

# API

---

APPLICATION PROGRAMMING INTERFACE

# What is API?

---

API is the acronym for **A**pplication **P**rogramming **I**nterface, which is a software intermediary that allows two applications to talk to each other.

Each time you use an app like Facebook, send an instant message, or check the weather on your phone, you're using an API.

# Types of REST API Client

---

- ❖ Retrofit
  - ❖ Volley
  - ❖ okhttp
  - ❖ Android Asynchronous HTTP client
- etc.

# Retrofit

---

Retrofit is a REST Client for Java and Android. It makes it relatively easy to retrieve and upload JSON (or other structured data) via a REST based webservice.

In Retrofit you configure which converter is used for the data serialization. Typically for **JSON** you use **Gson**, but you can add custom converters to process **XML** or other protocols.

Retrofit uses the **OkHttp** library for HTTP requests.

For testing we use this link

---

<http://dummy.restapiexample.com/>

# Using Retrofit

---

Retrofit mainly need three things in android :

1. Model class (the POJO we did in previous classes) which is used as a JSON model
2. Interfaces that define the possible HTTP operations (every method here represents a single possible API call)
3. Retrofit.Builder class - Instance which uses the interface and the Builder API to allow defining the URL end point for the HTTP operations

# Setting up

---

Define gradle dependencies :

**implementation 'com.squareup.retrofit2:retrofit:2.5.0'**

**implementation 'com.squareup.retrofit2:converter-gson:2.5.0'**

Provide Network permission in Android Manifest file :

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

# Using Retrofit

---

## 1. Model Class

```
public class Employee {  
  
    private int id;  
    private String employee_name;  
    private String employee_salary;  
    private String employee_age;  
    private String profile_image;  
  
    public Employee(int id, String employee_name, String employee_salary,  
                    String employee_age, String profile_image) {  
        this.id = id;  
        this.employee_name = employee_name;  
        this.employee_salary = employee_salary;  
        this.employee_age = employee_age;  
        this.profile_image = profile_image;  
    }  
  
    public int getId() {  
        return id;  
    }  
  
    public String getEmployee_name() {  
        return employee_name;  
    }  
  
    public String getEmployee_salary() {  
        return employee_salary;  
    }  
  
    public String getEmployee_age() {  
        return employee_age;  
    }  
}
```



# Using Retrofit

---

## 2. An Interface for possible API Call

```
public interface EmpAPI {  
  
    @GET ("employees")  
    Call<List<Employee>> getEmployee();  
  
}
```

# Using Retrofit

---

## 3. Retrofit Instance

```
Retrofit retrofit = new Retrofit.Builder()  
    .baseUrl(BASE_URL)  
    .addConverterFactory(GsonConverterFactory.create())  
    .build();
```

# Using Retrofit

---

## Making a request

```
EmpAPI empAPI = retrofit.create(EmpAPI.class);  
Call<List<Employee>> listCall = empAPI.getEmployee();  
  
listCall.enqueue(new Callback<List<Employee>>() {  
    @Override  
    public void onResponse(Call<List<Employee>> call,  
                           Response<List<Employee>> response) {  
    }  
  
    @Override  
    public void onFailure(Call<List<Employee>> call, Throwable t) {  
    }  
});
```

# Using Retrofit

```
@Override
public void onResponse(Call<List<Employee>> call, Response<List<Employee>> response) {
    if (!response.isSuccessful()) {
        tvData.setText("Code : " + response.code());
    }
    List<Employee> employeeList = response.body();
    for (Employee employee:employeeList) {
        String data = "";
        data += "ID: " + employee.getId() + "\n";
        data += "Name: " + employee.getEmployee_name() + "\n";
        data += "Salary: " + employee.getEmployee_salary() + "\n";
        data += "Age: " + employee.getEmployee_age() + "\n";
        data += "-----" + "\n";

        tvData.append(data);
    }
}
```

Showing data in TextView

# Using Retrofit

---

## Get error message

```
@Override  
public void onFailure(Call<List<Employee>> call, Throwable t) {  
    tvData.setText("Error: " + t.getMessage());  
}
```

# Retrofit example

---

```
public interface EmpAPI {

    @GET("employees")
    Call<List<Employee>> getEmployee();

    @GET("employee/{empId}")
    Call<Employee> getEmployeeById(@Path("empId") int empId);

    @PUT("update/{empId}")
    Call<Void> updateEmployee(@Path("empId") int empId,
                             @Body EmployeeCud employeeCud);

    @DELETE("delete/{empId}")
    Call<Void> deleteEmployee(@Path("empId") int empId);

    @POST("create")
    Call<Void> addEmployee(@Body EmployeeCud employeeCud);
}
```

