Node/Javascript, Java...
- API có thể trả kết quả về JSON/ XML

Ứng dụng của API

Web API:

- □ Là hệ thống API được sử dụng trong các hệ thống website, chẳng hạn: Google, Facebook... cung cấp hệ thống API cho phép bạn kết nối, lấy dữ liệu hoặc cập nhật cơ sở dữ liệu.
- □ Đa số Web API được thiết kế theo tiêu chuẩn RESTful.

API trên hệ điều hành:

- □ Windows hay Linux có rất nhiều API là đặc tả các hàm, phương thức cũng như các giao thức kết nối.
- giúp tạo ra các phần mềm ứng dụng có thể tương tác trực tiếp với hệ điều hành.

API của thư viện phần mềm (framework):

- ☐ Mô tả và quy định các hành động mà các thư viện cung cấp.
- □ Một API có thể có nhiều cách triển khai khác nhau, giúp cho một chương trình viết bằng ngôn ngữ này có thể sử dụng được thư viện viết bằng ngôn ngữ khác.

Web API

Khi phát triển ứng dụng di động, sử dụng Web API giúp tận dụng được các tài nguyên trên Internet, tạo ra ứng dụng linh hoạt và kết nối với nhiều dịch vụ khác nhau:

- ☐ Truy xuất dữ liệu từ Server
- ☐ Gửi Dữ liệu lên Server
- ☐ Thông báo và cập nhật thời gian thực
- ☐ Tích hợp Dịch vụ bên ngoài:
- □ ...



1. Sử dụng Mock API

- **Mock API:** Mock AI tạo ra API giả lập cho website
- (2) Sử dụng Công cụ để tạo Mock API: MockAPI / Postman/ Swagger/ WireMock

Ví dụ: Tạo API trả về ds sản phẩm (id, name, image, price)

https://6577c34f197926adf62ec1ca.mockapi.io/api/products

```
"Name": "Luxurious Steel Table",
"Image": "https://loremflickr.com/640/480",
"Price": "789.00",
"ld": "1"
```

```
// https://6577c34f197926adf62ec1ca.mockapi.io/api/products
3
            "Name": "Luxurious Steel Table",
            "Image": <a href="https://loremflickr.com/640/480",">"https://loremflickr.com/640/480",</a>
            "Price": "789.00",
            "Id": "1"
10
11 ▼
            "Name": "Generic Frozen Shirt",
12
            "Image": "https://loremflickr.com/640/480",
13
            "Price": "801.00".
14
            "Id": "2"
15
16
17
```



2. Web API .NET Core C# vs MSSQL server

Tham khảo: https://www.youtube.com/watch?v=b8fFRXoT38M

- (o) Tao project Web API .NET CORE
 - Cài đặt từ Nuget: Microsoft. Entity Framework Core;

Microsoft.EntityFrameworkCore.Tools

Microsoft.EntityFrameworkCore.SqlServer

- (1) Thực hiện mô hình code-first
 - appsettings.json: Bổ sung chuỗi kết nối
 - Tao Models cho **Product.cs** và **DataContext.cs**
 - Program.cs: Thêm DataContext của SQL server

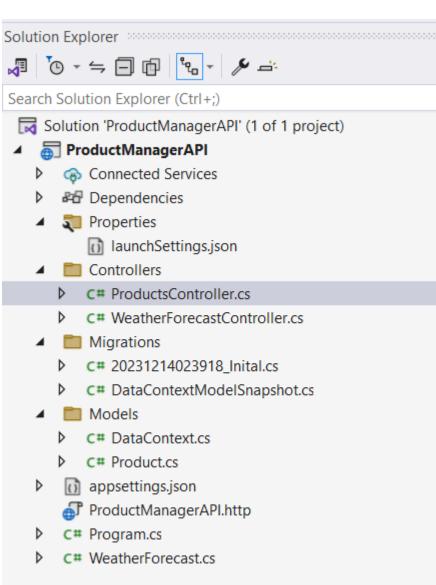
builder.Services.AddDbContext<DataContext>(options => options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection")));

