Android Interface Definition Language (AIDL)

AIDL (Android Interface Definition Language) is similar to other IDLs you might have worked with. It allows you to define the programming interface that both the client and service agree upon in order to communicate with each other using interprocess communication (IPC). On Android, one process cannot normally access the memory of another process. So to talk, they need to decompose their objects into primitives that the operating system can understand, and marshall the objects across that boundary for you. The code to do that marshalling is tedious to write, so Android handles it for you with AIDL.

Note: Using AIDL is necessary only if you allow clients from different applications to access your service for IPC and want to handle multithreading in your service. If you do not need to perform concurrent IPC across different applications, you should create your interface by implementing a Binder or, if you want to perform IPC, but do *not* need to handle multithreading, implement your interface using a Messenger. Regardless, be sure that you understand Bound Services before implementing an AIDL.

Before you begin designing your AIDL interface, be aware that calls to an AIDL interface are direct function calls. You should not make assumptions about the thread in which the call occurs. What happens is different depending on whether the call is from a thread in the local process or a remote process. Specifically:

- Calls made from the local process are executed in the same thread that is making the call. If this is your main UI thread, that thread continues to execute in the AIDL interface. If it is another thread, that is the one that executes your code in the service. Thus, if only local threads are accessing the service, you can completely control which threads are executing in it (but if that is the case, then you shouldn't be using AIDL at all, but should instead create the interface by implementing a Binder).
- Calls from a remote process are dispatched from a thread pool the platform maintains inside of your own process. You must be prepared for incoming calls from unknown threads, with multiple calls happening at the same time. In other words, an implementation of an AIDL interface must be completely thread-safe.
- The oneway keyword modifies the behavior of remote calls. When used, a remote call does not block; it simply sends the transaction data and immediately returns. The implementation of the interface eventually receives this as a regular call from the Binder thread pool as a normal remote call. If oneway is used with a local call, there is no impact and the call is still synchronous.

Defining an AIDL Interface

You must define your AIDL interface in an .aidl file using the Java programming language syntax, then save it in the source code (in the src/ directory) of both the application hosting the service and any other application that binds to the service.

(for example, List<String>). The actual concrete class that the other side receives is always an ArrayList, although the method is generated to use the List interface.

- Map

All elements in the Map must be one of the supported data types in this list or one of the other AIDL-generated interfaces or parcelables you've declared. Generic maps, (such as those of the form Map<String,Integer>are not supported. The actual concrete class that the other side receives is always a HashMap, although the method is generated to use the Map interface.

You must include an import statement for each additional type not listed above, even if they are defined in the same package as your interface.

When defining your service interface, be aware that:

- Methods can take zero or more parameters, and return a value or void.
- All non-primitive parameters require a directional tag indicating which way the data goes. Either in, out, orinout (see the example below).

Primitives are in by default, and cannot be otherwise.

Caution: You should limit the direction to what is truly needed, because marshalling parameters is expensive.

- All code comments included in the .aidl file are included in the generated IBinder interface (except for comments before the import and package statements).
- Only methods are supported; you cannot expose static fields in AIDL.

Here is an example .aidl file:

```
// IRemoteService.aidl
package com.example.android;

// Declare any non-default types here with import statements

/** Example service interface */
interface IRemoteService {
   /** Request the process ID of this service, to do evil things with it. */
   int getPid();

   /** Demonstrates some basic types that you can use as parameters
    * and return values in AIDL.
    */
   void basicTypes(int anInt, long aLong, boolean aBoolean, float aFloat,
```

```
        double aDouble, String aString);
}
```

Simply save your .aidl file in your project's src/ directory and when you build your application, the SDK tools generate the IBinder interface file in your project's gen/ directory. The generated file name matches the .aidlfile name, but with a .java extension (for example, IRemoteService.aidl results in IRemoteService.java).

If you use Eclipse, the incremental build generates the binder class almost immediately. If you do not use Eclipse, then the Ant tool generates the binder class next time you build your application—you should build your project with ant debug (or ant release) as soon as you're finished writing the .aidl file, so that your code can link against the generated class.

2. Implement the interface

When you build your application, the Android SDK tools generate a .java interface file named after your .aidlfile. The generated interface includes a subclass named Stub that is an abstract implementation of its parent interface (for example, YourInterface.Stub) and declares all the methods from the .aidl file.