# Retrofit example

---

## Model Class for GET

```
public class Employee {  
  
    private int id;  
    private String employee_name;  
    private String employee_salary;  
    private String employee_age;  
    private String profile_image;  
}
```

## Model Class for POST & PUT

```
public class EmployeeCud {  
  
    private String name;  
    private String age;  
    private String salary;  
}
```

# Retrofit example

---

```
public class UpdateActivity extends AppCompatActivity  
    implements View.OnClickListener {  
  
    Retrofit retrofit;  
    EmpAPI empAPI;  
    private static final String BASE_URL =  
        "http://dummy.restapiexample.com/api/v1/";
```



# Retrofit example

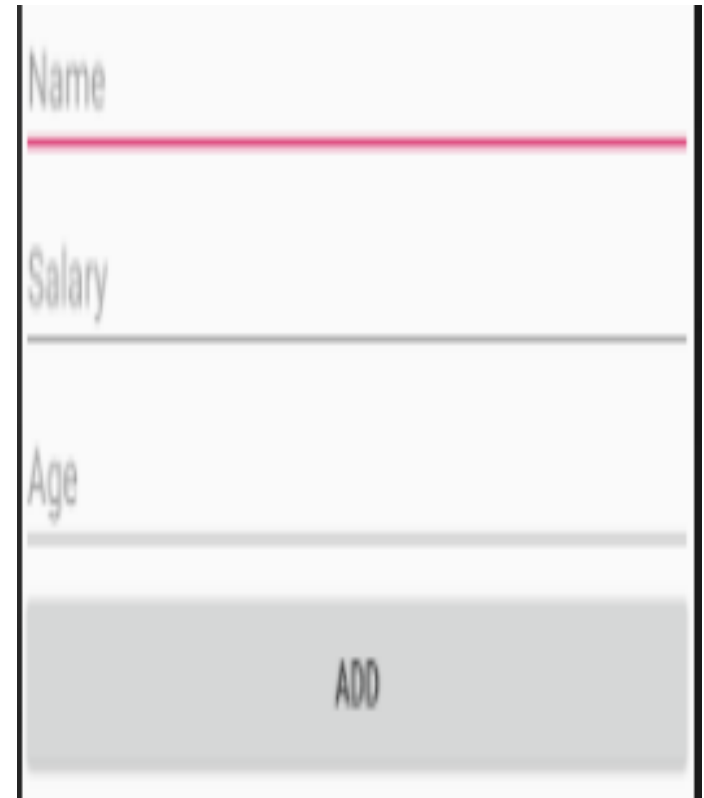
---

```
private void createInstance() {  
    retrofit = new Retrofit.Builder()  
        .baseUrl(BASE_URL)  
        .addConverterFactory(GsonConverterFactory.create())  
        .build();  
    empAPI = retrofit.create(EmpAPI.class);  
}
```

# Retrofit example

---

```
EditText editTextNam, editTextSal, editTextAge;  
Button buttonAdd;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_update);  
  
    editTextNam = findViewById(R.id.etName);  
    editTextSal = findViewById(R.id.etSalary);  
    editTextAge = findViewById(R.id.etAge);  
    buttonAdd = findViewById(R.id.btnAdd);  
}
```



The screenshot shows a mobile application interface with three text input fields stacked vertically. The first field is labeled 'Name' and has a red underline. The second field is labeled 'Salary' and has a grey underline. The third field is labeled 'Age' and has a grey underline. Below these fields is a grey button with the text 'ADD' in white capital letters.

# Retrofit example

```
public void addData() {  
    createInstance();  
    EmployeeCud employeeCud = new EmployeeCud(editTextNam.getText().toString(),  
        editTextAge.getText().toString(), editTextSal.getText().toString());  
    Call<Void> voidCall = empAPI.addEmployee(employeeCud);  
    voidCall.enqueue(new Callback<Void>() {  
        @Override  
        public void onResponse(Call<Void> call, Response<Void> response) {  
            Toast.makeText(context: UpdateActivity.this, text: "added",  
                Toast.LENGTH_SHORT).show();  
        }  
  
        @Override  
        public void onFailure(Call<Void> call, Throwable t) {  
            Toast.makeText(context: UpdateActivity.this, t.getMessage(),  
                Toast.LENGTH_SHORT).show();  
        }  
    });  
}
```