- Package console: Chạy các câu lệnh

Add-Migration Initial

Update-Database

(2) Add API từ ProductController

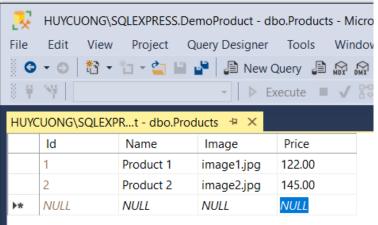


Web API

```
← → C º= localhost:7013/api/Products

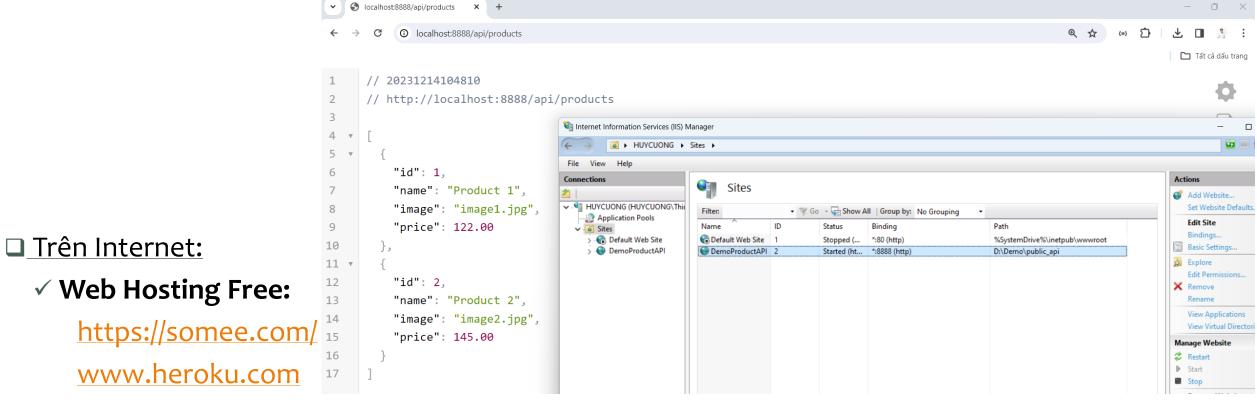
HUYCUONG\SQLEXPRESS DemoProduct - dbo Products - Min
```

```
// 20231214095805
      // https://localhost:7013/api/Products
3
4
5
          "id": 1,
6
          "name": "Product 1",
          "image": "image1.jpg",
8
          "price": 122.00
9
10
        },
11 ▼
          "id": 2,
12
          "name": "Product 2",
13
14
          "image": "image2.jpg",
          "price": 145.00
15
16
17
```



Publish web API

- (3) Publish Web API
- □ Local (IIS): Tìm hiểu Open IIS Manager Windows 10/11



• • • •

✓ Dịch Vụ Web Hosting

Kết nối API trong Android

- Add permission két női internet (AndroidManifest.Xml)
 - <uses-permission android:name="android.permission.INTERNET" />
- Lấy dữ liệu API trong Android:
 - Không sử dụng thư viện: AsyncTask
 - ☐ Sử dụng thư viện: Retrofit, Volley

Retrofit: không hỗ trợ hình ảnh





Phải truy cập được web API



onPostExecute()

Kết nối API trong Android

1 - Sử dụng AsyncTask

- AsyncTask sử dụng để thực hiện bên dưới nền của ứng dụng
- Phải Implement AsyncTask cho doInBackground và onPostExcute
- Sử dụng bộ đệm BufferedReader sao chép các phần vào bộ nhớ chính.

Trong **AsyncTask<Params, Progress, Result>** có 3 đối số là các Generic Type:

✓ Params: Là giá trị (biến) được truyền vào khi gọi thực thi tiến trình và nó sẽ

onPreExecute()

onProgressUpdate()

được truyền vào dolnBackground

✓ **Progress:** Là giá trị (biến) dùng để update giao diện diện lúc tiến trình thực thị, biến này sẽ được truyền vào hàm **onProgressUpdate**.

✓ Result: Lưu trữ kết quả trả về sau khi tiến trình thực hiện xong.

Những đối số không sử dụng trong tiến trình thì thay bằng Void.



Lấy tất cả các products

```
private class GetProductsTask extends AsyncTask<String, Void, List<Product>> {
    no usages
    @Override
    protected List<Product> doInBackground(String... params) {
        String apiUrl = params[0];
       try {
            URL url = new URL(apiUrl);
            HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();
            try {
                BufferedReader reader = new BufferedReader(new InputStreamReader(urlConnection.getInputStream()));
                StringBuilder stringBuilder = new StringBuilder();
                String line;
                while ((line = reader.readLine()) != null) {
                    stringBuilder.append(line).append("\n");
               return parseJson(stringBuilder.toString());
            } finally {
               urlConnection.disconnect();
        } catch (IOException e) {
            e.printStackTrace();
            return null;
    private List<Product> parseJson(String json) {
       // Implement JSON string and create a list of Product objects
        return null;
```

Kết nối API trong Android

2-Sử dụng thư viện: Retrofit2

(o) Cài đặt Retrofit (phiên bản 2) ở build.gradle.kts

```
implementation ("com.squareup.retrofit2:retrofit:2.9.0")
implementation ("com.squareup.retrofit2:converter-gson:2.9.0")
```

gson: hỗ trợ convert sang dữ liệu JSON

(1) Tạo Models/product.cs tương ứng

Hiểu về dữ liệu JSON nhận được

Thuộc tính nếu khác tên chuyển đổi thì cần @SerializedName("@tên_ở_JSON")

1 số trang hỗ trợ chuyển đổi JSON to Object: https://www.jsonschema2pojo.org/

(2) Tạo **Interface** để định nghĩa các API

```
public interface IProductService {
    @GET("api/products")
    Call<List<Product>> getProducts();
}
```

Sử dụng Retrofit2

- (3) Tạo Lớp **RetrofitInstance**: chịu trách nhiệm cấu hình và tạo các API service sử dụng để gửi các yêu cầu HTTP
- * BASE_URL: địa chỉ URL cơ bản mà tất cả các yêu cầu mạng trong một ứng dụng sử dụng Retrofit sẽ được kết hợp với interfaceAPI để tạo ra URL hoàn chỉnh.