Note: Stub also defines a few helper methods, most notably asInterface(), which takes an IBinder(usually the one passed to a client's onServiceConnected() callback method) and returns an instance of the stub interface. See the section Calling an IPC Method for more details on how to make this cast.

To implement the interface generated from the .aidl, extend the generated Binder interface (for example,YourInterface.Stub) and implement the methods inherited from the .aidl file.

Here is an example implementation of an interface called IRemoteService (defined by theIRemoteService.aidl example, above) using an anonymous instance:

```
private final IRemoteService.Stub mBinder = new IRemoteService.Stub() {
   public int getPid(){
      return Process.myPid();
   }
   public void basicTypes(int anInt, long aLong, boolean aBoolean,
      float aFloat, double aDouble, String aString) {
      // Does nothing
   }
};
```

Now the mBinder is an instance of the Stub class (a Binder), which defines the RPC interface for the service. In the next step, this instance is exposed to clients so they can interact with the service.

There are a few rules you should be aware of when implementing your AIDL interface:

Pattern 1:

1. (1/2) of a number is 3 more than the (1/6) of the same number?
a) 6 b)7 c)8 d)9

Solution:

Let the number be x,

((1/2)*x)=3+(1/6)*x,

Then solve x

2. (1/3) of a number is 3 more than the (1/6) of the same number?
a) 6 b)16 c)18 d)21

3. (1/3) of a number is 6 more than the (1/6) of the same number?
a) 6 b)18 c)36 d)24

4. (2/3) of a number is 4 more than the (1/6) of the same number?
a) 6 b)8 c)36 d)24

5. (1/3) of a number is 5 more than the (1/6) of the same number?
a) 6 b)36 c)30 d)72

Pattern 2:

1. There are two water tanks A and B, A is much smaller than B. While water fills at the rate of 1 liter every hour in A, it gets filled up like, 10, 20, 40, 80, 160 in tank B. (At the end of first hour, B has 10 liters, second hour it has 20 liters and so on). If tank B is 1/32 filled of the 21 hours, what is total duration of hours required to fill it completely?
a) 26 B)25 c)5 d)27

Solution: for every hour water in tank in B is doubled,

Let the duration to fill the tank B is x hours.

x/32 part of water in tank of B is filled in 21 hours,

Next hour it is doubled so,

2*(x/32) part i.e (x/16) part is filled in 22 hours,
Similarly (x/8)th part in 23 hours,(x/4)th part is filled in 24 hours,

(x/2)th part is filled in 25 hours, (x)th part is filled in 26 hours

So answer is 26 hours.

2. There are two pipes A and B. If A filled 10 liters in an hour, B can fill 20 liters in same time. Likewise B can fill 10, 20, 40, 80, 160. If B filled in 1/16 of a tank in 3 hours, how much time will it take to fill the tank completely?
a) 9 B) 8 c) 7 d) 6

3. There are two water tanks A and B, A is much smaller than B. While water fills at the rate of 1 liter every hour in A, it gets filled up like, 10, 20, 40,80, 160…..in tank B. 1/8 th of the tank B is filled in 22 hours. What is the time to fill the tank fully?
a) 26 B) 25 c) 5 d) 27

4. A tank is filled with water. In first hour 10 liters, second hours 20 liters, and third hour 40 liters and so on. If time taken to fill ¼ of the tank if 5 hours. What is the time taken to fill up the tank?
a) 5 B) 8 c) 7 d) 12.5

5. If a tank A can be filled within 10 hours and tank B can be filled ¼ in 19 hours then, what is the time taken to fill up the tank completely?
a) 21 B) 38 c) 57 d) 76

Pattern 3:

1. 6 persons standing in queue with different age group, after two years their average age will be 43 and seventh person joined with them. Hence the current average age has become 45. Find the age of seventh person?
a) 43 b) 69 c) 52 d) 31

Solution:

Total age of 6 persons is x hours,after two years total age of 6 persons is x+12

Average age of 6 persons is after two years is 43

So (x+12)/6=43,then solve x,

After 7th person is added then (x+7th person age)/7=45

Guidelines for filling up the online application form

S.N.	Description of Field	Input	Remarks
Personal Details>			
	Candidate's Name	Write	Maximum 46 characters Only characters are allowed As registered in class 12 or equivalent Examination
	Father's Name	Write	Maximum 46 characters Only characters are allowed As registered in class 12 or equivalent Examination. Salutations like Late, Mr., Ms., Mrs., Dr., Prof. should not used
	Mother's Name	Write	Maximum 46 characters Only characters are allowed As registered in class 12 or equivalent Examination. Salutations like Late, Mr., Ms., Mrs., Dr., Prof. should not used
	Date of Birth	Select	DD/MM/YYYY See Section 9.2 of Information Brochure
	Gender	Select	Male/ Female/ Transgender
	Domicile	Select	Assam/ Tripura/ West Bengal/ Other See section 9.3 and 15 of Information Brochure
	Category	Select	GEN/ SC/ ST/ OBC-A/ OBC-B for candidates domiciled in West Bengal GEN: for candidates from Assam/ Tripura/ Other States
	Person with Disability (PwD):	Select	Yes/ No for candidates from West Bengal No for candidates from Assam/ Tripura/ Other States See section 12.1.3 of Information Brochure
	Place of Residence	Select	1. Municipality/Corporation 2. Panchayat 3. Others
	Home District	Select	Select from drop down list
	Income Category		1. Above 10 lakhs 2. From 6.0 lakhs to below 10 lakhs 3. From 2.5 lakhs to below 6.0 lakhs 4. Below 2.5 lakhs For options 1 and 2, OBC will be treated as creamy layer

	Do you belongs to TFW?:	Auto filled	If income category given above is 4 and domicile of WB then TFW='YES' else TFW='NO'
	Religion	Select	Hindu/ Muslim/ Christian/ Buddhist/ Sikh/ Other
	Nationality	Auto filled	Only Indian citizens are eligible to apply
	Have you applied in JEE (Main) - 2016 ?	Select	Yes/ No
	Enter the Application Number	Write	Enter Application number, if applied in JEE (Main)-2016
Course Details>			
	Apply For:	Select	E - Engineering M - Medical C - Common For candidates other than WB domicile, 'E' is the only option.
	1st Choice of Exam Centre:	Select	Select examination centre zone from drop down list. Name of the examination centre will decided by the Board. See section 15 of Information Brochure and appendix-XI
	2nd Choice of Exam Centre:	Select	
	3rd Choice of Exam Centre:	Select	
Educational Details>			
10th or equivalent			
	Pass Status		Passed
	Course/Stream Name		10th
	Board/University Name		Select from drop down list
	Year Of Passing		Select from drop down list
	Marks (%)	Write	
	RollNo/Enroll No	Write	
	Institute Name, Address & Pin code	Write	
12th or equivalent			
	Pass Status		Passed/ Appearing
	Course/Stream Name		(10+2)/ ISC/ Equivalent
	Board/University Name		Select from drop down list
	Year Of Passing		Select from drop down list
	Marks (%)	Write	
	RollNo/Enroll No	Write	
	Institute Name, Address & Pin code	Write	

Communication Details>			
	Address	Write	Maximum 50 character
	Locality	Write	Maximum 50 character
	City/Town/Village	Write	
	District	Write	
	State	Write	
	Pin Code	Write	
	Email Address	Write	(30 character) and **Mandatory - NOT be changed till admission**
	Mobile Number	Write	(10 character) and **Mandatory - NOT be changed till admission**
	Land Line No. with STD code or any other Contact No.	Write	**OPTIONAL**
Pass word>			
	Choose your Password	Write	Password must be 8 to 13 characters long, must have at least one Upper case alphabet, at least one Lower case alphabet, at least one numeric value and at least one special characters like ! @ # $ % ^ & * - **DO NOT DIVULGE YOUR PASS WORD UNDER ANY CIRCUMSTANCES**
	Confirm Password	Write	Rewrite the same pass word
	Security Question	Select	Select from drop down list This question will be asked if you forget your pass word or to verify your identity
	Security Answer	Write	Write answer to the security question
Security Pin>			
	Enter security pin	Write	Has to match with the PIN displayed below
	Security Pin	View	System generated
	Reset/ Submit>	Click	Reset to start from beginning Submit to submit the application