# Retrofit example

```
EditText editTextSrch,editTextNam,editTextSal,editTextAge;  
Button buttonSrch,buttonUpdate,buttonDelete;
```

```
editTextSrch = findViewById(R.id.etSrch) ;  
editTextNam = findViewById(R.id.etName) ;  
editTextSal = findViewById(R.id.etSalary) ;  
editTextAge = findViewById(R.id.etAge) ;  
buttonSrch = findViewById(R.id.btnSrch) ;  
buttonUpdate = findViewById(R.id.btnUpdate) ;  
buttonDelete = findViewById(R.id.btnDelete) ;
```

11

FIND

Jena Gaines

90560

30

UPDATE

DELETE

# Retrofit example

```
public void loadData() {
    createInstance();
    Call<Employee> employeeCall =
        empAPI.getEmployeeById(Integer.parseInt
            ((EditTextSrch.getText().toString())));
    employeeCall.enqueue(new Callback<Employee>() {
        @Override
        public void onResponse(Call<Employee> call, Response<Employee> response) {
            editTextNam.setText(response.body().getEmployee_name());
            editTextAge.setText(response.body().getEmployee_age());
            editTextSal.setText(response.body().getEmployee_salary());
        }

        @Override
        public void onFailure(Call<Employee> call, Throwable t) {
            Log.d(tag: "Error", t.getMessage());
        }
    });
}
```

# Retrofit example

```
public void updateData() {
    createInstance();
    EmployeeCud employeeCud = new EmployeeCud(editTextNam.getText().toString(),
        editTextAge.getText().toString(), editTextSal.getText().toString());
    Call<Void> employeeCall = empAPI.updateEmployee(Integer.parseInt
        (editTextSrch.getText().toString()), employeeCud);
    employeeCall.enqueue(new Callback<Void>() {
        @Override
        public void onResponse(Call<Void> call, Response<Void> response) {
            Toast.makeText(context: UpdateActivity.this, text: "updated",
                Toast.LENGTH_SHORT).show();
        }

        @Override
        public void onFailure(Call<Void> call, Throwable t) {
            Toast.makeText(context: UpdateActivity.this, t.getMessage(),
                Toast.LENGTH_SHORT).show();
        }
    });
}
```

# Retrofit example

```
public void deleteData() {  
    createInstance();  
    Call<Void> voidCall = empAPI.deleteEmployee(Integer.parseInt  
    (editTextSrch.getText().toString()));  
    voidCall.enqueue(new Callback<Void>() {  
        @Override  
        public void onResponse(Call<Void> call, Response<Void> response) {  
            Toast.makeText(context: UpdateActivity.this, text: "deleted",  
                Toast.LENGTH_SHORT).show();  
        }  
  
        @Override  
        public void onFailure(Call<Void> call, Throwable t) {  
            Toast.makeText(context: UpdateActivity.this, t.getMessage(),  
                Toast.LENGTH_SHORT).show();  
        }  
    });  
}
```

# Retrofit example

---

```
@Override
public void onClick(View v) {
    if (v.getId() == R.id.btnAdd) {
        addData();
    }
    if (v.getId() == R.id.btnSrch) {
        loadData();
    }
    if (v.getId() == R.id.btnUpdate) {
        updateData();
    }
    if (v.getId() == R.id.btnDelete) {
        deleteData();
    }
}
```



# Loading image from URL - 1

```
private void StrictMode() {  
    android.os.StrictMode.ThreadPolicy policy =  
        new android.os.StrictMode.ThreadPolicy.Builder().permitAll().build();  
    android.os.StrictMode.setThreadPolicy(policy);  
}  
  
private void loadImage(String imgPath) {  
    StrictMode();  
    try {  
        URL url = new URL(imgPath);  
        Bitmap bmp = BitmapFactory.decodeStream((InputStream) url.getContent());  
        imageViewData.setImageBitmap(bmp);  
    }  
    catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

# Loading image from URL - 2

```
public class ShowImage extends AsyncTask<String,Void,Bitmap>{  
    ImageView imageView;  
    public ShowImage(ImageView imageView) { this.imageView = imageView; }  
  
    @Override  
    protected Bitmap doInBackground(String... url) {  
        String urlDisplay = url[0];  
        bitmap = null;  
        try {  
            InputStream is = new URL(urlDisplay).openStream();  
            bitmap = BitmapFactory.decodeStream(is);  
        }  
        catch (Exception e){  
            e.printStackTrace();  
        }  
  
        return bitmap;  
    }  
  
    @Override  
    protected void onPostExecute(Bitmap bitmap) {  
        super.onPostExecute(bitmap);  
        imageView.setImageBitmap(bitmap);  
    }  
}
```

# Loading image from URL - 3

---

implementation `"com.squareup.picasso:picasso:2.5.2"`

