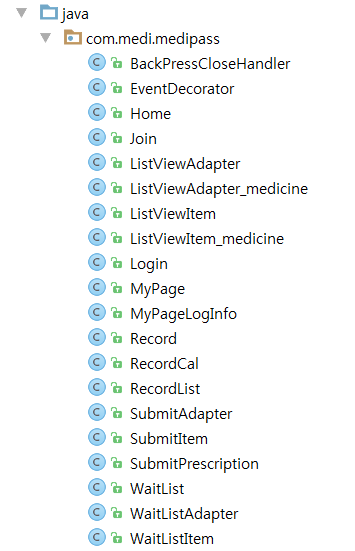
**< 파일 목록 - 아래 코드 순서대로 있음! >**



**package** com.medi.medipass;  
  
**import** android.app.Activity;  
**import** android.widget.Toast;  
  
*/\*\*  
 \* Created by shjj on 2016-04-03.  
 \*/***public class** BackPressCloseHandler {  
 **private long backkey\_pressedTime** = 0;  
 **private** Toast **toast**;  
  
 **private** Activity **activity**;  
  
 **public** BackPressCloseHandler(Activity context) {  
 **this**.**activity** = context;  
 }  
  
 *//처음 뒤로가기 버튼이 눌린 시간을 기억하고, 다음 눌린 시간이 2초이내면 종료한다.* **public void** onBackPressred() {  
 **if** (System.*currentTimeMillis*() > **backkey\_pressedTime** + 2000) {  
 **backkey\_pressedTime** = System.*currentTimeMillis*();  
 showGuide();  
 **return**;  
 }  
 **if** (System.*currentTimeMillis*() <= **backkey\_pressedTime** + 2000) {  
 **activity**.finish();  
 }  
 }  
  
 **public void** showGuide() {  
 Toast.*makeText*(**activity**, **"'뒤로'버튼 한번 더 누르시면 종료됩니다"**, Toast.***LENGTH\_SHORT***).show();  
 }  
}

**package** com.medi.medipass;  
  
**import** com.prolificinteractive.materialcalendarview.CalendarDay;  
**import** com.prolificinteractive.materialcalendarview.DayViewDecorator;  
**import** com.prolificinteractive.materialcalendarview.DayViewFacade;  
**import** com.prolificinteractive.materialcalendarview.spans.DotSpan;  
  
**import** java.util.Collection;  
**import** java.util.HashSet;  
  
*/\*\*  
 \* Created by shjj on 2016-06-03.  
 \*/***public class** EventDecorator **implements** DayViewDecorator {  
  
 **private int color**;  
 **private** HashSet<CalendarDay> **dates**;  
  
 **public** EventDecorator(**int** color, Collection<CalendarDay> dates) {  
 **this**.**color** = color;  
 **this**.**dates** = **new** HashSet<>(dates);  
 }  
  
 @Override  
 **public boolean** shouldDecorate(CalendarDay day) {  
 **return dates**.contains(day);  
 }  
  
 @Override  
 **public void** decorate(DayViewFacade view) {  
 view.addSpan(**new** DotSpan(5, **color**));  
 }  
}

**package** com.medi.medipass;  
  
**import** android.app.Activity;  
**import** android.content.Context;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.os.AsyncTask;  
**import** android.os.Build;  
**import** android.os.Bundle;  
**import** android.os.Handler;  
**import** android.provider.Settings;  
**import** android.support.v7.app.AlertDialog;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.hojung.nfc.HojungNFCReadLibrary;  
**import** com.hojung.nfc.interfaces.OnHojungNFCListener;  
**import** com.hojung.nfc.model.NfcModel;  
  
**import** java.io.BufferedReader;  
**import** java.io.BufferedWriter;  
**import** java.io.InputStreamReader;  
**import** java.io.OutputStream;  
**import** java.io.OutputStreamWriter;  
**import** java.net.HttpURLConnection;  
**import** java.net.URL;  
  
*/\*\*  
 \* Created by Elizabeth on 2016-03-25.  
 \*/***public class** Home **extends** AppCompatActivity {  
  
 */\*url\*/* String **register\_url** = **"http://condi.swu.ac.kr/Prof-Kang/2013111539/medipass/register\_wait\_list.php"**;  
 String **wait\_url** = **"http://condi.swu.ac.kr/Prof-Kang/2013111539/medipass/show\_waitnum.php"**;  
 String **my\_wait\_num\_url** = **"http://condi.swu.ac.kr/Prof-Kang/2013111539/medipass/my\_waitnum.php"**;  
 String **show\_url** = **"http://condi.swu.ac.kr/Prof-Kang/2013111539/medipass/show.php"**;  
  
 */\*태그 이름 지정\*/* **final** String **TAG** = **"NFC"**;  
 **final** String **PHP** = **"PHPR"**;  
  
 */\*대기인원 출력 위한 TextView지정\*/* TextView **tv**;  
  
 */\*대기인원 저장변수\*/* **static** String *wait\_num* = **""**;  
 **static** String *my\_waitnum* = **""**;  
 **public static** Activity *home\_activity*;  
  
 */\*nfc 병원코드 저장 변수\*/* **static** String *hospital\_code* = **"000001"**;  
  
 */\*대기인원 업데이트 위한 핸들러\*/* Handler **m\_handler**;  
 Runnable **m\_task**;  
  
  
 */\* 하단바를 이용해 home으로 올 때 직전activity를 끄기 위한 선언. activity들을 받아온다. \*/* Login **login\_activity** = (Login) Login.*login\_activity*;  
  
 **private** BackPressCloseHandler **backPressCloseHandler**;*// 뒤로가기 버튼 등록(두번 터치시 종료에 사용)  
  
  
 /\*NFC라이브러리 사용위해 선언\*/* HojungNFCReadLibrary **hojungNFCReadLibrary**;  
  
  
 Context **mContext**;  
 Boolean **recptClicked**;  
  
 *//SubmitPrescription submit\_activity = (SubmitPrescription)SubmitPrescription.submit\_activity;* @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***home***);  
  
 *home\_activity* = Home.**this**;*// 현재 activity를 변수에 넣는다.* **mContext** = **this**;  
  
 */\*핸들러 시작\*/* **m\_handler** = **new** Handler();  
 **m\_task** = **new** Runnable() {  
 @Override  
 **public void** run() {  
 */\*대기인원 업데이트 함수\*/* updateWait();  
 **m\_handler**.postDelayed(**m\_task**, 2000);  
  
 **final** Button bt\_receipt = (Button) findViewById(R.id.***home\_receipt***);  
  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(**mContext**);  
  
 builder.setTitle(**"진료안내"**)  
 .setMessage(**"진료실로 들어와주세요."**) *// 메세지 설정* .setCancelable(**false**) *// 뒤로 버튼 클릭시 취소 가능 설정* .setNegativeButton(**"확인"**, **new** DialogInterface.OnClickListener() {  
 *// 취소 버튼 클릭시 설정* **public void** onClick(DialogInterface dialog, **int** whichButton) {  
 dialog.cancel();  
 **tv**.setText(**"진료 접수"**);  
 **tv**.setTextSize(16.0f);  
 }  
 });  
  
 **final** AlertDialog dialog = builder.create(); *//다이얼로그 생성* **if** (*wait\_num* **instanceof** String) {  
 Log.*d*(**"DIALOG"**, **"wait\_num(S) :"** + *wait\_num*);  
 }  
  
 String wn = *wait\_num*;  
 Log.*d*(**"DIALOG"**, **"wn"** + wn);  
  
 **if** (wn.equals(**"0\n"**)) {  
 **m\_handler**.removeMessages(0);  
 Log.*d*(**"DIALOG"**, **"show"**);  
 bt\_receipt.setEnabled(**true**);  
 dialog.show();  
  
 } **else** {  
 Log.*d*(**"DIALOG"**, **"not zero"**);  
 **tv**.setText(**"내 번호\n"** + *wait\_num*);  
 }  
 }  
 };  
  
 */\*다이얼로그(nfc 안내 팝업 띄우기)\*/* AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);  
  
 builder.setTitle(**"병원입니다"**)  
 .setMessage(**"NFC스티커에 태그해주세요."**) *// 메세지 설정* .setCancelable(**false**) *// 뒤로 버튼 클릭시 취소 가능 설정* .setNegativeButton(**"취소"**, **new** DialogInterface.OnClickListener() {  
 *// 취소 버튼 클릭시 설정* **public void** onClick(DialogInterface dialog, **int** whichButton) {  
 dialog.cancel();  
 }  
 });  
  
 **final** AlertDialog dialog = builder.create(); *//다이얼로그 생성* **backPressCloseHandler** = **new** BackPressCloseHandler(**this**);*// 뒤로가기 버튼 객체 생성  
  
 /\* intent시 직전 activity종료 \*/* **login\_activity**.finish();  
  
  
 **if** (**login\_activity** != **null**) {  
 **login\_activity**.finish();  
 }  
  
 */\*버튼 선언\*/* **final** Button bt\_record = (Button) findViewById(R.id.***home\_record***);  
 Button bt\_myPage = (Button) findViewById(R.id.***home\_mypage***);  
 Button bt\_waitList = (Button) findViewById(R.id.***home\_check\_number***);  
 **final** Button bt\_receipt = (Button) findViewById(R.id.***home\_receipt***);  
 Button bt\_submit = (Button) findViewById(R.id.***home\_submit***);  
bt\_record.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent intent\_record = **new** Intent(getApplicationContext(), Record.**class**);  
 startActivity(intent\_record);  
 }  
 });  
  
 bt\_myPage.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent intent\_mypage = **new** Intent(getApplicationContext(), MyPage.**class**);  
 startActivity(intent\_mypage);  
 }  
 });  
  
 bt\_waitList.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent intent\_waitlist = **new** Intent(getApplicationContext(), WaitList.**class**);  
 startActivity(intent\_waitlist);  
 }  
 });  
  
 */\* 병원 접수하기 버튼 클릭시 다이얼로그 뜸\*/* bt\_receipt.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **recptClicked** = **true**; *//다이얼로그 떠있는 상태일 때에만 태그 동작하기 위한 bool값* dialog.show(); *// 알림창 띄우기* }  
 });  
  
  
 */\* 처방전 제출버튼 \*/* bt\_submit.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent intent\_submit = **new** Intent(getApplicationContext(), SubmitPrescription.**class**);  
 startActivity(intent\_submit);  
 }  
 });  
  