Sử dụng Retrofit2

(4) Viết hàm sử dụng Retrofit

"enqueue": thực hiện các yêu cầu mạng bất đồng bộ

□ onRespone: được gọi

Khi yêu cầu thành công

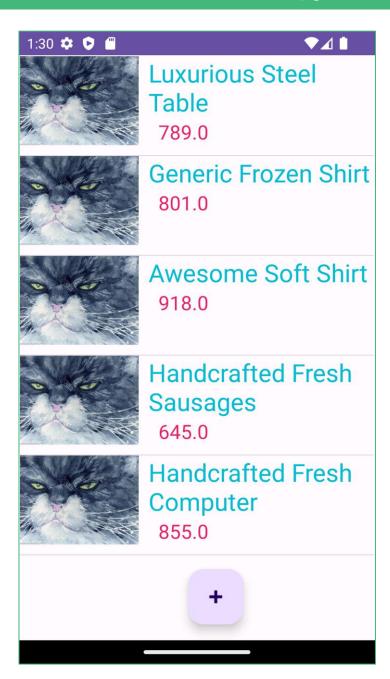
□onFailure: khi có lỗi

xảy ra

```
private void GetListProductAPI() {
  Call<List<Product>> call = RetrofitInstance.getRetrofitInstance().create(IProductService.class).getProducts();
  call.enqueue(new Callback<List<Product>>() {
     @Override
    public void onResponse(Call<List<Product>> call, Response<List<Product>> response) {
       if (response.isSuccessful()) {
         listProduct.addAll(response.body());
         adapter.notifyDataSetChanged();
         Toast.makeText(getBaseContext(), "Total products: " + listProduct.size(), Toast.LENGTH_SHORT).show();
       } else {
         //int statusCode = response.code();
         Toast.makeText(getBaseContext(), "Error: " + response.message(), Toast.LENGTH_SHORT).show();
     @Override
    public void onFailure(Call<List<Product>> call, Throwable t) {
       // Handle the failure, Handle the failure appropriately based on your application's requirements
       Log.e("Network Request", "Error", t);
```

Lấy tất cả các sản phẩm từ API - GET

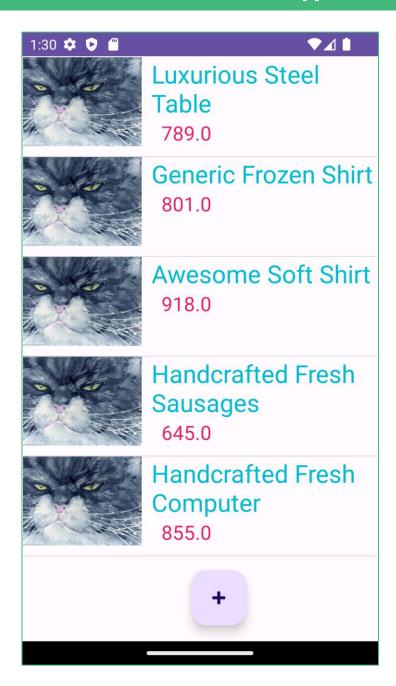
```
List<Product> listProduct = new ArrayList<>();
ProductAdapter adapter;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    GetListProductAPI();
    adapter = new ProductAdapter(listProduct);
    RecyclerView rcvProduct = findViewById(R.id.rcvProduct);
    rcvProduct.setAdapter(adapter);
    rcvProduct.setLayoutManager(new LinearLayoutManager(context: this));
    RecyclerView.ItemDecoration itemDecoration = new DividerItemDecoration(context this, DividerItemDecoration.VERTICAL);
    rcvProduct.addItemDecoration(itemDecoration);
    FloatingActionButton btnAdd = findViewById(R.id.btnAdd);
    btnAdd.setOnClickListener(new View.OnClickListener() {...});
private void GetListProductAPI() {
    Call<List<Product>> call = RetrofitInstance.getRetrofitInstance().create(IProductService.class).getProducts();
    call.enqueue(new Callback<List<Product>>() {
        public void onResponse(Call<List<Product>> call, Response<List<Product>> response
            if (response.isSuccessful()) {
                listProduct.addAll(response.body());
                adapter.notifyDataSetChanged();
                Toast.makeText(getBaseContext(), text: "Total products: " + listProduct.size(), Toast.LENGTH_SHORT).show();
            } else {
                //int statusCode = response.code();
                Toast.makeText(qetBaseContext(), text: "Error: " + response.message(), Toast.LENGTH_SHORT).show();
        @Override
        public void onFailure(Call<List<Product>> call, Throwable t) {
            // Handle the failure, Handle the failure appropriately based on your application's requirements
            Log.e( tag: "Network Request", msg: "Error", t);
    });
```



});

DANO Thêm sản phẩm - POST

```
viewDialog.findViewById(R.id.btnDialogSaveProduct).setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View view) {
     Product p = new Product(txtName.getText().toString(), Float.parseFloat(txtprice.getText().toString()),
txtImage.getText().toString());
    Call<Product> call =RetrofitInstance.getRetrofitInstance().create(IProductService.class).createProduct(p);
    call.enqueue(new Callback<Product>() {
       @Override
       public void onResponse(Call<Product> call, Response<Product> response) {
         if (response.isSuccessful()) {
            // Handle successful response
            Product createdProduct = response.body();
            listProduct.add(createdProduct);
          // adapter.notifyDataSetChanged();
            adapter.notifyItemInserted(listProduct.size()-1);
           else {// Handle error response
       @Override
       public void onFailure(Call<Product> call, Throwable t) {
         // Handle failure
    //thong bao thành công
     Toast.makeText(viewDialog.getContext(), "Thêm dữ liệu thành công", Toast.LENGTH_SHORT).show();
    alert.dismiss(); //thoát khỏi dialog
```



