Creating vector object (image.png)

Right click on app and choose vector, then choose image, change name and click next. It will push it to the drawings

private final String CHANNEL\_ID = "personal\_notifications"; - create first constant   
private final int NOTIFICATION\_ID = 001; - create second constant

createNotification(); - creating channel (it is needed for android > 8.0)  
  
 Intent landing = new Intent(this, Main2Activity.class); - creating intent for new activity after click  
 landing.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK); - set flags   
  
 PendingIntent pending = PendingIntent.getActivity(this, 0, landing, PendingIntent.FLAG\_ONE\_SHOT); - creating new activity on clicking notification   
  
 NotificationCompat.Builder builder = new NotificationCompat.Builder(this, CHANNEL\_ID); - create main bilder  
 builder.setSmallIcon(R.drawable.ic\_wc); - set img  
 builder.setContentTitle("notification tile");   
 builder.setContentText("main text");  
 builder.setPriority(NotificationCompat.PRIORITY\_DEFAULT);  
 builder.setAutoCancel(true); - close after click

builder.setContentIntent(pending); - set activity on clicking notification  
 builder.addAction(R.drawable…., "…", pending); - add button with activity and image, image doesn't work on emulator  
  
 NotificationManagerCompat compat = NotificationManagerCompat.from(this);  
 compat.notify(NOTIFICATION\_ID, builder.build()); - run notification ( has to be applied channel if version is > 8.0)  
  
private void createNotification(){  
 if(Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O){ - if android version > 8.0  
 CharSequence name = "name";  
 String description = "description";  
 int importance = NotificationManager.IMPORTANCE\_DEFAULT;  
  
 NotificationChannel channel = new NotificationChannel(CHANNEL\_ID,name,importance);  
 channel.setDescription(description);  
 NotificationManager manager = (NotificationManager)getSystemService(NOTIFICATION\_SERVICE);  
 manager.createNotificationChannel(channel);  
 }  
}

Close from another activity

NotificationManager manager = (NotificationManager) getSystemService(NOTIFICATION\_SERVICE);  
manager.cancel(MainActivity.NOTIFICATION\_ID); - close notification from another activity

Notification with writable answer

public static String TXT\_REPLY = "…";

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.N) – check version  
{  
 RemoteInput remoteInput = new RemoteInput.Builder(TXT\_REPLY).setLabel("placeholder ").build();  
  
 Intent intent = new Intent(this, Main2Activity.class); - new activity   
 intent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);  
 PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent, PendingIntent.FLAG\_ONE\_SHOT);  
  
 NotificationCompat.Action action = new NotificationCompat.Action.Builder(R.drawable…., "title of text", pendingIntent).addRemoteInput(remoteInput).build();  
 builder.addAction(action);  
  
}

then get this message in needed activity by constant:

Bundle bundle = RemoteInput.getResultsFromIntent(getIntent());  
if (bundle != null){  
 bundle.getCharSequence(MainActivity.TXT\_REPLY).toString();  
}

Create download line

class MyThread extends Thread{ - must be in another thread because it will stop your application with while loop

define and set variables:  
 NotificationManagerCompat compat; NotificationCompat.Builder builder;  
 MyThread(NotificationCompat.Builder b, NotificationManagerCompat c){  
 builder = b; compat = c;  
 }

@Override  
 public void run() {  
 super.run();  
 NotificationManagerCompat compat = NotificationManagerCompat.from(context); - context is “this” in main func.  
 int max = 100, current = 0;  
 while (current <= max){  
 builder.setProgress(max, current++,false); - set progress bar  
 compat.notify(NOTIFICATION\_ID, builder.build()); - build bar  
 }  
 builder.setProgress(0,0,false); - delete progress bar   
 compat.notify(NOTIFICATION\_ID, builder.build());

}  
}

Expanding notification with image or text and icon

Bitmap bitmap = BitmapFactory.decodeResource(getResources(), R.drawable.pic9); - get image  
builder.setLargeIcon(bitmap); - setting small icon on the right  
builder.setStyle(new NotificationCompat.BigPictureStyle().bigPicture(bitmap).bigLargeIcon(null)); - set big icon  
builder.setStyle(new NotificationCompat.BigTextStyle().bigText("…")); - set large text

Custom layout

https://www.youtube.com/watch?v=4g97m0mgLXI&list=PLshdtb5UWjSrOJfpFOE-u55s3SnY2EO9v&index=51

to manage TextView maybe try to get it by id and change