  
  
 */\*nfc 사용 안내\*/* android.nfc.NfcAdapter mNfcAdapter = android.nfc.NfcAdapter.*getDefaultAdapter*(**mContext**);  
  
 */\*NFC 미지원단말\*/* **if** (mNfcAdapter == **null**) {  
 Toast.*makeText*(getApplicationContext(), **"NFC를 지원하지 않는 단말기입니다."**, Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
  
 **try** {  
 */\* NFC꺼져있는 경우, NFC켜기 \*/* **if** (!mNfcAdapter.isEnabled()) {  
  
 AlertDialog.Builder alertbox = **new** AlertDialog.Builder(**mContext**);  
 alertbox.setTitle(**"Info"**);  
 alertbox.setMessage(**"본 서비스를 이용하기 위해 NFC를 사용하셔야 합니다."**);  
 alertbox.setPositiveButton(**"Turn On"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 **if** (Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***JELLY\_BEAN***) {  
 Intent intent = **new** Intent(Settings.***ACTION\_NFC\_SETTINGS***);  
 startActivity(intent);  
 } **else** {  
 Intent intent = **new** Intent(Settings.***ACTION\_WIRELESS\_SETTINGS***);  
 startActivity(intent);  
 }  
 }  
 });  
 alertbox.setNegativeButton(**"Close"**, **new** DialogInterface.OnClickListener() {  
  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
  
 }  
 });  
 alertbox.show();  
  
 }  
 } **catch** (Exception e) {  
  
 }  
  
 **hojungNFCReadLibrary** = **new** HojungNFCReadLibrary(getIntent(), Home.**this**, **new** OnHojungNFCListener() {  
  
 @Override  
 **public void** onReceiveMessage(NfcModel[] models) {  
 *//* ***TODO Auto-generated method stub* try** {  
  
 Log.*d*(**"NFC1"**, **"type : "** + models[0].getTypeStr() + **" , "** + **"payload : "** + models[0].getPayloadStr() + **" , "** + **"recptClicked : "** + **recptClicked**);  
 String spot = models[0].getTypeStr();  
 *hospital\_code* = models[0].getPayloadStr();  
  
 **if** (spot **instanceof** String) {  
  
 Log.*d*(**"NFC1"**, **"spot : "** + spot);  
 }  
  
 */\*접수하기 다이얼로그 떠있는 상태일 때\*/* **if** (**recptClicked** == **true** && spot.equals(**"hospital"**)) {  
  
 */\* NFC태그값 토스트로 띄워주고 다이얼로그 끔\*/  
 //Toast.makeText(Home.this, "type : " + models[0].getTypeStr() + " , " + "payload : " + models[0].getPayloadStr(), Toast.LENGTH\_SHORT).show();* Toast.*makeText*(Home.**this**, **"접수되었습니다."**, Toast.***LENGTH\_SHORT***).show();  
  
 **recptClicked** = **false**;  
  
 */\*대기목록에 이름 추가하기\*/* registerWaitList();  
  
 */\*접수 진행 후, 대기인원 출력, 버튼 비활성화\*/* myWaitNum();  
 *//showWaitNum();* bt\_receipt.setEnabled(**false**);  
  
 */\*다이얼로그 종료\*/* dialog.dismiss();  
  
 */\*nfc태그시 대기인원 업데이트 위한 쓰레드 시작지점\*/* **m\_handler**.postDelayed(**m\_task**, 2000);  
 */\*태그 정보\*/* Log.*d*(**"NFC"**, **"type"** + models[0].getTypeStr() + **"payload : "** + models[0].getPayloadStr());  
 } **else if** (**recptClicked** == **false** && spot.equals(**"hospital"**)) {  
 Toast.*makeText*(Home.**this**, **"접수가 완료되었습니다. 순서를 기다려주세요."**, Toast.***LENGTH\_SHORT***).show();  
 } **else** {  
 Toast.*makeText*(Home.**this**, **"병원 접수 버튼입니다. 처방전 제출하기 버튼을 눌러주세요."**, Toast.***LENGTH\_SHORT***).show();  
 }  
  
  
 } **catch** (Exception e) {  
  
 }  
  
 }  
  
 @Override  
 **public void** onError(String arg0) {  
 *//* ***TODO Auto-generated method stub*** }  
 });  
  
 }  
  
  
 **private void** initNFC() {  
 **try** {  
 Log.*d*(**"NFC"**, **"intent : "** + getIntent().getAction());  
 Intent intent = getIntent();  
 **hojungNFCReadLibrary**.onResume(intent);  
 } **catch** (Exception e) {  
  
 }  
  
 }  
  
 **public void** onResume() {  
 **super**.onResume();  
 Log.*d*(**TAG**, **"onResume"**);  
 initNFC();  
@Override  
 **protected void** onPause() {  
 **super**.onPause();  
 Log.*d*(**TAG**, **"onPause"**);  
 **hojungNFCReadLibrary**.onPause();  
 }  
  
  
 @Override  
 **public void** onNewIntent(Intent intent) {  
 Log.*d*(**TAG**, **"onNewIntent"**);  
 **hojungNFCReadLibrary**.onNewIntent(intent);  
 }  
  
 @Override  
 */\* 뒤로가기 버튼 동작 시, 두번 눌러야 꺼지게 \*/* **public void** onBackPressed() {  
 **backPressCloseHandler**.onBackPressred();  
 }  
  
  
 */\*대기목록에 등록하기\*/* **public void** registerWaitList() {  
 GettingPHP gPHP = **new** GettingPHP();  
 gPHP.execute(**register\_url**);  
 }  
  
 */\*대기인원 출력하기\*/* **public void** showWaitNum() {  
  
 GettingPHP gPHP = **new** GettingPHP();  
 gPHP.execute(**wait\_url**);  
 *//gPHP.execute(show\_url);* }  
  
 */\*내 번호 저장하기\*/* **public void** myWaitNum() {  
 GettingPHP gPHP = **new** GettingPHP();  
 gPHP.execute(**my\_wait\_num\_url**);  
 showWaitNum();  
 }  
  
 */\*대기인원 업데이트하기\*/* **public void** updateWait() {  
 showWaitNum();  
 **tv** = (TextView) findViewById(R.id.***home\_receipt***);  
 **tv**.setText(**"내 번호\n"** + *wait\_num*);  
 }  
  
  
 *//AsyncTask : thread + handler  
 //Async(비동기화) : 병렬회로. 계속 요청을 보내는 통로와 응답을 받는 통로를 따로 만들어두는 것  
 //sync(동기화) : 직렬회로. 일이 순차적으로 진행되면서 하나가 해결되면 그다음 일이 진행되는 식으로 네트워크에서는 요청(request)를 보내면 항상 응답(response)을 받아야 진행하는 방식으로 구현* **class** GettingPHP **extends** AsyncTask<String, Integer, String> { *//<Param, Progress, Result>* @Override  
 **protected void** onPreExecute() {  
 **super**.onPreExecute();  
 }  
  
 *//php에서 데이터를 읽어오는 역할, 백그라운드 스레드로 동작해야 하는 작업을 실행한다 : 필수구현  
 //execute메서드로 전달한 data tye이 params 인수로 전달되는데 여러개의 인수를 전달할 수 있으므로 배열 타입으로 되어 있다.  
 //그래서 하나의 인수만 필요하다면 params[0]만 사용하면 된다.* @Override  
 **protected** String doInBackground(String... params) {  
 Log.*d*(**PHP**, **"doInBackground "** + params[0]);  
 StringBuilder jsonHtml = **new** StringBuilder();  
 String response = **""**;  
 **try** {  
 *// URL --> openConnection() --> URLConnection --> getInputStream --> InputStream (내용읽음)* Log.*d*(**PHP**, **"back\_try"**);  
 URL phpUrl = **new** URL(params[0]);  
 HttpURLConnection conn = (HttpURLConnection) phpUrl.openConnection(); *//URL내용을 읽어오거나 GET/POST로 전달할 때 사용* **if** (conn != **null**) {  
  
 **if** (params[0].equals(**register\_url**)) {  
 **if** (conn.getResponseCode() == HttpURLConnection.***HTTP\_OK***) {  
 BufferedReader br = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream()));  
 **while** (**true**) {  
 String line = br.readLine();  
 **if** (line == **null**) **break**;  
 jsonHtml.append(line + **"\n"**);  
 Log.*d*(**"HHH"**, **"list\_line : "** + line);  
 }  
 }  
 } **else if** (params[0].equals(**wait\_url**)) {  
 String data = **"hospital\_code="** + *hospital\_code* + **"& my\_wait\_num="** + *my\_waitnum*;  
 Log.*d*(**PHP**, **"data "** + data);  
 conn.setReadTimeout(10000);  
 conn.setConnectTimeout(5000);  
 conn.setRequestMethod(**"POST"**);  
 conn.setDoInput(**true**);  
 conn.setDoOutput(**true**);  
 *//conn.setRequestProperty("Content-Type", "application/json");* conn.setUseCaches(**false**);  
  
 OutputStream os = conn.getOutputStream();  
 BufferedWriter bw = **new** BufferedWriter(**new** OutputStreamWriter(os, **"UTF-8"**));  
 bw.write(data);  
 bw.flush();  
 bw.close();  
 Log.*d*(**PHP**, **"data push end"** + data);  
  
 *//post메세지가 전송된다* conn.connect();  
  
 **if** (conn.getResponseCode() == HttpURLConnection.***HTTP\_OK***) {  
 BufferedReader br = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream()));  
 **while** (**true**) {  
 String line = br.readLine();  
 **if** (line == **null**) **break**;  
 jsonHtml.append(line + **"\n"**);  
 Log.*d*(**"HHH"**, **"list\_line\_wait : "** + line);  
  
 */\*대기인원 저장\*/  
 wait\_num* = jsonHtml.toString();  
 Log.*d*(**"hhh"**, **"waitn"** + *wait\_num*);  
 }  
 br.close();  
 }  
 } **else if** (params[0].equals(**my\_wait\_num\_url**)) {  
 String data = **"hospital\_code="** + *hospital\_code*;  
 Log.*d*(**PHP**, **"data "** + data);  
 conn.setReadTimeout(10000);  
 conn.setConnectTimeout(5000);  
 conn.setRequestMethod(**"POST"**);  
 conn.setDoInput(**true**);  
 conn.setDoOutput(**true**);  
 *//conn.setRequestProperty("Content-Type", "application/json");* conn.setUseCaches(**false**);  
  