CÁC DỊCH VỤ API

Firebase API

Google Map API

Google Drive API

Youtube API

Facebook API

1- Firebase API

- Tổng quan về firebase
 - □ 1 nền tảng giúp phát triển các ứng dụng di động
 - □ 1 dịch vụ cơ sở dữ liệu hoạt động trên nền tảng cloud với hệ thống máy chủ mạnh mẽ của Google.
 - ☐ Chứa cơ sở dữ liệu mang đến khả năng code nhanh và thuận tiện hơn
 - ☐ Đơn giản hóa các thao tác với cơ sở dữ liệu sẵn có.
- Lịch sử hình thành và phát triển firebase
- □ 2011, Evolve (James Tamplin và Andrew Lee) chuyên cung cấp các API để tích hợp tính năng trò chuyện vào các trang web. phát triển Envole và tạo nên Firebase
- □ 2014, Firebase được Google mua lại và trở thành một ứng dụng đa năng trên nền tảng di động và website.



Các tính năng chính của Firebase

Các tính năng chính:











Các tính năng chính của Firebase Realtime Database

Realtime Database

- Realtime Database là một cơ sở dữ liệu thời gian thực được lưu trữ dưới dạng **JSON** và được đồng bộ hóa theo thời gian thực đối với mọi kết nối.
- Đối với các ứng dụng được xây dựng trên đa nền tảng như Android, iOS và WebApp, tất cả client sẽ cùng sử dụng một cơ sở dữ liệu. Bên cạnh đó, hệ thống dữ liệu này sẽ tự động cập nhật khi lập trình viên phát triển ứng dụng. Sau đó, tất cả dữ liệu này sẽ được truyền tải thông qua các kết nối SSI có 2048 bit.



Các tính năng chính của Firebase Authentication

Authentication

Authentication là tính năng giúp xác thực danh tính của người dùng ứng dụng. Firebase cung cấp các bước xác thực thông qua Email, Facebook, Twitter, GitHub hay Google. Điều này giúp cho các thông tin cá nhân của khách hàng được bảo vệ một cách tốt nhất, hạn chế được tình trạng bị hacker đánh cắp. Đồng thời việc xác thực danh tính qua Firebase sẽ giúp người dùng tiếp cận sản phẩm nhanh chóng và an toàn hơn.



Các tính năng chính của Firebase

Cloud Storage

Cloud Storage

Cloud Storage là tính năng cho phép lưu trữ và quản lý nội dung đã tạo ra như ảnh, video, nội dung, văn bản,... Firebase Storage cung cấp các API hỗ trợ bạn upload và download các file từ ứng dụng một cách trơn tru mà không cần quan tâm đến chất lượng đường truyền mạng với độ bảo mật cao.



Các tính năng chính của Firebase Hosting

Hosting

□ Hosting được phân phối thông qua tiêu chuẩn công nghệ bảo mật SSI từ hệ thống mạng CDN. CDN là một mạng lưới máy chủ giúp lưu trữ các bản sao của các nội dung tĩnh trên website. Thông qua CDN, người dùng có thể truy cập và sử dụng các dịch vụ trên web khi cài Firebase Hosting một cách nhanh chóng và ổn định hơn.



Các tính năng chính của Firebase Analytics

Analytics

Analytics giúp bạn có thể phân tích hành vi của người sử dụng ứng dụng của bạn. Qua đó, bạn sẽ biết được khách hàng thường xuyên truy cập tính năng nào và các thông tin về hiệu quả quảng cáo, tình trạng trả phí,... để có thể đưa ra được chiến lược phát triển phù hợp. Để thực hiện tính năng Analytics của Firebase, bạn cần cài đặt Software Development Kit (SDK).



Ưu nhược điểm của firebase

- Uu điểm
 - ☐Sử dụng miễn phí và thuận tiện
 - □Dễ sử dụng và tích hợp
 - ■Đáp ứng nhu cầu của người dùng
 - □Cập nhật liên tục và đa nền tảng

- Nhược điểm
 - ☐Giới hạn về quy mô ứng dụng
 - ☐ Khả năng tùy chỉnh hạn chế
 - ☐Giá các dịch vụ của Firebase



- (1) TẠO DỰ ÁN TRÊN FIREBASE
- (2) XỬ LÝ MỘT SỐ THAO TÁC TRÊN FIREBASEFIRESTORE
- (3) AUTHENTICATION TRÊN FIREBASE XÁC THỰC BẰNG EMAIL/PASSWORD XÁC THỰC BẰNG SỐ ĐIỆN THOẠI

1- Tạo 1 dự ÁN TRÊN FIREBASE: 4 STEPS

× Add Firebase to your Android app Register app Android package name ② com.company.appname App nickname (optional) ② My Android App Debug signing certificate SHA-1 (optional) ① Required for Dynamic Links, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings. Register app Download and then add config file Add Firebase SDK Next steps

