写AIDL的步骤

1. 在代码处右键，New -> AIDL -> AIDL File

填写名字，确定。然后再main 的目录下就多了一个 aidl 文件夹，里面包含了刚刚创建的 aidl 文件。

2. 要创建一个在 aidl 中通信的类，则这个类必须实现 Parcelable 接口。例子 Book.java：

public class Book implements Parcelable {  
  
 private int bookId ;  
 private String bookName ;  
  
 protected Book(Parcel in) {  
 bookId = in.readInt();  
 bookName = in.readString();  
 }  
  
 public static final Creator<Book> CREATOR = new Creator<Book>() {  
 @Override  
 public Book createFromParcel(Parcel in) {  
 return new Book(in);  
 }  
  
 @Override  
 public Book[] newArray(int size) {  
 return new Book[size];  
 }  
 };  
  
 public int getBookId() {  
 return bookId;  
 }  
  
 public void setBookId(int bookId) {  
 this.bookId = bookId;  
 }  
  
 public String getBookName() {  
 return bookName;  
 }  
  
 public void setBookName(String bookName) {  
 this.bookName = bookName;  
 }  
  
 @Override  
 public int describeContents() {  
 return 0;  
 }  
  
 @Override  
 public void writeToParcel(Parcel dest, int flags) {  
 dest.writeInt(bookId);  
 dest.writeString(bookName);  
 }  
}

aidl 文件夹中的两个文件：

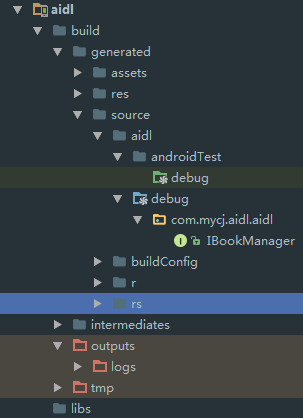
**IbookManager.aidl ：**

import com.mycj.aidl.aidl.Book;  
  
interface IBookManager {  
 /\*\*  
 \* 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);  
  
 List<Book> getBooksItem();  
  
 void addBook(in Book book);  
  
}

**Book.aidl ：这个文件对应 Book.java ，必须要有，不然编译不通过**

// Book.aidl  
package com.mycj.aidl.aidl;  
  
// Declare any non-default types here with import statements  
parcelable Book;