 OutputStream os = conn.getOutputStream();  
 BufferedWriter bw = **new** BufferedWriter(**new** OutputStreamWriter(os, **"UTF-8"**));  
 bw.write(data);  
 bw.flush();  
 bw.close();  
 Log.*d*(**PHP**, **"data push end"** + data);  
  
 *//post메세지가 전송된다* conn.connect();  
  
 Log.*d*(**PHP**, **"conn"** + conn.getResponseCode());  
 **if** (conn.getResponseCode() == HttpURLConnection.***HTTP\_OK***) {  
 Log.*d*(**PHP**, **"mynumconn"**);  
 BufferedReader br = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream()));  
  
 String line;  
 **while** ((line = br.readLine()) != **null**) {  
 response += line;  
 Log.*d*(**PHP**, **"response: "** + response);  
 */\*내 번호 저장\*/  
 my\_waitnum* = response;  
 Log.*d*(**"hhh"**, **"mywaitn"** + *my\_waitnum*);  
 }  
 br.close();  
 }  
 }  
 }  
 conn.disconnect();  
} **catch** (Exception e) {  
 Log.*d*(**PHP**, **"Error"**);  
 e.printStackTrace();  
 }  
 Log.*d*(**PHP**, **"end of doInBackground : "** + jsonHtml.toString());  
  
 **return** jsonHtml.toString();  
 }  
  
 *//가져온 데이터를 이용해 원하는 일을 하도록 한다* @Override  
 **protected void** onPostExecute(String str) {  
 Log.*d*(**PHP**, **"onPostExecute"** + str);  
 **try** {  
  
 Log.*d*(**PHP**, **"post\_try"**);  
  
 */\*대기인원 지정\*/  
 //int wt;  
 //wt=Integer.parseInt(wait\_num);  
 //wt=wt-1;  
 //Log.d(PHP, "Wt : "+String.valueOf(wt));* **tv** = (TextView) findViewById(R.id.***home\_receipt***);  
 **tv**.setText(**"내 번호\n"** + *wait\_num*);  
 **tv**.setTextSize(30.0f);  
  
  
 } **catch** (Exception e) {  
 Log.*d*(**PHP**, **"onPost Error"**);  
 e.printStackTrace();  
 }  
 }  
 }  
  
  
}

**package** com.medi.medipass;  
  
**import** android.os.Bundle;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.view.View;  
**import** android.widget.Button;  
  
*/\*\*  
 \* Created by shjj on 2016-03-25.  
 \*/***public class** Join **extends** AppCompatActivity {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***join***);  
  
 Button bt\_back = (Button) findViewById(R.id.***join\_back***);  
 bt\_back.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 finish();  
 }  
 });  
 }  
}

**package** com.medi.medipass;  
  
**import** android.content.Context;  
**import** android.util.Log;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.BaseAdapter;  
**import** android.widget.TextView;  
  
**import** java.util.ArrayList;  
  
*/\*\*  
 \* Created by shjj on 2016-04-14.  
 \*/***public class** ListViewAdapter **extends** BaseAdapter {  
 Context **mcontext** = **null**;  
  
 *//Adapter에 추가된 데이터를 저장하기 위한 ArrayList* **private** ArrayList<ListViewItem> **listViewItemList** = **new** ArrayList<ListViewItem>();  
  
 *//ListViewAdapter의 생성자* **public** ListViewAdapter() {  
 }  
  
 *//Adapter에 사용되는 데이터의 개수를 리턴 : 필수구현* @Override  
 **public int** getCount() {  
 **return listViewItemList**.size();  
 }  
  
 *//position에 위치한 데이터를 화면에 출력하는데 사용될 view를 리턴 : 필수구현* @Override  
 **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 **final int** pos = position;  
 **mcontext** = parent.getContext();  
  
 *//listview\_item 레이아웃을 inflate하여 convertView 참조 획득* **if** (convertView == **null**) {  
 LayoutInflater inflater = (LayoutInflater) **mcontext**.getSystemService(**mcontext**.***LAYOUT\_INFLATER\_SERVICE***);  
 convertView = inflater.inflate(R.layout.***listview\_item***, parent, **false**);  
  
 convertView.setTag(position);  
 }  
  
 *//화면에 표시될 view(layout이 inflate된)으로부터 위젯에 대한 참조 획득* TextView item\_date = (TextView) convertView.findViewById(R.id.***item\_date***);  
 TextView item\_disName = (TextView) convertView.findViewById(R.id.***item\_disName***);  
  
 *//Data Set(ListViewItem)에서 position에 위치한 데이터 참조 획득* ListViewItem listviewItem = **listViewItemList**.get(position);  
  
 *//아이템 내 각 위젯이 데이터 반영* item\_date.setText(listviewItem.getItem\_date());  
 item\_disName.setText(listviewItem.getItem\_disName());  
  
 **return** convertView;  
 }  
  
 *//지정한 위치(position)에 있는 데이터와 아이템(row)의 ID를 리턴 : 필수구현* @Override  
 **public long** getItemId(**int** position) {  
 **return** position;  
 }  
  
 *//지정한 위치(position)에 있는 데이터 리턴 : 필수구현* @Override  
 **public** Object getItem(**int** position) {  
 **return listViewItemList**.get(position);  
 }  
  
  
 *//아이템 데이터 추가를 위한 함수. 개발자가 원하는대로 작성 가능.* **public void** addItem(String date, String disName) {  
 Log.*d*(**"PHP"**, **"addItem"**);  
 ListViewItem item = **new** ListViewItem();  
  
 item.setItem\_date(date);  
 item.setItem\_disName(disName);  
  
 **listViewItemList**.add(item);  
 notifyDataSetChanged();  
 }  
  
 **public void** init() {  
 **listViewItemList**.clear();  
 }  
  
}

**package** com.medi.medipass;  
  
**import** android.content.Context;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.BaseAdapter;  
**import** android.widget.TextView;  
  
**import** java.util.ArrayList;  
  
*/\*\*  
 \* Created by shjj on 2016-04-14.  
 \*/***public class** ListViewAdapter\_medicine **extends** BaseAdapter {  
 Context **mcontext** = **null**;  
  
 *//Adapter에 추가된 데이터를 저장하기 위한 ArrayList* **private** ArrayList<ListViewItem\_medicine> **listViewItemList\_medicine** = **new** ArrayList<ListViewItem\_medicine>();  
  
 *//ListViewAdapter의 생성자* **public** ListViewAdapter\_medicine() {  
 }  
  
 *//Adapter에 사용되는 데이터의 개수를 리턴 : 필수구현* @Override  
 **public int** getCount() {  
 **return listViewItemList\_medicine**.size();  
 }  
  
 *//position에 위치한 데이터를 화면에 출력하는데 사용될 view를 리턴 : 필수구현* @Override  
 **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 **final int** pos = position;  
 **mcontext** = parent.getContext();  
  
 *//listview\_item 레이아웃을 inflate하여 convertView 참조 획득* **if** (convertView == **null**) {  
 LayoutInflater inflater = (LayoutInflater) **mcontext**.getSystemService(**mcontext**.***LAYOUT\_INFLATER\_SERVICE***);  
 convertView = inflater.inflate(R.layout.***listview\_item\_medicine***, parent, **false**);  
 }  
  
 *//화면에 표시될 view(layout이 inflate된)으로부터 위젯에 대한 참조 획득* TextView item\_medName = (TextView) convertView.findViewById(R.id.***listview\_medName***);  
 TextView item\_onceNum = (TextView) convertView.findViewById(R.id.***listview\_once\_num***);  
 TextView item\_dayNum = (TextView) convertView.findViewById(R.id.***listview\_day\_num***);  
 TextView item\_notice = (TextView) convertView.findViewById(R.id.***listview\_notice***);  
  
 *//Data Set(ListViewItem)에서 position에 위치한 데이터 참조 획득* ListViewItem\_medicine listviewItem = **listViewItemList\_medicine**.get(position);  
  
 item\_medName.setText(listviewItem.getMedName());  
 item\_onceNum.setText(String.*valueOf*(listviewItem.getOnce\_num()));  
 item\_dayNum.setText(String.*valueOf*(listviewItem.getDay\_num()));  
 item\_notice.setText(listviewItem.getNotice());  
  
 **return** convertView;  
 }  
  
 *//지정한 위치(position)에 있는 데이터와 아이템(row)의 ID를 리턴 : 필수구현* @Override  
 **public long** getItemId(**int** position) {  
 **return** position;  
 }  
  
 *//지정한 위치(position)에 있는 데이터 리턴 : 필수구현* @Override  
 **public** Object getItem(**int** position) {  
 **return listViewItemList\_medicine**.get(position);  
 }  
  
 *//아이템 데이터 추가를 위한 함수. 개발자가 원하는대로 작성 가능.* **public void** addItem(String medName, **int** onceNum, **int** dayNum, String notice) {  
 ListViewItem\_medicine item = **new** ListViewItem\_medicine();  
  
 item.setMedName(medName);  
 item.setOnce\_num(onceNum);  
 item.setDay\_num(dayNum);  
 item.setNotice(notice);  
  
 **listViewItemList\_medicine**.add(item);  
 notifyDataSetChanged();  
 }  
  
 **public void** init() {  
 **listViewItemList\_medicine**.clear();  
 }  
  
}

**package** com.medi.medipass;  
  
*/\*\*  
 \* Created by shjj on 2016-04-14.  
 \*/***public class** ListViewItem {  
 **private** String **item\_date**;  
 **private** String **item\_disName**;  
  
 **public void** setItem\_date(String item\_date) {  
 **this**.**item\_date** = item\_date;  
 }  
  
 **public void** setItem\_disName(String item\_disName) {  
 **this**.**item\_disName** = item\_disName;  
 }  
  
 **public** String getItem\_date() {  
 **return item\_date**;  
 }  
  
 **public** String getItem\_disName() {  
 **return item\_disName**;  
 }  
}

**package** com.medi.medipass;  
  
*/\*\*  
 \* Created by shjj on 2016-05-10.  
 \*/***public class** ListViewItem\_medicine {  
 String **medName**;  
 **int once\_num**;  
 **int day\_num**;  
 String **notice**;  
  
