



បណ្ណបណ្ណាលក្នុង សម្បត្តិ អេច អ ឌី **Korea Software HRD Center**

Server Communication

១. អ្វីទៅជា Web Service API ?

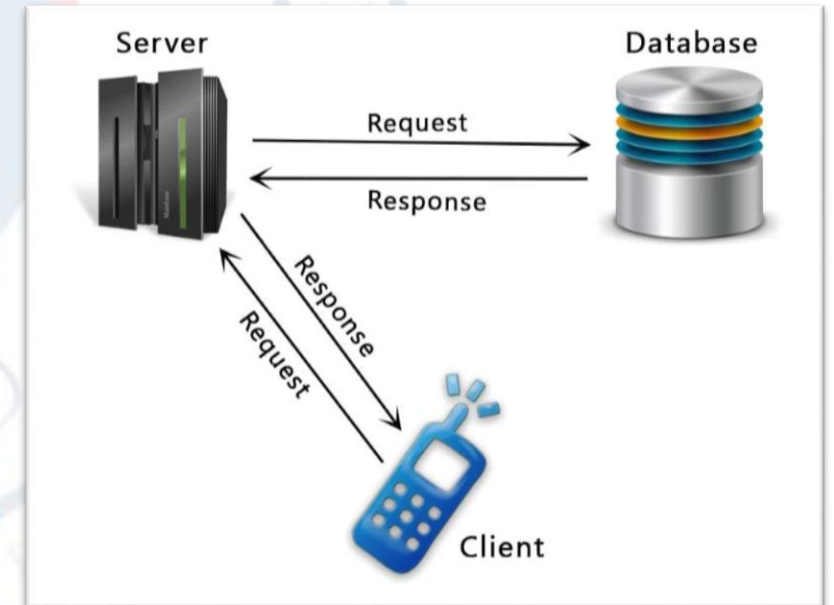
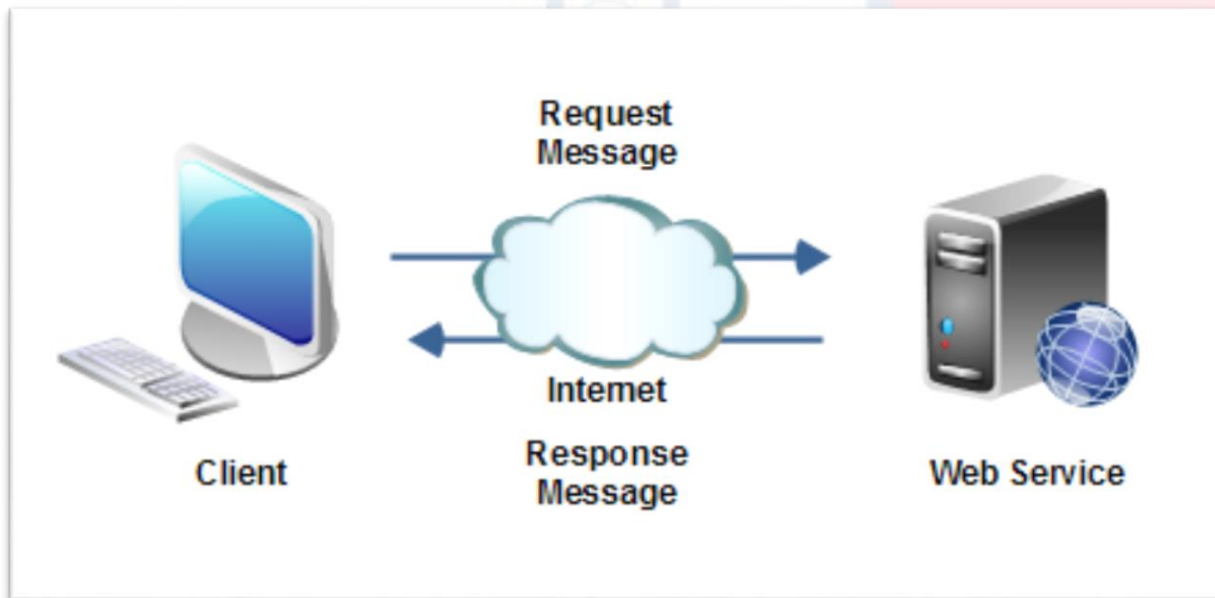
១.១. JSON Format

១.២. AsyncTask

១.៣. Retrofit

១. អ្វីទៅជា Web Service API?