XỬ LÝ MỘT SỐ THAO TÁC TRÊN FIREBASEFIRESTORE

- Lắng nghe dữ liệu trên FirebaseFirestore khi thêm/ thay đổi/xóa 1 dòng dữ liệu mới
- Tham khảo

https://firebase.google.com/docs/fires tore/query-data/listen#java

```
public void listenProductFirestore(ProductAdapter adapter) {
    db.collection( collectionPath: "Product").addSnapshotListener(new EventListener<QuerySnapshot>() {
        public void onEvent(@Nullable QuerySnapshot snapshots, @Nullable FirebaseFirestoreException e) {
            if (e != null) {
                Log.w( tag: "FireBase", msg: "listen:error", e);
                return;
            for (DocumentChange dc : snapshots.getDocumentChanges()) {
                Product product = dc.getDocument().toObject(Product.class);
                switch (dc.getType()) {
                    case ADDED:
                        adapter.getListProduct().add(product);
                        adapter.notifyItemInserted( position: adapter.getListProduct().size() - 1);
                        break;
                    case MODIFIED:
                        if (dc.getOldIndex() == dc.getNewIndex()) //néu vi trí đối tương tương đồng với vi trí mới
                            adapter.getListProduct().set(dc.getOldIndex(), product);
                            adapter.notifyItemChanged(dc.getOldIndex());
                        } else //khác vị trí sẽ xóa đối tượng ở danh sách và thêm lại
                            adapter.getListProduct().remove(dc.getOldIndex());
                            adapter.getListProduct().add(dc.getNewIndex(), product);
                            adapter.notifyItemMoved(dc.getOldIndex(), dc.getNewIndex());
                        break:
                    case REMOVED:
                        adapter.getListProduct().remove(product);
                        adapter.notifyItemRemoved(dc.getOldIndex());
                        break;
    });
```

XỬ LÝ MỘT SỐ THAO TÁC TRÊN FIREBASEFIRESTORE

Thêm, thay đổi, xóa một dữ liệu

https://firebase.google.com/docs/firestore/manage-data/add-data https://firebase.google.com/docs/firestore/manage-data/delete-data

XÁC THỰC BẰNG EMAIL/PASSWORD

Đăng ký email/password

Đăng nhập

```
mAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener( activity: this, new OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if (task.isSuccessful()) {
            //Iấy thông tin tài khoản mới vừa đặng ký
            FirebaseUser user = mAuth.getCurrentUser();
            Toast.makeText( context: SignUpActivity.this, text: "Đặng ký thành công", Toast.LENGTH_SHORT).show();
        finish();
        } else {
            Log.w(TAG, msg: "createUserWithEmail:failure", task.getException());
            Toast.makeText( context: SignUpActivity.this, text: "Đặng ký thật bại", Toast.LENGTH_SHORT).show();
        }
    }
}
});
```

XÁC THỰC BẰNG EMAIL/PASSWORD

Quên mật khẩu

```
auth.sendPasswordResetEmail(emailAddress).addOnCompleteListener(new OnCompleteListener<Void>() {
    @Override
    public void onComplete(@NonNull Task<Void> task) {
        if (task.isSuccessful()) {
            Toast.makeText( context: LoginActivity.this, text: "Vui lòng kiểm tra hộp thư để cập nhật lại mật khẩu", Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText( context: LoginActivity.this, text: "Lỗi gửi mail", Toast.LENGTH_SHORT).show();
        }
    }
}
});
```

Đăng xuất

```
FirebaseAuth.getInstance().signOut();
```

XÁC THỰC BẰNG SỐ ĐIỆN THOẠI

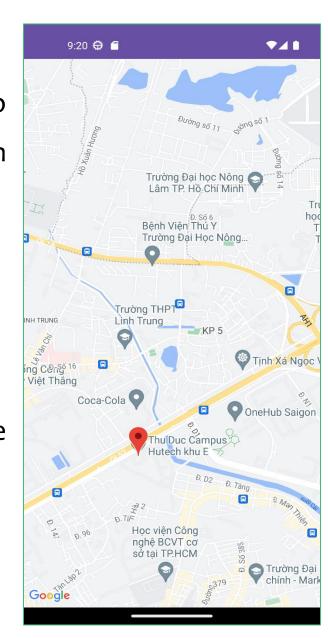
Yêu cầu gửi OTP về số điện thoại

XÁC THỰC BẰNG SỐ ĐIỆN THOẠI

Xác thực số điện thoại

```
private void verifyOTP(String code)
    PhoneAuthCredential credential = PhoneAuthProvider.getCredential(mVerificationId, code);
    signInWithPhoneAuthCredential(credential);
private void signInWithPhoneAuthCredential(PhoneAuthCredential credential) {
    mAuth.signInWithCredential(credential)
            .addOnCompleteListener( activity: this, new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful()) {
                        // Sign in success, update UI with the signed-in user's information
                        Toast.makeText( context: LoginWithPhoneActivity.this, text: "Đăng nhập thành công", Toast.LENGTH_SHORT).show();
                        FirebaseUser user = task.getResult().getUser();
                        startActivity(new Intent( packageContext: LoginWithPhoneActivity.this,LogoutActivity.class));
                        // Update UI
                    } else {
                        // Sign in failed, display a message and update the UI
                        Log.w(TAG, msg: "signInWithCredential:failure", task.getException());
                        if (task.getException() instanceof FirebaseAuthInvalidCredentialsException) {
                            // The verification code entered was invalid
            });
```