 **public** String getMedName() {  
 **return medName**;  
 }  
  
 **public int** getOnce\_num() {  
 **return once\_num**;  
 }  
  
 **public int** getDay\_num() {  
 **return day\_num**;  
 }  
  
 **public** String getNotice() {  
 **return notice**;  
 }  
  
 **public void** setMedName(String medName) {  
 **this**.**medName** = medName;  
 }  
  
 **public void** setOnce\_num(**int** once\_num) {  
 **this**.**once\_num** = once\_num;  
 }  
  
 **public void** setDay\_num(**int** day\_num) {  
 **this**.**day\_num** = day\_num;  
 }  
  
 **public void** setNotice(String notice) {  
 **this**.**notice** = notice;  
 }  
}

**package** com.medi.medipass;  
  
**import** android.app.Activity;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.view.View;  
**import** android.widget.Button;  
  
*/\*\*  
 \* Created by shjj on 2016-03-25.  
 \*/***public class** Login **extends** AppCompatActivity {  
  
 **public static** Activity *login\_activity*;*// 해당 activity를 담을 변수* **private** BackPressCloseHandler **backPressCloseHandler**;*// 뒤로가기 버튼 등록(두번 터치시 종료에 사용)* @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***login***);  
  
 *login\_activity* = Login.**this**;*// 현재 activity를 변수에 넣는다* **backPressCloseHandler** = **new** BackPressCloseHandler(**this**);*// 뒤로가기 버튼 객체 생성성  
  
 /\* 로그인하기, 회원가입하기 버튼 등록 \*/* Button bt\_login = (Button) findViewById(R.id.***bt\_login***);  
 Button bt\_join = (Button) findViewById(R.id.***bt\_join***);  
 bt\_login.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent intent\_login = **new** Intent(getApplicationContext(), Home.**class**);  
 startActivity(intent\_login);  
 }  
 });  
 bt\_join.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent intent\_join = **new** Intent(getApplicationContext(), Join.**class**);  
 startActivity(intent\_join);  
 }  
 });  
 }  
  
 */\* 뒤로가기 버튼 동작 시, 두번 눌러야 꺼지게 \*/* @Override  
 **public void** onBackPressed() {  
 **backPressCloseHandler**.onBackPressred();  
 }  
}

**package** com.medi.medipass;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.view.View;  
**import** android.widget.Button;  
  
*/\*\*  
 \* Created by shjj on 2016-03-25.  
 \*/***public class** MyPage **extends** AppCompatActivity **implements** View.OnClickListener {  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***my\_page***);  
  
 */\* xml의 버튼을 찾아와서 임시객체에 등록 \*/* Button myPage\_logInfo = (Button) findViewById(R.id.***mypage\_info***);  
 Button bt\_record = (Button) findViewById(R.id.***bottom\_mypage\_record***);  
 Button bt\_home = (Button) findViewById(R.id.***bottom\_mypage\_home***);  
 Button bt\_mypage = (Button) findViewById(R.id.***bottom\_mypage\_mypage***);  
  
 */\* onClick은 밑에 따로 메소드로 구현 \*/* myPage\_logInfo.setOnClickListener(**this**);  
 bt\_record.setOnClickListener(**this**);  
 bt\_home.setOnClickListener(**this**);  
 bt\_mypage.setOnClickListener(**this**);  
  
 bt\_mypage.setSelected(**true**);  
 }  
  
 */\* 뒤로가기 버튼 동작 시, Home으로 가기 \*/* @Override  
 **public void** onBackPressed() {  
 Intent intent\_home = **new** Intent(getApplicationContext(), Home.**class**);  
 intent\_home.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TOP*** | Intent.***FLAG\_ACTIVITY\_SINGLE\_TOP***);  
 startActivity(intent\_home);  
 }  
  
 @Override  
 **public void** onClick(View v) {  
 **switch** (v.getId()) {  
 **case** R.id.***mypage\_info***:  
 Intent intent\_mypage\_loginfo = **new** Intent(getApplicationContext(), MyPageLogInfo.**class**);  
 startActivity(intent\_mypage\_loginfo);  
 **break**;  
 **case** R.id.***bottom\_mypage\_record***:  
 Intent intent\_record = **new** Intent(getApplicationContext(), Record.**class**);  
 intent\_record.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TOP*** | Intent.***FLAG\_ACTIVITY\_SINGLE\_TOP***);  
 startActivity(intent\_record);  
 **break**;  
 **case** R.id.***bottom\_mypage\_home***:  
 Intent intent\_home = **new** Intent(getApplicationContext(), Home.**class**);  
 intent\_home.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TOP*** | Intent.***FLAG\_ACTIVITY\_SINGLE\_TOP***);  
 startActivity(intent\_home);  
 **break**;  
 **case** R.id.***bottom\_mypage\_mypage***:  
 Intent intent\_mypage = **new** Intent(getApplicationContext(), MyPage.**class**);  
 intent\_mypage.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TOP*** | Intent.***FLAG\_ACTIVITY\_SINGLE\_TOP***);  
 startActivity(intent\_mypage);  
 **break**;  
 }  
 }  
}

**package** com.medi.medipass;  
  
**import** android.content.Intent;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
  
*/\*\*  
 \* Created by shjj on 2016-03-25.  
 \*/***public class** MyPageLogInfo **extends** AppCompatActivity **implements** View.OnClickListener {  
  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***my\_page\_loginfo***);  
  
 */\* xml의 버튼을 찾아와서 임시객체에 등록 \*/* Button bt\_record = (Button) findViewById(R.id.***bottom\_mypage\_loginfo\_record***);  
 Button bt\_home = (Button) findViewById(R.id.***bottom\_mypage\_loginfo\_home***);  
 Button bt\_mypage = (Button) findViewById(R.id.***bottom\_mypage\_loginfo\_mypage***);  
  
 */\* onClick은 밑에 따로 메소드로 구현 \*/* bt\_record.setOnClickListener(**this**);  
 bt\_home.setOnClickListener(**this**);  
 bt\_mypage.setOnClickListener(**this**);  
  
 bt\_mypage.setSelected(**true**);  
 }  
  
 @Override  
 **public void** onClick(View v) {  
 **switch** (v.getId()) {  
 **case** R.id.***bottom\_mypage\_loginfo\_record***:  
 Intent intent\_record = **new** Intent(getApplicationContext(), Record.**class**);  
 intent\_record.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TOP*** | Intent.***FLAG\_ACTIVITY\_SINGLE\_TOP***);  
 startActivity(intent\_record);  
 **break**;  
 **case** R.id.***bottom\_mypage\_loginfo\_home***:  
 Intent intent\_home = **new** Intent(getApplicationContext(), Home.**class**);  
 intent\_home.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TOP*** | Intent.***FLAG\_ACTIVITY\_SINGLE\_TOP***);  
 startActivity(intent\_home);  
 **break**;  
 **case** R.id.***bottom\_mypage\_loginfo\_mypage***:  
 Intent intent\_mypage = **new** Intent(getApplicationContext(), MyPage.**class**);  
 intent\_mypage.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TOP*** | Intent.***FLAG\_ACTIVITY\_SINGLE\_TOP***);  
 startActivity(intent\_mypage);  
 **break**;  
 }  
 }  
}

**package** com.medi.medipass;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.support.v4.app.Fragment;  
**import** android.support.v4.app.FragmentTransaction;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.view.View;  
**import** android.widget.Button;  
  
*/\*\*  
 \* Created by shjj on 2016-05-03.  
 \*/***public class** Record **extends** AppCompatActivity **implements** View.OnClickListener {  
 View **frag\_record**; *// fragment View* **public final int FRAGMENT\_RECORD\_LIST** = 1;  
 **public final int FRAGMENT\_RECORD\_CAL** = 2;  
  
 **int current\_fragment\_index** = **FRAGMENT\_RECORD\_LIST**;*// 기본화면은 list* Button **bt\_list**, **bt\_cal**;  
  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***record***);  
  
 **frag\_record** = (View) findViewById(R.id.***ll\_fragment\_record***);  
  
 */\* xml의 버튼을 찾아와서 임시객체에 등록 \*/* **bt\_list** = (Button) findViewById(R.id.***up\_menu\_list***);  
 **bt\_cal** = (Button) findViewById(R.id.***up\_menu\_cal***);  
 Button bt\_record = (Button) findViewById(R.id.***bottom\_record\_record***);  
 Button bt\_home = (Button) findViewById(R.id.***bottom\_record\_home***);  
 Button bt\_mypage = (Button) findViewById(R.id.***bottom\_record\_mypage***);  
  
 */\* onClick은 밑에 따로 메소드로 구현 \*/* bt\_record.setOnClickListener(**this**);  
 bt\_home.setOnClickListener(**this**);  
 bt\_mypage.setOnClickListener(**this**);  
 **bt\_list**.setOnClickListener(**this**);  
 **bt\_cal**.setOnClickListener(**this**);  
  
 **bt\_list**.setSelected(**true**);  
 bt\_record.setSelected(**true**);  
  
 fragmentReplace(**current\_fragment\_index**);  
 }  
  
 */\* 넘겨받은 fragment index를 가지고 getFragment메소드를 가서 실질적인 fragment의 객체 생성을 하고 그 객체를 리턴 \*/* **public void** fragmentReplace(**int** new\_fragment\_index) {  
 Fragment new\_fragment = **null**;  
 new\_fragment = getFragment(new\_fragment\_index);  
  
 */\* replace fragment \*/* **final** FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();  
 transaction.replace(R.id.***ll\_fragment\_record***, new\_fragment);  
  
 */\* commit the transaction(변경사항을 적용) \*/* transaction.commit();  
 }  
  
 **private** Fragment getFragment(**int** idx) {  
 Fragment newFragment = **null**;  
  
 **switch** (idx) {  
 **case FRAGMENT\_RECORD\_LIST**:  
 newFragment = **new** RecordList();  
 **break**;  
 **case FRAGMENT\_RECORD\_CAL**:  
 newFragment = **new** RecordCal();  
 **break**;  
 **default**:  
 **break**;  
 }  
  