- **Web service API** គឺជា web service ដែលប្រើប្រាស់ដើម្បីផ្លាស់ប្តូរទិន្នន័យទៅវិញទៅមកជាមួយ Application ឬ System ដែលបោះពុម្ពជាប្រភេទ JSON ។



9.9. JSON format?

- ❖ JSON : JavaScript Object Notation
- ❖ JSON is a syntax for storing and exchanging data
- ❖ JSON is an easier-to-use alternative to XML

```
{ "PERSON": [  
    { "ID": 1, "Name": "Phan Pirang" },  
    { "ID": 2, "Name": "John Doe" }  
]  
}
```

១.២ AsyncTask

AsyncTask ត្រូវបានប្រើដើម្បីផ្តាច់ការងាររវាង Main Thread(UI Thread) and Worker Thread (background Thread) ។

➤ ហេតុអ្វីបានជាយើងចាំបាច់ធ្វើដូច្នេះ?

ឧទាហរណ៍: យើងត្រូវការធ្វើការ request web service API

❖ ប្រើ UI thread នាំអោយ app slow, បង្កើត background thread ដើម្បីធ្វើការ

❖ ប្រើ Background thread ហើយយើងមិនអាចយក data ដែលទទួលបាន update ទៅលើ UI បាន

❖ ត្រូវសរសេរ code ដើម្បី handle បញ្ហាទាំងអស់នេះ(បញ្ហា update UI ពីក្នុង background thread)

9.2 AsyncTask (❌)

- Generic type **හිස්** **ෆ්**
- Params, the type of the parameters sent to the task upon execution.
- Progress, the type of the progress units published during the background computation.
- Result, the type of the result of the background computation.

```
class CustomAsyn extends AsyncTask<String, Integer, Boolean> {
```


១.២ AsyncTask (ត)

ដើម្បីប្រើប្រាស់ AsyncTask យើងត្រូវ បង្កើត class មួយ ដើម្បី extends ទៅកាន់ AsyncTask<Url,Progress,data> បន្ទាប់មកត្រូវ override ទៅលើ 4 method គឺ:

- ❖ onPreExecute() : ធ្វើការនៅលើ UI Thread មុនពេល Task ចាប់ផ្តើមដំណើរការ
 - ❖ doInBackground() : ធ្វើការបន្ទាប់ពី onPreExecute() នៅលើ Background Thread
 - ❖ onProgressUpdate() : ធ្វើការនៅលើ UI Thread នៅពេលដែល user call publishProgress()
 - ❖ onPostExecute() : ធ្វើការបន្ទាប់ពី doInBackground() method នៅលើ UI Thread
- ដើម្បីលុបចោលនូវ Task មួយ យើងត្រូវប្រើ cancel(boolean) method

9.3 AsyncTask (❗)

```
public class MyAsyncTask extends AsyncTask<String,Void,Bitmap> {  
    // UI Thread  
    @Override  
    protected void onPreExecute() {  
        super.onPreExecute();  
    }  
    // Background Thread  
    @Override  
    protected Bitmap doInBackground(String... strings) {  
        return loadDataFromUrl();  
    }  
    // UI Thread  
    @Override  
    protected void onProgressUpdate(Void... values) {  
        super.onProgressUpdate(values);  
    }  
    // UI Thread  
    @Override  
    protected void onPostExecute(Bitmap bitmap) {  
        super.onPostExecute(bitmap);  
    }  
}
```


