

Programmation Android

TP6 –Communication réseau sous Android

Utilisation de Retrofit dans une application Android pour récupérer des données d'une API REST.

L'objectif de ce TP Objectif est de familiariser avec l'utilisation de Retrofit, une bibliothèque très populaire pour effectuer des appels réseau dans une application Android, en insérant et récupérant des données à partir d'une API REST et les affiche dans l'interface de l'application.

Partie serveur :

- ☞ Configuration de l'environnement de développement :
- ☞ Installer et configurer un serveur web local (Apache, etc.).
- ☞ Installer et configurer un système de base de données (MySQL, etc.).
- ☞ Création d'un web service en utilisant un langage de programmation serveur (par exemple, PHP) pour se connecter à la base de données et effectuer des opérations CRUD sur une entité technologie.

Partie Android :

Etape 1 : Ajout des librairies requises

Ajouter des dépendances suivantes dans le fichier build.gradle :

```
implementation 'com.squareup.retrofit2:retrofit:2.1.0' //Bibliothèque de Retrofit
implementation 'com.squareup.retrofit2:converter-gson:2.1.0' // Bibliothèque Gson
Converter de Retrofit
```

Etape 2 : Création d'un modèle de données

Il s'agit d'une classe « technologie » qui représente les informations qu'on doit récupérer depuis l'API.

```
public class technologie {
    private int id;
    private String nom;
    ....}
```

Etape 3 : Création d'une interface de service Retrofit

On crée des méthodes pour récupérer les données de l'API :

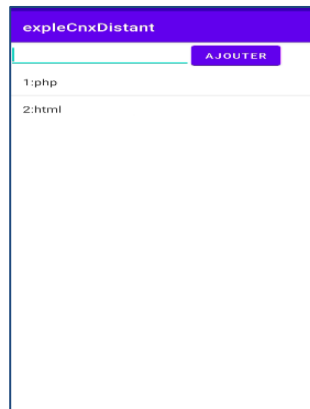
```
public interface myapi {
    //on prépare les données pour api en spécifiant quel script interrogé
    @FormUrlEncoded
    @POST("insert.php")
```

```

    Call<technologie> adtech(@Field("nom") String nom);
    @GET( "bd.php")
    Call<List<technologie>> getalltech();
}

```

Etape 4 : Création de layout de l'activité principale



Etape 5 :

Dans le MainActivity.java , vous créez le code java suivant :

- Création d'une instance de l'interface Retrofit.
- Utilisation de Retrofit pour effectuer une demande.
- Affichage des données dans l'interface (dans liste View)
- Gestion des erreurs.

```

public class MainActivity extends AppCompatActivity {
    ArrayAdapter<String> adapter=null;
    EditText edit;
    ListView listTech;
    Button btn;
    String url="http:// 10.0.2.2:80/testRetrofit/";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        listTech=findViewById(R.id.list);
        edit=findViewById(R.id.edi);
        btn=findViewById(R.id.bt);
        actualiser(); //pour afficher les données récupérées dans la liste view
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Retrofit retrofit= new Retrofit.Builder()
                    .baseUrl(url)
                    .addConverterFactory(GsonConverterFactory.create())
                    .build();
                myapi api=retrofit.create(myapi.class); //on crée une instance de l'interface
            }
        });
    }
}

```

```

        Call<technologie> call=api.adtech(edit.getText().toString());
        call.enqueue(new Callback<technologie>() {
            public void onResponse(Call<technologie> call, Response<technologie> response)
        {
            if (response.isSuccessful()) {
                technologie data = response.body();
                edit.setText(""); // Effacez le champ texte après l'insertion
                Toast.makeText(MainActivity.this, "Inséré avec succès",
                    Toast.LENGTH_SHORT).show();
                actualiser();
            } else {
                Log.e(TAG, "Échec de la réponse de Retrofit : " + response.message());
                // Gérer le cas où la réponse n'est pas réussie
            }
        }

        @Override
        public void onFailure(Call<technologie> call, Throwable t) {
            Log.e(TAG, "Erreur de rétroaction: " + t.getMessage());
            // Gérer les erreurs de réseau ou les échecs de requête
        }
    }
}

public void actualiser(){
    Retrofit retrofit = new Retrofit.Builder()
        .baseUrl(url)
        .addConverterFactory(GsonConverterFactory.create())
        .build();

    myapi api = retrofit.create(myapi.class);
    Call<List<technologie>> call = api.getalltech();
    call.enqueue(new Callback<List<technologie>>() {
        public void onResponse(Call<List<technologie>> call, Response<List<technologie>>
response) {
            List<technologie> data = response.body();
            adapter = new ArrayAdapter<>(MainActivity.this, android.R.layout.simple_list_item_1, Listin(data));
            listTech.setAdapter(adapter);
        }
    }

    @Override
    public void onFailure(Call<List<technologie>> call, Throwable t) {
        Log.e(TAG, "Erreur de rétroaction: " + t.getMessage());
    }
}

ArrayList Listin(List<technologie> l){
    ArrayList<String> maliste=new ArrayList<>();
    for(int i=0;i<l.size();i++){
        maliste.add(l.get(i).getId()+":"+l.get(i).getNom());
    }
    return maliste;
}

```