 **return** newFragment;  
 }  
  
 @Override  
 **public void** onClick(View v) {  
 **switch** (v.getId()) {  
 **case** R.id.***bottom\_record\_record***:  
 Intent intent\_record = **new** Intent(getApplicationContext(), Record.**class**);  
 intent\_record.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TOP*** | Intent.***FLAG\_ACTIVITY\_SINGLE\_TOP***);  
 startActivity(intent\_record);  
 **break**;  
 **case** R.id.***bottom\_record\_home***:  
 Intent intent\_home = **new** Intent(getApplicationContext(), Home.**class**);  
 intent\_home.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TOP*** | Intent.***FLAG\_ACTIVITY\_SINGLE\_TOP***);  
 startActivity(intent\_home);  
 **break**;  
 **case** R.id.***bottom\_record\_mypage***:  
 Intent intent\_mypage = **new** Intent(getApplicationContext(), MyPage.**class**);  
 intent\_mypage.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TOP*** | Intent.***FLAG\_ACTIVITY\_SINGLE\_TOP***);  
 startActivity(intent\_mypage);  
 **break**;  
 **case** R.id.***up\_menu\_list***:  
 **bt\_list**.setSelected(**true**);  
 **bt\_cal**.setSelected(**false**);  
 **current\_fragment\_index** = **FRAGMENT\_RECORD\_LIST**;  
 fragmentReplace(**current\_fragment\_index**);  
 **break**;  
 **case** R.id.***up\_menu\_cal***:  
 **bt\_list**.setSelected(**false**);  
 **bt\_cal**.setSelected(**true**);  
 **current\_fragment\_index** = **FRAGMENT\_RECORD\_CAL**;  
 fragmentReplace(**current\_fragment\_index**);  
 **break**;  
 }  
 }  
  
 */\* 뒤로가기 버튼 동작 시, Home으로 가기 \*/  
 //http://diyall.tistory.com/781  
 //http://comxp.tistory.com/109* @Override  
 **public void** onBackPressed() {  
 Intent intent\_home = **new** Intent(getApplicationContext(), Home.**class**);  
 intent\_home.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TOP*** | Intent.***FLAG\_ACTIVITY\_SINGLE\_TOP***);  
 startActivity(intent\_home);  
 }  
}

**package** com.medi.medipass;  
  
**import** android.app.AlertDialog;  
**import** android.graphics.Color;  
**import** android.os.AsyncTask;  
**import** android.os.Bundle;  
**import** android.support.annotation.NonNull;  
**import** android.support.annotation.Nullable;  
**import** android.support.v4.app.Fragment;  
**import** android.util.Log;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.AdapterView;  
**import** android.widget.ListView;  
**import** android.widget.TextView;  
  
**import** com.prolificinteractive.materialcalendarview.CalendarDay;  
**import** com.prolificinteractive.materialcalendarview.MaterialCalendarView;  
**import** com.prolificinteractive.materialcalendarview.OnDateSelectedListener;  
  
**import** org.json.JSONArray;  
**import** org.json.JSONException;  
**import** org.json.JSONObject;  
  
**import** java.io.BufferedReader;  
**import** java.io.BufferedWriter;  
**import** java.io.InputStreamReader;  
**import** java.io.OutputStream;  
**import** java.io.OutputStreamWriter;  
**import** java.net.HttpURLConnection;  
**import** java.net.URL;  
**import** java.text.DateFormat;  
**import** java.text.SimpleDateFormat;  
**import** java.util.ArrayList;  
**import** java.util.Calendar;  
  
**import** butterknife.Bind;  
**import** butterknife.ButterKnife;  
  
*/\*\*  
 \* Created by shjj on 2016-05-03.  
 \*/***public class** RecordCal **extends** Fragment **implements** OnDateSelectedListener {  
 **private static final** DateFormat ***FORMATTER*** = SimpleDateFormat.*getDateInstance*();  
  
 String **url\_list** = **"http://condi.swu.ac.kr/Prof-Kang/2013111539/medipass/write\_app.php"**;  
 String **url\_medicine** = **"http://condi.swu.ac.kr/Prof-Kang/2013111539/medipass/write\_medicine.php"**;  
 GettingPHP **gPHP**;  
  
 ArrayList<CalendarDay> **dates** = **new** ArrayList<>();  
  
  
 @Bind(R.id.***calendarView***)  
 MaterialCalendarView **widget**;  
  
 **final** ListViewAdapter **adapter** = **new** ListViewAdapter();  
  
 **private static** ArrayList<String> *dateArrayList* = **new** ArrayList<String>();  
 **private static** ArrayList<String> *nameArrayList* = **new** ArrayList<String>();  
 **int num** = 0;  
  
 ListViewAdapter\_medicine **med\_adapter** = **new** ListViewAdapter\_medicine();  
 **static** String *date*, *disName*;  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.***record\_cal***, container, **false**);  
  
 ButterKnife.*bind*(**this**, view);  
 **widget**.setOnDateChangedListener(**this**);  
  
 **final** ListView listview = (ListView) view.findViewById(R.id.***listview\_cal***);  
 listview.setAdapter(**adapter**);  
  
 **gPHP** = **new** GettingPHP();  
 **gPHP**.execute(**url\_list**);  
  
 *//클릭 이벤트 정의* listview.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(AdapterView parent, View v, **int** position, **long** id) {  
 ListViewItem item = (ListViewItem) parent.getItemAtPosition(position);  
  
 View dialogView = (View) v.*inflate*(getContext(), R.layout.***record\_item\_click***, **null**);  
 ListView listview\_med\_cal = (ListView) dialogView.findViewById(R.id.***record\_item\_click\_listView***);  
 listview\_med\_cal.setAdapter(**med\_adapter**);  
  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(getContext());  
 builder.setTitle(**"처방목록"**);  
 builder.setView(dialogView);  
 builder.setPositiveButton(**"확인"**, **null**);  
  
 *date* = item.getItem\_date();  
 *disName* = item.getItem\_disName();  
  
 *//php를 읽어올때 사용할 변수* GettingPHP mPHP = **new** GettingPHP();  
 mPHP.execute(**url\_medicine**);  
  
 TextView tvdate = (TextView) dialogView.findViewById(R.id.***textView***);  
 tvdate.setText(item.getItem\_date());  
  
 builder.show();  
 }  
 });  
  
 **return** view;  
 }  
  
 @Override  
 **public void** onDateSelected(@NonNull MaterialCalendarView widget, @Nullable CalendarDay date, **boolean** selected) {  
 **adapter**.init();  
 **for** (**int** i = 0; i < **num**; i++) {  
 **if** (*dateArrayList*.get(i).equals(getSelectedDatesString())) {  
 **adapter**.addItem(*dateArrayList*.get(i), *nameArrayList*.get(i));  
 }  
 }  
 }  
  
 **private** String getSelectedDatesString() {  
 CalendarDay date = **widget**.getSelectedDate();  
 **if** (date == **null**) {  
 **return "No Selection"**;  
 }  
 **return *FORMATTER***.format(date.getDate());  
 }  
  
  
 **class** GettingPHP **extends** AsyncTask<String, Integer, String> { *//<Param, Progress, Result(doInBackground의 반환값, onPostExcute의 매개변수)>* @Override  
 **protected void** onPreExecute() {  
 **super**.onPreExecute();  
 }  
  
 *//php에서 데이터를 읽어오는 역할, 백그라운드 스레드로 동작해야 하는 작업을 실행한다 : 필수구현  
 //execute메서드로 전달한 data tye이 params 인수로 전달되는데 여러개의 인수를 전달할 수 있으므로 배열 타입으로 되어 있다.  
 //그래서 하나의 인수만 필요하다면 params[0]만 사용하면 된다.* @Override  
 **protected** String doInBackground(String... params) {  
 StringBuilder jsonHtml = **new** StringBuilder();  
 **try** {  
 *// URL --> openConnection() --> URLConnection --> getInputStream --> InputStream (내용읽음)* URL phpUrl = **new** URL(params[0]);  
 HttpURLConnection conn = (HttpURLConnection) phpUrl.openConnection(); *//URL내용을 읽어오거나 GET/POST로 전달할 때 사용* **if** (conn != **null**) {  
 **if** (params[0].equals(**url\_medicine**)) {  
 String data = **"date="** + *date* + **"& disName="** + *disName*;  
 Log.*d*(**"HHH"**, **"data : "** + data);  
  
 conn.setReadTimeout(10000);  
 conn.setConnectTimeout(5000);  
 conn.setRequestMethod(**"POST"**);  
 conn.setDoInput(**true**);  
 conn.setDoOutput(**true**);  
 conn.setUseCaches(**false**);  
  
 OutputStream os = conn.getOutputStream();  
 BufferedWriter bw = **new** BufferedWriter(**new** OutputStreamWriter(os, **"UTF-8"**));  
 bw.write(data);  
 bw.flush();  
 bw.close();  
  
 *//post메세지가 전송된다* conn.connect();  
 }  
  
 **if** (conn.getResponseCode() == HttpURLConnection.***HTTP\_OK***) {  
 BufferedReader br = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream()));  
 **while** (**true**) {  
 String line = br.readLine();  
 **if** (line == **null**) **break**;  
 jsonHtml.append(line + **"\n"**);  
 }  
 br.close();  
 }  
 }  
 conn.disconnect();  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 **return** jsonHtml.toString();  
 }  
  
 *//가져온 데이터를 이용해 원하는 일을 하도록 한다* @Override  
 **protected void** onPostExecute(String str) {  
 **try** {  
 *//php에서 받아온 JSON데이터를 JSON오브젝트로 변환* JSONObject jobject = **new** JSONObject(str);  
 *//results라는 key는 JSON배열로 되어있다* JSONArray results = jobject.getJSONArray(**"results"**);  
  
 **if** (jobject.get(**"status"**).equals(**"list"**)) {  
 Log.*d*(**"HHH"**, **"list"**);  
 **int** j = 0;  
 *//Calendar calendar = Calendar.getInstance();* **for** (**int** i = 0; i < results.length(); i++) { *//length->child의 갯수* JSONObject temp = results.getJSONObject(i);  
  
 *//http://hyeonstorage.tistory.com/205* Calendar calendar = Calendar.*getInstance*();  
 Log.*d*(**"ddd"**, temp.getString(**"date\_cal"**).substring(0, 4) + **" + "** + temp.getString(**"date\_cal"**).substring(4, 6) + **" + "** + temp.getString(**"date\_cal"**).substring(6, 8));  
 calendar.set(Integer.*parseInt*(temp.getString(**"date\_cal"**).substring(0, 4)),  
 Integer.*parseInt*(temp.getString(**"date\_cal"**).substring(4, 6)) - 1,  
 Integer.*parseInt*(temp.getString(**"date\_cal"**).substring(6, 8)));  
  