- Tổng quan về Google Map API
- Google đã xây dựng 1 bộ dữ liệu khổng lồ về bản đồ điện tử và cung cấp API để các lập trình viên có thể sử dụng và tùy biến cho nhiều mục đích khác nhau.
- Các bước thực hiện:
 - ✓ Máy ảo được tích hợp Google play services
 - ✓ Cài đặt Google API vào Android Studio
 - ✓ Lấy key cho project để có thể truy cập Google Maps API từ Google Console
 - √ Tạo Project trên Android Studio sử dụng Google Map



Các bước thực hiện

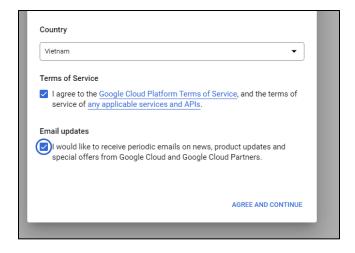
Bước 1:

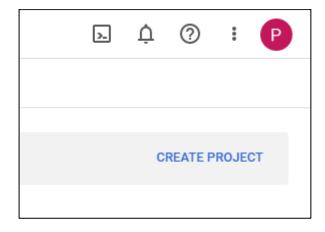
Truy cập vào trang web & đăng nhập bằng tài khoản google của mình

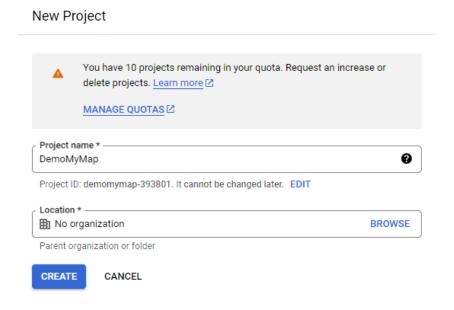
https://console.cloud.google.com/projectselector2/google/maps-apis

check và nhấn AGREE AND CONTINUE

- Góc phải màn hình chọn vào CREATE PROJECT
- Đặt tên cho project và nhấn CREATE

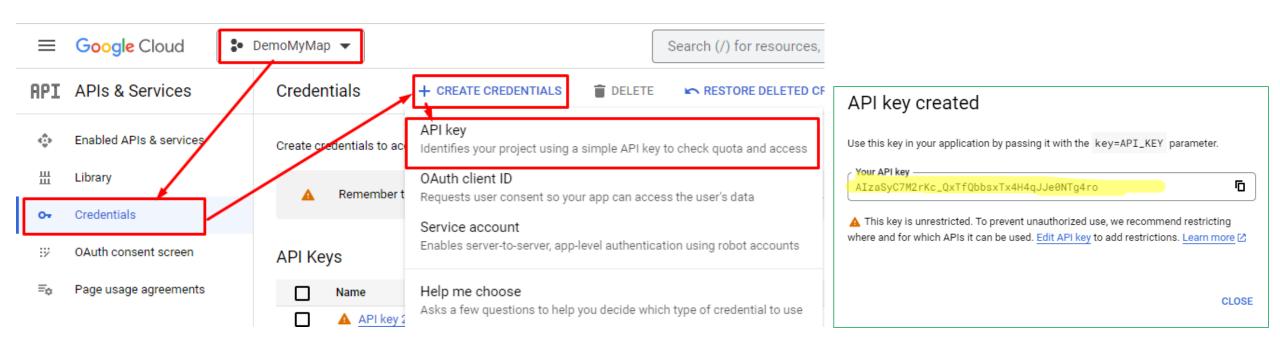






Các bước thực hiện

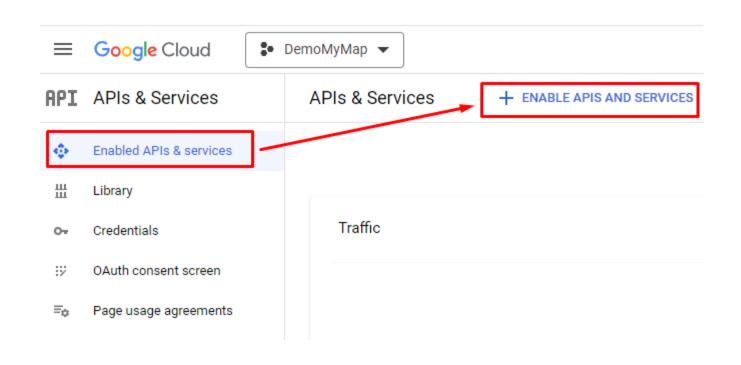
• Truy cập vào https://console.cloud.google.com, chọn project vừa tạo ở trên. Trong mục APIs & Services chọn Credentials, CREATE CREDENTIALS & API key Lưu lại API key để bước sau sử dụng.