```
imageView = findViewById(R.id.imageView);  
String url = "imageurlhere";  
Picasso.with(this).load(url).into(imageView);
```

# Uploading Image- 1

---

Add the following dependency in build.gradle

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

Declare the following permission on Manifest file

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

# Uploading Image- 2

---

Create an Interface as ImageAPI with the following code

```
public interface ImageAPI {  
    @Multipart  
    @POST("/3/upload")  
    Call<ResponseBody> upload(  
        @Header("Authorization") String clientId,  
        @Part() MultipartBody.Part file  
    );  
}
```

# Uploading Image- 3

---

In the Activity file from where you wish to upload image initialize retrofit with a base url

```
Retrofit retrofit = new Retrofit.Builder()
    .baseUrl("https://api.imgur.com/ ")
    .addConverterFactory(GsonConverterFactory.create())
    .build();
```

Check the run time permission to read external storage as

```
int permissionCheck = ContextCompat.checkSelfPermission( context: this, Manifest.permission.READ_EXTERNAL_STORAGE);

if (permissionCheck != PackageManager.PERMISSION_GRANTED) {
    ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.permission.READ_EXTERNAL_STORAGE}, requestCode: 1);
} else {
    permissionGranted = true;
}
```

# Uploading Image- 4

---

Override onActivityResult and add the following code

```
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == RESULT_OK) {
        if (data == null) {
            Toast.makeText(getApplicationContext(), text: "Please select an Image", Toast.LENGTH_SHORT).show();
        }
    }

    Uri uri = data.getData();
    imagePath = getRealPathFromURI(uri);
    previewImage(imagePath);
}
```

# Uploading Image- 5

---

Create the following method to get the real path of the image in the device

```
private String getRealPathFromURI(Uri uri) {  
    String[] projection = {MediaStore.Images.Media.DATA};  
    CursorLoader loader = new CursorLoader(getApplicationContext(), uri, projection, selection: null, selectionArgs: null, sortOrder:  
    Cursor cursor = loader.loadInBackground();  
    int colIndex = cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);  
    cursor.moveToFirst();  
    String result = cursor.getString(colIndex);  
    System.out.println("Image Path is " + result);  
    cursor.close();  
    return result;  
}
```



# Uploading Image- 6

Create the following method to finally upload the image

```
private void uploadImage() {

    progressDialog.show();
    File file = new File(imagePath);
    System.out.println("The image name is " + file.getName());

    ByteArrayOutputStream stream = new ByteArrayOutputStream();
    bitmap.compress(Bitmap.CompressFormat.JPEG, quality: 100, stream);
    byte[] byteArray = stream.toByteArray();
    MultipartBody.Part body = MultipartBody.Part.createFormData(
        name: "image", /* type */
        file.getName(), /*name of image file */
        RequestBody.create(MediaType.parse("image/*"), /* media type */
            byteArray /* actual content which is image here*/
        )
    );
    ImageAPI imageAPI = retrofit.create(ImageAPI.class);
    Call<ResponseBody> call = imageAPI.upload( clientId: "Client-ID 9cdb0fba5f6b1d0", body);
    call.enqueue(new Callback() {

        @Override
        public void onResponse(Call call, Response response) {
            if(response.isSuccessful()) {
                System.out.println("Response Success " + response.toString());
                progressDialog.hide();
                Toast.makeText(getApplicationContext(), text: "Successfully uploaded Image " + response.message(), Toast.LENGTH_SHORT).show();
            }
        }

        @Override
        public void onFailure(Call call, Throwable t) {
            System.out.println("Response Error" + t.getLocalizedMessage());
            progressDialog.hide();
            Toast.makeText(getApplicationContext(), text: "Error uploading image", Toast.LENGTH_SHORT).show();
        }
    });
}
```

# Uploading Image- 7

---

Add click listener on the buttons as

```
@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.btnSelect:
            if (permissionGranted) {
                Intent intent = new Intent(Intent.ACTION_PICK);
                intent.setType("image/*");
                startActivityForResult(intent, requestCode: 0);
            } else {
                Toast.makeText(getApplicationContext(), text: "Please Grant File Read Permission", Toast.LENGTH_SHORT).show();
            }

            break;
        case R.id.btnUpload:
            uploadImage();
            break;
    }
}
```