9.2 AsyncTask (ଟି)

```
class ImageDown extends AsyncTask<String, Void, Bitmap> {
    @Override
    protected void onPreExecute() {
        Toast.makeText(ClassSync.this, "onPreExecute", Toast.LENGTH_SHORT).show();
    }
    @Override
    protected Bitmap doInBackground(String... strings) {
        Bitmap bmp = null;
        try{
            URL ulrn = new URL(strings[0]);
            HttpURLConnection con = (HttpURLConnection)ulrn.openConnection();
            InputStream is = con.getInputStream();
            bmp = BitmapFactory.decodeStream(is);
            if (null != bmp)
                return bmp;

        }catch(Exception e){}
        return null;
    }

    @Override
    protected void onPostExecute(Bitmap bitmap) {
        Toast.makeText(ClassSync.this, "DONE", Toast.LENGTH_SHORT).show();
        img.setImageBitmap(bitmap);
    }
}
```

១.៣ Retrofit

- Retrofit គឺជា REST Client បង្កើតដោយក្រុមហ៊ុន Square ប្រើសំរាប់ Android និង Java
- វាជា Library ដែលងាយស្រួលរៀន និង មានច្រើនសមាសធាតុ
- Method មានដូចជា GET, POST, PUT, DELETE, HEAD, PATCH ..etc
- អាចប្រើ Picasso/Glide សំរាប់ Image Loading

```
@GET("users")
```

```
@GET("users?sort=desc")
```

១.៣ Retrofit (ត)

URL MANIPULATION

```
@GET("group/{id}/users")  
Call<List<User>> groupList(@Path("id") int groupId);
```

- {id} : URL parameter
- @Path : ប្រើសម្រាប់ match {id} ជាមួយនិង method parameter

9.3 Retrofit (🐉)

```
@GET("group/{id}/users") Call<List<User>> groupList(  
    @Path("id") int groupId,  
    @Query("sort") String sort);
```

- @Query : 🐉 query parameter

```
@GET("group/{id}/users") Call<List<User>> groupList(@Path("id") int groupId,  
    @QueryMap Map<String, String> options);
```

- @QueryMap : 🐉 complex query parameters

១.៣ Retrofit (ត)

REQUEST BODY

ប្រើ object ដើម្បីធ្វើការ request body ។ ជាទូទៅគេប្រើ ជាមួយ method POST ដើម្បី pass data to server

```
@POST("users/new")
```

```
Call<User> createUser(@Body User user);
```

១.៣ Retrofit (ត)

Form Encode and multipart

Form-encode ត្រូវបាន send ទៅ server ដោយប្រើ `@FormUrlEncoded` annotation នៅលើ method។

```
@FormUrlEncoded
```

```
@POST("user/edit")
```

```
Call<User> updateUser(@Field("first_name") String first, @Field("last_name") String last);
```


9.3 Retrofit (ㄹ)

Multipart

```
@Multipart @POST("user/photo")
```

```
Call<User> updateUser(@Part("photo") RequestBody photo, @Part("description") RequestBody  
description);
```

១.៣ Retrofit (ត)

នៅក្នុង AndroidManifest.xml

Adding Internet Permission

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

នៅក្នុង Gradle

Adding Retrofit Library to Project ()

- `compile 'com.squareup.retrofit:retrofit:2.2.0'`

១.៣ Retrofit (ត)

ចាប់ data JSON ដោយប្រើ Retrofit

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

GitHubService service = retrofit.create(GitHubService.class);

Call<Repository> call = service.getRepository();

call.enqueue(new Callback<Repository>() {
    @Override
    public void onResponse(Response<Repository> response, Retrofit retrofit) {
```

9.3 Retrofit (❷)

```
try {  
  
    text_id_1.setText(" ID   : " + response.body().getId());  
    text_name_1.setText(" Name   : " + response.body().getName());  
  
} catch (Exception e) {  
    Log.d("onResponse", "There is an error");  
    e.printStackTrace();  
}  
  
}  
  
@Override  
public void onFailure(Throwable t) {  
    Log.d("onFailure", t.toString());  
}  
});
```