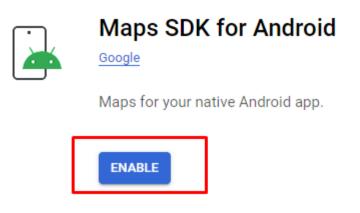


Các bước thực hiện

Chuyển sang Enabled APIs & Services, Chọn Enable APIs & Services

Chọn Maps SDK for Android và Enable





Các bước thực hiện

- Bước 2: Trong project android ta thực hiện setup các bước sau:
- https://developers.google.com/maps/documentation/android-sdk/config
- settings.gradle có pluginManagement ở trước
- build.gradle (Project) thêm plugins id 'com.google.android.libraries.mapsplatform.secrets-gradle-plugin' version '2.0.1' apply false
- □ build.gradle (Module)
 - > plugins
 - id 'com.google.android.libraries.mapsplatform.secrets-gradle-plugin'
 - dependencies

implementation 'com.google.android.gms:play-services-maps:18.1.0'

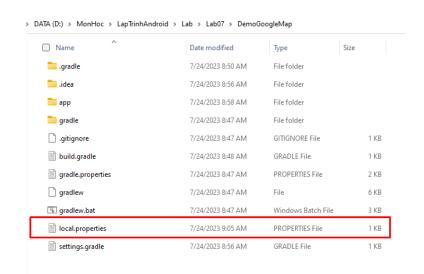
Save & sync your project with Grade

```
pluginManagement {
    repositories {
        gradlePluginPortal()
        google()
        mavenCentral()
    }
}
```

Các bước thực hiện

local.properties: thêm MAPS_API_KEY lấy ở bước 1

MAPS_API_KEY=AlzaSyC7M2rKc_QxTfQbbsxTx4H4qJJeoNTg4ro



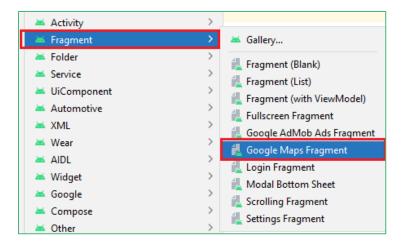
AndroidManifest: thêm meta-data vào application

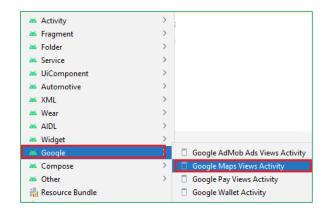
```
<meta-data
android:name="com.google.android.geo.API_KEY"
android:value="${MAPS_API_KEY}" />
```



Các bước thực hiện

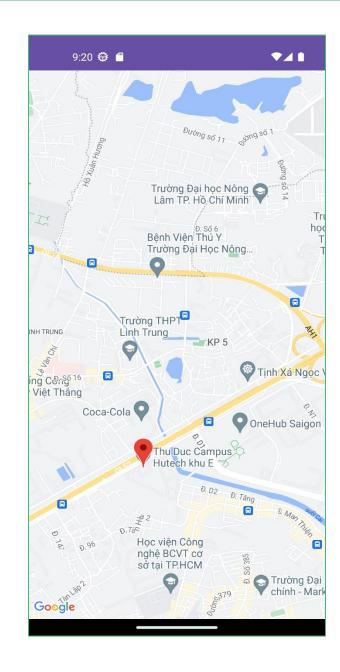
Bước 3: Sử dụng từ Activity/ Fragment





☐ Đánh dấu maker Thu Duc campus

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    // Add a marker in Sydney and move the camera
    LatLng hutechE = new LatLng(10.8555748, 106.7851027);
    mMap.addMarker(new MarkerOptions().position(hutechE).title("Thu Duc campus"));
    mMap.moveCamera(CameraUpdateFactory.newLatLng(hutechE));
}
```



ASM 7.1 Quiz App

 Xây dựng API trả về danh sách các câu hỏi và các đáp án lựa chọn (2đ)

```
"id": 1,
          "title": "Khi môt lập trình viên nói \"Hello World!\" nghĩa là gì?",
          "answer1": "chào đón thế giới mới",
          "answer2": "in ra một chuỗi đơn giản trong ngôn ngữ lập trình",
          "answer3": " gặp một lỗi và cần giúp đỡ",
          "answer4": "Code theo mau",
          "answers": null
14
15
          "id": 2,
          "title": " Làm thế nào lập trình viên giao tiếp với người khác?",
16
          "answer1": "Bằng cách nói \"Hello World!\"",
          "answer2": " Bằng cách nói ngôn ngữ lập trình của mình",
          "answer3": "Bằng cách gửi email",
          "answer4": "Bằng cách tự tin nhấn \"Enter\"",
          "answers": null
23
          "id": 3,
          "title": "Kí hiệu nào thường được sử dụng để đánh giá điểm của sinh viên trong hệ thống điểm số trung bình?",
          "answer1": "GPA",
          "answer2": "SAT",
          "answer3": "NAT"
          "answer4": "NET".
          "answers": null
```

2. Sử dụng RecycleView để hiện thị danh sách câu hỏi và đáp án được lấy từ API (5đ)



11:43 🌣 🖺 📤 💮 🗸 🖺
QuizApp
CODE THEO MÃU
Làm thế nào lập trình viên giao tiếp với người khác?
Bằng Cách nói "HELLO WORLD!"
BẰNG CÁCH NÓI NGÔN NGỮ LẬP TRÌNH CỦA MÌNH
BằNG CÁCH GỬI EMAIL
BẰNG CÁCH TỰ TIN NHẤN "ENTER"
Kí hiệu nào thường được sử dụng để đánh giá điểm của sinh viên trong hệ thống điểm số trung bình?
GPA
SAT
NAT
NET
SUBMIT
Số câu trả lời đúng: 0

ASM 7.1 Quiz App

3. RecycleView được thiết kế scrollable. Khi Lựa chọn 1 phương án trả lời sẽ được tô màu (1đ)



ASM 7.1 Quiz App

- 4. Xử lý Submit button (2đ)
 - 4.1 Xây dựng API để lấy kết quả trả về danh sách các câu hỏi và đáp án đúng
 - 4.2 Thông báo số câu trả lời đúng
 - 4.3 Set màu xanh đáp án đúng