 CalendarDay day = CalendarDay.*from*(calendar);  
 Log.*d*(**"ddd"**, **"day : "** + day);  
  
 **dates**.add(day);  
  
 *dateArrayList*.add(temp.getString(**"record\_date"**));  
 *nameArrayList*.add(temp.getString(**"disease\_name"**));  
 j++;  
 }  
 **widget**.addDecorator(**new** EventDecorator(Color.***RED***, **dates**));  
 **num** = j;  
 }  
  
 **if** (jobject.get(**"status"**).equals(**"medicine"**)) {  
 Log.*d*(**"HHH"**, **"medicine"**);  
 **med\_adapter**.init();  
 **for** (**int** i = 0; i < results.length(); i++) { *//length->child의 갯수* JSONObject temp = results.getJSONObject(i);  
 String medName = temp.getString(**"medName"**);  
 **int** onceNum = temp.getInt(**"onceNum"**);  
 **int** dayNum = temp.getInt(**"dayNum"**);  
 String notice = temp.getString(**"notice"**);  
 **med\_adapter**.addItem(medName, onceNum, dayNum, notice);  
 }  
 }  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
}

**package** com.medi.medipass;  
  
**import** android.content.Context;  
**import** android.util.Log;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.BaseAdapter;  
**import** android.widget.TextView;  
  
**import** java.util.ArrayList;  
  
*/\*\*  
 \* Created by Elizabeth on 2016-05-18.  
 \*/***public class** SubmitAdapter **extends** BaseAdapter {  
 **private** LayoutInflater **inflater**;  
 **private** ArrayList<SubmitItem> **data**;  
 **private int layout**;  
 Context **mcontext** = **null**;  
  
 *//Adapter에 추가된 데이터를 저장하기 위한 ArrayList* **private** ArrayList<SubmitItem> **submitItemList** = **new** ArrayList<SubmitItem>();  
  
 *//SubmitAdapter의 생성자* **public** SubmitAdapter() {  
 }  
  
  
 @Override  
 **public int** getCount() {  
 **return submitItemList**.size();  
 }  
  
  
 *//지정한 위치(position)에 있는 데이터와 아이템(row)의 ID를 리턴 : 필수구현* @Override  
 **public long** getItemId(**int** position) {  
 **return** position;  
 }  
  
 *//지정한 위치(position)에 있는 데이터 리턴 : 필수구현* @Override  
 **public** Object getItem(**int** position) {  
 **return submitItemList**.get(position);  
 }  
  
 *//아이템 데이터 추가를 위한 함수. 개발자가 원하는대로 작성 가능.* **public void** addItem(String vTitle, String vDate, String vHospital, String vDisease, String vPresnum) {  
 Log.*d*(**"PHP"**, **"addItem"**);  
 SubmitItem item = **new** SubmitItem(vTitle, vDate, vHospital, vDisease, vPresnum);  
  
 item.setTitle(vTitle);  
 item.setDate(vDate);  
 item.setHospital(vHospital);  
 item.setDisease(vDisease);  
 item.setPrescription(vPresnum);  
  
 **submitItemList**.add(item);  
 notifyDataSetChanged();  
 }  
  
 **public void** init() {  
 **submitItemList**.clear();  
 }  
  
 **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 **final int** pos = position;  
 **mcontext** = parent.getContext();  
  
 *//listview\_item 레이아웃을 inflate하여 convertView 참조 획득* **if** (convertView == **null**) {  
 LayoutInflater inflater = (LayoutInflater) **mcontext**.getSystemService(**mcontext**.***LAYOUT\_INFLATER\_SERVICE***);  
 convertView = inflater.inflate(R.layout.***submit\_item***, parent, **false**);  
  
 convertView.setTag(position);  
 }  
  
 *//화면에 표시될 view(layout이 inflate된)으로부터 위젯에 대한 참조 획득* TextView title = (TextView) convertView.findViewById(R.id.***tv\_title***);  
 TextView date = (TextView) convertView.findViewById(R.id.***tv\_date***);  
 TextView hospital = (TextView) convertView.findViewById(R.id.***tv\_hospital***);  
 TextView disease = (TextView) convertView.findViewById(R.id.***tv\_disease***);  
 TextView presnum = (TextView) convertView.findViewById(R.id.***tv\_presnum***);  
  
 *//Data Set(ListViewItem)에서 position에 위치한 데이터 참조 획득* SubmitItem listviewItem = **submitItemList**.get(position);  
  
  
 *//아이템 내 각 위젯이 데이터 반영* title.setText(listviewItem.getTitle());  
 date.setText(listviewItem.getDate());  
 hospital.setText(listviewItem.getHospital());  
 disease.setText(listviewItem.getDisease());  
 presnum.setText(listviewItem.getPrescription());  
  
  
 **return** convertView;  
 }  
  
  
}

**package** com.medi.medipass;  
  
*/\*\*  
 \* Created by Elizabeth on 2016-05-18.  
 \*/***public class** SubmitItem {  
 **protected** String **vTitle**;*//vName//tv\_title 진료일* **protected** String **vDate**;*//vSurname//tv\_date 처방전 발급 마감일* **protected** String **vHospital**;*//vEmail//tv\_hospital 병원이름* **protected** String **vDisease**;*//vTitle//tv\_disease 병이름* **protected** String **vPrescription**;*//vPres//tv\_presnum 처방전번호* **public** String getTitle() {  
 **return vTitle**;  
 }  
  
 **public void** setTitle(String vTitle) {  
 **this**.**vTitle** = vTitle;  
 }  
  
 **public** String getDate() {  
 **return vDate**;  
 }  
  
 **public void** setDate(String vDate) {  
 **this**.**vDate** = vDate;  
 }  
  
 **public** String getHospital() {  
 **return vHospital**;  
 }  
  
 **public void** setHospital(String vHospital) {  
 **this**.**vHospital** = vHospital;  
 }  
  
 **public** String getDisease() {  
 **return vDisease**;  
 }  
  
 **public void** setDisease(String vDisease) {  
 **this**.**vDisease** = vDisease;  
 }  
  
 **public** String getPrescription() {  
 **return vPrescription**;  
 }  
  
 **public void** setPrescription(String vPrescription) {  
 **this**.**vPrescription** = vPrescription;  
 }  
  
 **public** SubmitItem(String vTitle, String vDate, String vHospital, String vDisease, String vPrescription) {  
 **this**.**vTitle** = vTitle;  
 **this**.**vDate** = vDate;  
 **this**.**vHospital** = vHospital;  
 **this**.**vDisease** = vDisease;  
 **this**.**vPrescription** = vPrescription;  
 }  
  
}

**package** com.medi.medipass;  
  
**import** android.content.Context;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.os.AsyncTask;  
**import** android.os.Build;  
**import** android.os.Bundle;  
**import** android.os.Handler;  
**import** android.provider.Settings;  
**import** android.support.v4.widget.SwipeRefreshLayout;  
**import** android.support.v7.app.AlertDialog;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.AdapterView;  
**import** android.widget.ListView;  
**import** android.widget.Toast;  
  
**import** com.hojung.nfc.HojungNFCReadLibrary;  
**import** com.hojung.nfc.interfaces.OnHojungNFCListener;  
**import** com.hojung.nfc.model.NfcModel;  
  
**import** org.json.JSONArray;  
**import** org.json.JSONException;  
**import** org.json.JSONObject;  
  
**import** java.io.BufferedReader;  
**import** java.io.BufferedWriter;  
**import** java.io.InputStreamReader;  
**import** java.io.OutputStream;  
**import** java.io.OutputStreamWriter;  
**import** java.net.HttpURLConnection;  
**import** java.net.URL;  
  
*//https://www.binpress.com/tutorial/android-l-recyclerview-and-cardview-tutorial/156  
  
/\*\*  
 \* Created by Elizabeth on 2016-05-18.  
 \*/  
  
//https://www.simplifiedcoding.net/android-recyclerview-and-cardview-tutorial/***public class** SubmitPrescription **extends** AppCompatActivity **implements** SwipeRefreshLayout.OnRefreshListener {  
  
 *//http://ggari.tistory.com/528* SwipeRefreshLayout **mSwipeRefreshLayout**;*//새로고침* ListView **listView**;  
  
  
 String **url\_showPrescription** = **"http://condi.swu.ac.kr/Prof-Kang/2013111539/medipass/show\_prescription.php"**;  
 String **url\_submitPrescription** = **"http://condi.swu.ac.kr/Prof-Kang/2013111539/medipass/submit\_prescription.php"**;  
 String **url\_registerPharm** = **"http://condi.swu.ac.kr/Prof-Kang/2013111539/medipass/register\_wait\_list\_pharm.php"**;  
  
 *//Adapter생성* **final** SubmitAdapter **adapter** = **new** SubmitAdapter();  
  
  
 */\*태그 이름 지정\*/* **final** String **TAG** = **"NFCP"**;  
 **final** String **PHP** = **"PHPP"**;  
  
 */\*NFC라이브러리 사용위해 선언\*/* HojungNFCReadLibrary **hojungNFCReadLibrary**;  
 Context **mContext**;  
 Boolean **submitClicked**;  
  
 */\*처방전 제출위함\*/* **static** String *presnum*, *pharm\_code*;  
  
  
 **private** BackPressCloseHandler **backPressCloseHandler**;*// 뒤로가기 버튼 등록(두번 터치시 종료에 사용)* @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***submit\_list***);  
  
 **mSwipeRefreshLayout** = (SwipeRefreshLayout) findViewById(R.id.***swipeRefresh***);*//새로고침* **mSwipeRefreshLayout**.setOnRefreshListener(**this**);  
 */\*리스트\*/  
//SubmitItem(String vTitle,String vDate, String vHospital, String vDisease)* **listView** = (ListView) findViewById(R.id.***submitlist***);  
  
 *//리스트뷰 참조 및 Adapter달기* **listView**.setAdapter(**adapter**);  
  
 *//php를 읽어올때 사용할 변수* GettingPHP gPHP = **new** GettingPHP();  
 gPHP.execute(**url\_showPrescription**);  
**mContext** = **this**;  
  
 */\*다이얼로그(nfc 안내 팝업 띄우기)\*/* AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);  
  
 builder.setTitle(**"약국입니다."**)  
 .setMessage(**"NFC스티커에 태그해주세요."**) *// 메세지 설정* .setCancelable(**false**) *// 뒤로 버튼 클릭시 취소 가능 설정* .setNegativeButton(**"취소"**, **new** DialogInterface.OnClickListener() {  
 *// 취소 버튼 클릭시 설정* **public void** onClick(DialogInterface dialog, **int** whichButton) {  
 dialog.cancel();  
 }  
 });  
  