3. 点击菜单栏的 Build -> Make-Project 。



IbookManager的内容如下：

package com.mycj.aidl.aidl;  
public interface IBookManager extends android.os.IInterface  
{  
/\*\* Local-side IPC implementation stub class. \*/  
**public static abstract class Stub extends android.os.Binder implements com.mycj.aidl.aidl.IBookManager**  
{  
private static final java.lang.String DESCRIPTOR = "com.mycj.aidl.aidl.IBookManager";  
/\*\* Construct the stub at attach it to the interface. \*/  
public Stub()  
{  
this.attachInterface(this, DESCRIPTOR);  
}  
/\*\*  
 \* Cast an IBinder object into an com.mycj.aidl.aidl.IBookManager interface,  
 \* generating a proxy if needed.  
 \*/  
public static com.mycj.aidl.aidl.IBookManager asInterface(android.os.IBinder obj)  
{  
if ((obj==null)) {  
return null;  
}  
android.os.IInterface iin = obj.queryLocalInterface(DESCRIPTOR);  
if (((iin!=null)&&(iin instanceof com.mycj.aidl.aidl.IBookManager))) {  
return ((com.mycj.aidl.aidl.IBookManager)iin);  
}  
return new com.mycj.aidl.aidl.IBookManager.Stub.Proxy(obj);  
}  
@Override public android.os.IBinder asBinder()  
{  
return this;  
}  
@Override public boolean onTransact(int code, android.os.Parcel data, android.os.Parcel reply, int flags) throws android.os.RemoteException  
{  
switch (code)  
{  
case INTERFACE\_TRANSACTION:  
{  
reply.writeString(DESCRIPTOR);  
return true;  
}  
case TRANSACTION\_basicTypes:  
{  
data.enforceInterface(DESCRIPTOR);  
int \_arg0;  
\_arg0 = data.readInt();  
long \_arg1;  
\_arg1 = data.readLong();  
boolean \_arg2;  
\_arg2 = (0!=data.readInt());  
float \_arg3;  
\_arg3 = data.readFloat();  
double \_arg4;  
\_arg4 = data.readDouble();  
java.lang.String \_arg5;  
\_arg5 = data.readString();  
this.basicTypes(\_arg0, \_arg1, \_arg2, \_arg3, \_arg4, \_arg5);  
reply.writeNoException();  
return true;  
}  
case TRANSACTION\_getBooksItem:  
{  
data.enforceInterface(DESCRIPTOR);  
java.util.List<com.mycj.aidl.aidl.Book> \_result = this.getBooksItem();  
reply.writeNoException();  
reply.writeTypedList(\_result);  
return true;  
}  
case TRANSACTION\_addBook:  
{  
data.enforceInterface(DESCRIPTOR);  
com.mycj.aidl.aidl.Book \_arg0;  
if ((0!=data.readInt())) {  
\_arg0 = com.mycj.aidl.aidl.Book.CREATOR.createFromParcel(data);  
}  
else {  
\_arg0 = null;  
}  
this.addBook(\_arg0);  
reply.writeNoException();  
return true;  
}  
}  
return super.onTransact(code, data, reply, flags);  
}  
**private static class Proxy implements com.mycj.aidl.aidl.IbookManager // Stub的内部类**  
{  
private android.os.IBinder mRemote;  
Proxy(android.os.IBinder remote)  
{  
mRemote = remote;  
}  
@Override public android.os.IBinder asBinder()  
{  
return mRemote;  
}  
public java.lang.String getInterfaceDescriptor()  
{  
return DESCRIPTOR;  
}  
 /\*\*  
 \* Demonstrates some basic types that you can use as parameters  
 \* and return values in AIDL.  
 \*/  
@Override public void basicTypes(int anInt, long aLong, boolean aBoolean, float aFloat, double aDouble, java.lang.String aString) throws android.os.RemoteException  
{  
android.os.Parcel \_data = android.os.Parcel.obtain();  
android.os.Parcel \_reply = android.os.Parcel.obtain();  
try {  
\_data.writeInterfaceToken(DESCRIPTOR);  
\_data.writeInt(anInt);  
\_data.writeLong(aLong);  
\_data.writeInt(((aBoolean)?(1):(0)));  
\_data.writeFloat(aFloat);  
\_data.writeDouble(aDouble);  
\_data.writeString(aString);  
mRemote.transact(Stub.TRANSACTION\_basicTypes, \_data, \_reply, 0);  
\_reply.readException();  
}  
finally {  
\_reply.recycle();  
\_data.recycle();  
}  
}  
@Override public java.util.List<com.mycj.aidl.aidl.Book> getBooksItem() throws android.os.RemoteException  
{  
android.os.Parcel \_data = android.os.Parcel.obtain();  
android.os.Parcel \_reply = android.os.Parcel.obtain();  
java.util.List<com.mycj.aidl.aidl.Book> \_result;  
try {  
\_data.writeInterfaceToken(DESCRIPTOR);  
mRemote.transact(Stub.TRANSACTION\_getBooksItem, \_data, \_reply, 0);  
\_reply.readException();  
\_result = \_reply.createTypedArrayList(com.mycj.aidl.aidl.Book.CREATOR);  
}  
finally {  
\_reply.recycle();  
\_data.recycle();  
}  
return \_result;  
}  
@Override public void addBook(com.mycj.aidl.aidl.Book book) throws android.os.RemoteException  
{  
android.os.Parcel \_data = android.os.Parcel.obtain();  
android.os.Parcel \_reply = android.os.Parcel.obtain();  
try {  
\_data.writeInterfaceToken(DESCRIPTOR);  
if ((book!=null)) {  
\_data.writeInt(1);  
book.writeToParcel(\_data, 0);  
}  
else {  
\_data.writeInt(0);  
}  
mRemote.transact(Stub.TRANSACTION\_addBook, \_data, \_reply, 0);  
\_reply.readException();  
}  
finally {  
\_reply.recycle();  
\_data.recycle();  
}  
}  
}

// Stub 的三个方法的相关参数（非系统定义的）  
static final int TRANSACTION\_basicTypes = (android.os.IBinder.FIRST\_CALL\_TRANSACTION + 0);  
static final int TRANSACTION\_getBooksItem = (android.os.IBinder.FIRST\_CALL\_TRANSACTION + 1);  
static final int TRANSACTION\_addBook = (android.os.IBinder.FIRST\_CALL\_TRANSACTION + 2);  
}  
/\*\*  
 \* Demonstrates some basic types that you can use as parameters  
 \* and return values in AIDL.  
 \*/  
public void basicTypes(int anInt, long aLong, boolean aBoolean, float aFloat, double aDouble, java.lang.String aString) throws android.os.RemoteException;  
public java.util.List<com.mycj.aidl.aidl.Book> getBooksItem() throws android.os.RemoteException;  
public void addBook(com.mycj.aidl.aidl.Book book) throws android.os.RemoteException;  
}

分析： IbookManager 接口，是IDE根据我们写的 aidl 文件，生成的java代码。