```
" [
    "questionId": 1,
    "correctAnswer": "in ra một chuỗi đơn giản trong ngôn ngữ lập trình"
},
    {
        "questionId": 2,
        "correctAnswer": "Bằng cách gửi email"
},
    {
        "questionId": 3,
        "correctAnswer": "GPA"
}
```



ASM 7.2: EHUTECH

Chương trình E-Hutech cho Trang Chủ (lấy dữ liệu từ API)

```
"id": 1.
      "title": "Sáng tạo bảo tàng bằng tư duy hiện đại, cựu sinh viên HUTECH xuất sắc giành Giải thưởng Kiến trúc Quốc tế AMP 2023",
      "urlNews": "https://www.hutech.edu.vn/homepage/tin-tuc/tin-hutech/14614366-sang-tao-bao-tang-bang-tu-duy-hien-dai-cuu-sinh-vien-hutech.edu.vn/homepage/tin-tuc/tin-hutech/14614366-sang-tao-bao-tang-bang-tu-duy-hien-dai-cuu-sinh-vien-hutech.edu.vn/homepage/tin-tuc/tin-hutech/14614366-sang-tao-bao-tang-bang-tu-duy-hien-dai-cuu-sinh-vien-hutech.edu.vn/homepage/tin-tuc/tin-hutech/14614366-sang-tao-bao-tang-bang-tu-duy-hien-dai-cuu-sinh-vien-hutech.edu.vn/homepage/tin-tuc/tin-hutech/14614366-sang-tao-bao-tang-bang-tu-duy-hien-dai-cuu-sinh-vien-hutech-xuat-sac-gianh-giai-thuong-kien-truc-
quo",
      "urlImage": "https://file1.hutech.edu.vn/file/editor/homepage1/876407-a7407515jpg.jpg",
      "createDate": "2023-12-16"
      "id": 2.
      "title": "Giảng viên Viện Công nghệ Việt - Nhật HUTECH tập huấn xây dựng và vận hành LMS",
      "urlNews": "https://www.hutech.edu.vn/homepage/tin-tuc/tin-hutech/14614332-giang-vien-vien-cong-nghe-viet-nhat-hutech-tap-huan-xay-dung-va-van-hanh-lms",
      "urlImage": "https://file1.hutech.edu.vn/file/editor/homepage1/835820-dsc01341jpg.jpg",
      "createDate": "2023-12-15"
      "id": 4,
      "title": "HUTECH làm việc cùng Công ty Asahi Engineering và Công ty Hitachi Sekkei về hợp tác dự án nghiên cứu HUTECH - Asahi giai đoạn 2\r\n",
      "urlNews": "https://www.hutech.edu.vn/homepage/tin-tuc/tin-hutech/14614328-hutech-lam-viec-cung-cong-ty-asahi-engineering-co-ltd-va-cong-ty-hitachi-sekkei-co-ltd-ve-hop-tac-
      "urlImage": "https://file1.hutech.edu.vn/file/editor/homepage1/72701-a7401253.jpg",
      "createDate": "2023-12-15"
```



ASM 7.2: EHUTECH

Chương trình E-Hutech cho TKB (lấy dữ liệu từ API)

- Lấy dữ liệu cho tuần này
- Lọc dữ liệu cho hôm nay

```
"id": 1,
"createDate": "2023-12-18T00:00:00",
  "id": 1,
  "from": 2.
  "mon": "Lâp trình Windows"
  "lop": "20DTHA1",
  "phong": "E01-07.02"
"ca2":
  "id": 2,
  "from": 7,
  "to": 11,
  "mon": "Lập trình Web"
  "lop": "20DTHA2",
  "phong": "E01-07.04"
"id": 4,
"createDate": "2023-12-19T00:00:00",
"ca1": {
  "id": 3,
 "from": 2,
  "mon": "Lập trình trên thiết bị di động",
  "lop": "20DTHA3",
  "phong": "E01-07.04"
"ca2": null
"id": 7.
"createDate": "2023-12-20T00:00:00",
"ca1": null,
"ca2": {
  "id": 4.
  "from": 7,
  "mon": "Ngôn ngữ phát triển ứng dụng mới",
  "lop": "20DTHA4",
  "phong": "E01-07.08"
```