 **final** AlertDialog dialog = builder.create(); *//다이얼로그 생성* **backPressCloseHandler** = **new** BackPressCloseHandler(**this**);*// 뒤로가기 버튼 객체 생성* **listView**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(AdapterView parent, View v, **int** position, **long** id) {  
 **submitClicked** = **true**; *//다이얼로그 떠있는 상태일 때에만 태그 동작하기 위한 bool값* SubmitItem item = (SubmitItem) parent.getItemAtPosition(position);  
 *presnum* = item.getPrescription();  
 Log.*d*(**"PRES"**, **"presnum : "** + *presnum*);  
 dialog.show();  
 }  
 });  
  
 */\*nfc 사용 안내\*/* android.nfc.NfcAdapter mNfcAdapter1 = android.nfc.NfcAdapter.*getDefaultAdapter*(**mContext**);  
  
 */\*NFC 미지원단말\*/* **if** (mNfcAdapter1 == **null**) {  
 Toast.*makeText*(getApplicationContext(), **"NFC를 지원하지 않는 단말기입니다."**, Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
  
 **try** {  
 */\* NFC꺼져있는 경우, NFC켜기 \*/* **if** (!mNfcAdapter1.isEnabled()) {  
  
 AlertDialog.Builder alertbox = **new** AlertDialog.Builder(**mContext**);  
 alertbox.setTitle(**"Info"**);  
 alertbox.setMessage(**"본 서비스를 이용하기 위해 NFC를 사용하셔야 합니다."**);  
 alertbox.setPositiveButton(**"Turn On"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 **if** (Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***JELLY\_BEAN***) {  
 Intent intent = **new** Intent(Settings.***ACTION\_NFC\_SETTINGS***);  
 startActivity(intent);  
 } **else** {  
 Intent intent = **new** Intent(Settings.***ACTION\_WIRELESS\_SETTINGS***);  
 startActivity(intent);  
 }  
 }  
 });  
 alertbox.setNegativeButton(**"Close"**, **new** DialogInterface.OnClickListener() {  
  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
  
 }  
 });  
 alertbox.show();  
  
 }  
 } **catch** (Exception e) {  
  
 }  
  
 **hojungNFCReadLibrary** = **new** HojungNFCReadLibrary(getIntent(), SubmitPrescription.**this**, **new** OnHojungNFCListener() {  
  
 @Override  
 **public void** onReceiveMessage(NfcModel[] models) {  
 *//* ***TODO Auto-generated method stub* try** {  
  
 Log.*d*(**"NFC1"**, **"type : "** + models[0].getTypeStr() + **" , "** + **"payload : "** + models[0].getPayloadStr() + **" , "** + **"submitClicked : "** + **submitClicked**);  
 String spot = models[0].getTypeStr();  
  
 **if** (spot **instanceof** String) {  
  
 Log.*d*(**"NFC1"**, **"spot : "** + spot);  
 }  
  
 */\*접수하기 다이얼로그 떠있는 상태일 때\*/* **if** (**submitClicked** == **true** && spot.equals(**"pharmacy"**)) {  
  
  
 */\* NFC태그값 토스트로 띄워주고 다이얼로그 끔\*/  
 //Toast.makeText(Home.this, "type : " + models[0].getTypeStr() + " , " + "payload : " + models[0].getPayloadStr(), Toast.LENGTH\_SHORT).show();* Toast.*makeText*(SubmitPrescription.**this**, **"제출되었습니다."**, Toast.***LENGTH\_SHORT***).show();  
  
 **submitClicked** = **false**;  
  
 submitPrescription();  
  
 *//adapter.notifyDataSetChanged();  
 //listView.invalidate();  
  
 /\*다이얼로그 종료\*/* dialog.dismiss();  
  
 */\*태그 정보\*/* Log.*d*(**"NFC"**, **"type"** + models[0].getTypeStr() + **"payload : "** + models[0].getPayloadStr());  
 *pharm\_code* = models[0].getPayloadStr();  
 Log.*d*(**"NFC"**, **"pharm\_code:"** + *pharm\_code*);  
 } **else if** (**submitClicked** == **false** && spot.equals(**"pharmacy"**)) {  
 Toast.*makeText*(SubmitPrescription.**this**, **"처방전을 클릭해주세요."**, Toast.***LENGTH\_SHORT***).show();  
 } **else** {  
 Toast.*makeText*(SubmitPrescription.**this**, **"처방전 제출 버튼입니다. 접수하기 버튼을 눌러주세요."**, Toast.***LENGTH\_SHORT***).show();  
 }  
  
  
 } **catch** (Exception e) {  
  
 }  
  
 }  
  
 @Override  
 **public void** onError(String arg0) {  
 *//* ***TODO Auto-generated method stub*** }  
 });  
  
 }  
  
  
 *//새로고침* @Override  
 **public void** onRefresh() {  
 **mSwipeRefreshLayout**.setRefreshing(**true**);  
 **new** Handler().postDelayed(**new** Runnable() {  
 @Override  
 **public void** run() {  
 **adapter**.init();  
 **adapter**.notifyDataSetChanged();  
 *//해당 어댑터를 서버와 통신한 값이 나오면 됨* GettingPHP gPHP = **new** GettingPHP();  
 gPHP.execute(**url\_showPrescription**);  
 **listView**.setAdapter(**adapter**);  
 **mSwipeRefreshLayout**.setRefreshing(**false**);  
 }  
 }, 1000);  
 }  
  
 **private void** initNFC() {  
 **try** {  
 Log.*d*(**"NFC"**, **"intent : "** + getIntent().getAction());  
 Intent intent = getIntent();  
 **hojungNFCReadLibrary**.onResume(intent);  
 } **catch** (Exception e) {  
  
 }  
  
 }  
  
 **public void** onResume() {  
 **super**.onResume();  
 Log.*d*(**TAG**, **"onResume"**);  
 initNFC();  
*// Intent intent=getIntent();  
// hojungNFCReadLibrary.onResume(intent);* }@Override  
 **protected void** onPause() {  
 **super**.onPause();  
 Log.*d*(**TAG**, **"onPause"**);  
 **hojungNFCReadLibrary**.onPause();  
 }  
  
  
 @Override  
 **public void** onNewIntent(Intent intent) {  
 Log.*d*(**TAG**, **"onNewIntent"**);  
 **hojungNFCReadLibrary**.onNewIntent(intent);  
 }  
  
  
 */\*처방전 제출하기\*/* **public void** submitPrescription() {  
 GettingPHP gPHP = **new** GettingPHP();  
 gPHP.execute(**url\_submitPrescription**);  
 Log.*d*(**"Pharmcode"**,**"Pharmcode : "**+*pharm\_code*);  
 registePharm();  
 }  
  
 **public void** registePharm() {  
 GettingPHP gPHP = **new** GettingPHP();  
 gPHP.execute(**url\_registerPharm**);  
 }  
  
  
 *//AsyncTask : thread + handler  
 //Async(비동기화) : 병렬회로. 계속 요청을 보내는 통로와 응답을 받는 통로를 따로 만들어두는 것  
 //sync(동기화) : 직렬회로. 일이 순차적으로 진행되면서 하나가 해결되면 그다음 일이 진행되는 식으로 네트워크에서는 요청(request)를 보내면 항상 응답(response)을 받아야 진행하는 방식으로 구현* **class** GettingPHP **extends** AsyncTask<String, Integer, String> { *//<Param, Progress, Result(doInBackground의 반환값, onPostExcute의 매개변수)>* @Override  
 **protected void** onPreExecute() {  
 **super**.onPreExecute();  
 }  
  
 *//php에서 데이터를 읽어오는 역할, 백그라운드 스레드로 동작해야 하는 작업을 실행한다 : 필수구현  
 //execute메서드로 전달한 data tye이 params 인수로 전달되는데 여러개의 인수를 전달할 수 있으므로 배열 타입으로 되어 있다.  
 //그래서 하나의 인수만 필요하다면 params[0]만 사용하면 된다.* @Override  
 **protected** String doInBackground(String... params) {  
 StringBuilder jsonHtml = **new** StringBuilder();  
 **try** {  
 *// URL --> openConnection() --> URLConnection --> getInputStream --> InputStream (내용읽음)* URL phpUrl = **new** URL(params[0]);  
 HttpURLConnection conn = (HttpURLConnection) phpUrl.openConnection(); *//URL내용을 읽어오거나 GET/POST로 전달할 때 사용* **if** (conn != **null**) {  
 *//if (params[0].equals(url\_submitPrescription)) {* Log.*d*(**"NFC"**, **"pharm\_code11:"** + *pharm\_code*);  
 String data = **"presnum="** + *presnum* + **"& pharm\_code="** + *pharm\_code*;  
 Log.*d*(**PHP**, data);  
 conn.setReadTimeout(10000);  
 conn.setConnectTimeout(5000);  
 conn.setRequestMethod(**"POST"**);  
 conn.setDoInput(**true**);  
 conn.setDoOutput(**true**);  
 *//conn.setRequestProperty("Content-Type", "application/json");* conn.setUseCaches(**false**);  
  
 OutputStream os = conn.getOutputStream();  
 BufferedWriter bw = **new** BufferedWriter(**new** OutputStreamWriter(os, **"UTF-8"**));  
 bw.write(data);  
 bw.flush();  
 bw.close();  
  
 *//post메세지가 전송된다* conn.connect();  
**if** (conn.getResponseCode() == HttpURLConnection.***HTTP\_OK***) {  
 BufferedReader br = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream()));  
 **while** (**true**) {  
 String line = br.readLine();  
 **if** (line == **null**) **break**;  
 jsonHtml.append(line + **"\n"**);  
 Log.*d*(**"HHH"**, **"list\_line : "** + line);  
 }  
 br.close();  
 }  
 }  
 conn.disconnect();  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 **return** jsonHtml.toString();  
 }  
  
 *//가져온 데이터를 이용해 원하는 일을 하도록 한다* @Override  
 **protected void** onPostExecute(String str) {  
 **try** {  
 *//php에서 받아온 JSON데이터를 JSON오브젝트로 변환* JSONObject jobject = **new** JSONObject(str);  
 *//results라는 key는 JSON배열로 되어있다* JSONArray results = jobject.getJSONArray(**"results"**);  
  
 **if** (jobject.get(**"status"**).equals(**"listp"**)) {  
 Log.*d*(**"LIST"**, **"list\_pres"**);  
 **for** (**int** i = 0; i < results.length(); i++) { *//length->child의 갯수* JSONObject temp = results.getJSONObject(i);  
 String date = temp.getString(**"record\_date"**);  
 String valid\_date = temp.getString(**"valid\_date"**);  
 String hos\_name = temp.getString(**"hospital\_name"**);  
 String pres\_num = temp.getString(**"prescription\_num"**);  
 String dis\_name = temp.getString(**"disease\_name"**);  
 **adapter**.addItem(date, valid\_date, hos\_name, dis\_name, pres\_num);  
  
 }  
 }  
  
 **adapter**.notifyDataSetChanged();  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
}

**package** com.medi.medipass;  
  
**import** android.content.Context;  
**import** android.os.AsyncTask;  
**import** android.os.Bundle;  
**import** android.os.Handler;  
**import** android.support.v4.widget.SwipeRefreshLayout;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.util.Log;  
**import** android.widget.ListView;  
  
**import** org.json.JSONArray;  
**import** org.json.JSONException;  
**import** org.json.JSONObject;  
  
**import** java.io.BufferedReader;  
**import** java.io.InputStreamReader;  
**import** java.net.HttpURLConnection;  
**import** java.net.URL;  
  
*/\*\*  
 \* Created by Elizabeth on 2016-04-07.  
 \*/***public class** WaitList **extends** AppCompatActivity **implements** SwipeRefreshLayout.OnRefreshListener {  
 */\* 로그인 안하고 대기인원 확인하기 \*/* SwipeRefreshLayout **mSwipeRefreshLayout**;*//새로고침* ListView **listView**;  
  
 *//data를 받아올 php주소* String **url\_hosinfo** = **"http://condi.swu.ac.kr/Prof-Kang/2013111539/medipass/all\_hosinfo.php"**;  
  
 *//Adapter생성* **final** WaitListAdapter **adapter** = **new** WaitListAdapter();  
  
  
 Context **mContext**;  
  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***wait\_list***);  
  
 **mSwipeRefreshLayout** = (SwipeRefreshLayout) findViewById(R.id.***swipeRefresh\_wait***);*//새로고침* **mSwipeRefreshLayout**.setOnRefreshListener(**this**);  
  
  
 **listView** = (ListView) findViewById(R.id.***waitlist***);  
 *//리스트뷰 참조 및 Adapter달기* **listView**.setAdapter(**adapter**);  
  
 *//php를 읽어올때 사용할 변수* GettingPHP gPHP = **new** GettingPHP();  
 gPHP.execute(**url\_hosinfo**);  
 **mContext** = **this**;  
 }  
  
  
 *//새로고침* @Override  
 **public void** onRefresh() {  
 **mSwipeRefreshLayout**.setRefreshing(**true**);  
 **new** Handler().postDelayed(**new** Runnable() {  
 @Override  
 **public void** run() {  
 **adapter**.init();  
 **adapter**.notifyDataSetChanged();  
 *//해당 어댑터를 서버와 통신한 값이 나오면 됨* GettingPHP gPHP = **new** GettingPHP();  
 gPHP.execute(**url\_hosinfo**);  
 **listView**.setAdapter(**adapter**);  
 **mSwipeRefreshLayout**.setRefreshing(**false**);  
 }  
 }, 1000);  
 }  
  
 *//AsyncTask : thread + handler  
 //Async(비동기화) : 병렬회로. 계속 요청을 보내는 통로와 응답을 받는 통로를 따로 만들어두는 것  
 //sync(동기화) : 직렬회로. 일이 순차적으로 진행되면서 하나가 해결되면 그다음 일이 진행되는 식으로 네트워크에서는 요청(request)를 보내면 항상 응답(response)을 받아야 진행하는 방식으로 구현* **class** GettingPHP **extends** AsyncTask<String, Integer, String> { *//<Param, Progress, Result(doInBackground의 반환값, onPostExcute의 매개변수)>* @Override  
 **protected void** onPreExecute() {  
 **super**.onPreExecute();  
 }  
  
 *//php에서 데이터를 읽어오는 역할, 백그라운드 스레드로 동작해야 하는 작업을 실행한다 : 필수구현  
 //execute메서드로 전달한 data tye이 params 인수로 전달되는데 여러개의 인수를 전달할 수 있으므로 배열 타입으로 되어 있다.  
 //그래서 하나의 인수만 필요하다면 params[0]만 사용하면 된다.* @Override  
 **protected** String doInBackground(String... params) {  
 StringBuilder jsonHtml = **new** StringBuilder();  
 **try** {  
 *// URL --> openConnection() --> URLConnection --> getInputStream --> InputStream (내용읽음)* URL phpUrl = **new** URL(params[0]);  
 HttpURLConnection conn = (HttpURLConnection) phpUrl.openConnection(); *//URL내용을 읽어오거나 GET/POST로 전달할 때 사용* **if** (conn != **null**) {  
  
 **if** (conn.getResponseCode() == HttpURLConnection.***HTTP\_OK***) {  
 BufferedReader br = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream()));  
 **while** (**true**) {  
 String line = br.readLine();  
 **if** (line == **null**) **break**;  
 jsonHtml.append(line + **"\n"**);  
 Log.*d*(**"HHH"**, **"list\_line : "** + line);  
 }  
 br.close();  
 }  
 }  
 conn.disconnect();  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 **return** jsonHtml.toString();  
 }  
  
 *//가져온 데이터를 이용해 원하는 일을 하도록 한다* @Override  
 **protected void** onPostExecute(String str) {  
 **try** {  
 *//php에서 받아온 JSON데이터를 JSON오브젝트로 변환* JSONObject jobject = **new** JSONObject(str);  
 *//results라는 key는 JSON배열로 되어있다* JSONArray results = jobject.getJSONArray(**"results"**);  
 Log.*d*(**"LIST"**, **"listw"**);  
 **if** (jobject.get(**"status"**).equals(**"listw"**)) {  
 Log.*d*(**"LIST"**, **"listw"**);  
 **for** (**int** i = 0; i < results.length(); i++) { *//length->child의 갯수* JSONObject temp = results.getJSONObject(i);  
 String hospital\_name = temp.getString(**"hospital\_name"**);  
 String wait\_num = temp.getString(**"wait\_num"**);  
 **adapter**.addItem(hospital\_name, wait\_num);  
 }  
 }  
  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
}

**package** com.medi.medipass;  
  
**import** android.content.Context;  
**import** android.util.Log;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.BaseAdapter;  
**import** android.widget.TextView;  
  
**import** java.util.ArrayList;  
  
*/\*\*  
 \* Created by Elizabeth on 2016-05-31.  
 \*/***public class** WaitListAdapter **extends** BaseAdapter {  
 Context **mcontext** = **null**;  
  
 *//Adapter에 추가된 데이터를 저장하기 위한 ArrayList* **private** ArrayList<WaitListItem> waitItemList = **new** ArrayList<WaitListItem>();  
  
 *//WaitListAdapter 생성자* **public** WaitListAdapter() {  
 }  
  
 *//Adapter에 사용되는 데이터의 개수를 리턴 : 필수구현* @Override  
 **public int** getCount() {  
 **return** waitItemList.size();  
 }  
  
 *//지정한 위치(position)에 있는 데이터와 아이템(row)의 ID를 리턴 : 필수구현* @Override  
 **public long** getItemId(**int** position) {  
 **return** position;  
 }  
  
 *//지정한 위치(position)에 있는 데이터 리턴 : 필수구현* @Override  
 **public** Object getItem(**int** position) {  
 **return** waitItemList.get(position);  
 }  
  
  
 *//아이템 데이터 추가를 위한 함수. 개발자가 원하는대로 작성 가능.* **public void** addItem(String hosname, String waitnum) {  
 Log.*d*(**"PHP"**, **"addItem"**);  
 WaitListItem item = **new** WaitListItem(hosname, waitnum);  
  
 item.setHosname(hosname);  
 item.setHoswait(waitnum);  
  
 waitItemList.add(item);  
 notifyDataSetChanged();  
 }  
  
 **public void** init() {  
 waitItemList.clear();  
 }  
  
 *//position에 위치한 데이터를 화면에 출력하는데 사용될 view를 리턴 : 필수구현* @Override  
 **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 **final int** pos = position;  
 **mcontext** = parent.getContext();  
  
 *//listview\_item 레이아웃을 inflate하여 convertView 참조 획득* **if** (convertView == **null**) {  
 LayoutInflater inflater = (LayoutInflater) **mcontext**.getSystemService(**mcontext**.***LAYOUT\_INFLATER\_SERVICE***);  
 convertView = inflater.inflate(R.layout.***wait\_item***, parent, **false**);  
  
 convertView.setTag(position);  
 }  
  
 *//화면에 표시될 view(layout이 inflate된)으로부터 위젯에 대한 참조 획득  
 //waithos, waitnum* TextView item\_hosname = (TextView) convertView.findViewById(R.id.***waithos***);  
 TextView item\_waitnum = (TextView) convertView.findViewById(R.id.***waitnum***);  
  
 *//Data Set(ListViewItem)에서 position에 위치한 데이터 참조 획득* WaitListItem listviewItem = waitItemList.get(position);  
  
 *//아이템 내 각 위젯이 데이터 반영* item\_hosname.setText(listviewItem.getHosname());  
 item\_waitnum.setText(listviewItem.getHoswait());  
  
 **return** convertView;  
 }  
  
  
}

**package** com.medi.medipass;  
  
*/\*\*  
 \* Created by Elizabeth on 2016-05-31.  
 \*/***public class** WaitListItem {  
 *//waithos, waitnum* **private** String **hosname**;  
 **private** String **hoswait**;  
  
 **public** String getHosname() {  
 **return hosname**;  
 }  
  
 **public void** setHosname(String hosname) {  
 **this**.**hosname** = hosname;  
 }  
  
 **public** String getHoswait() {  
 **return hoswait**;  
 }  
  
 **public void** setHoswait(String hoswait) {  
 **this**.**hoswait** = hoswait;  
 }  
  
 **public** WaitListItem(String hosname, String hoswait) {  
 **this**.**hosname** = hosname;  
 **this**.**hoswait** = hoswait;  
 }  
  
}