

Android 开发 · BMI计算器

22920182204218

李晓旭

一、实现内容

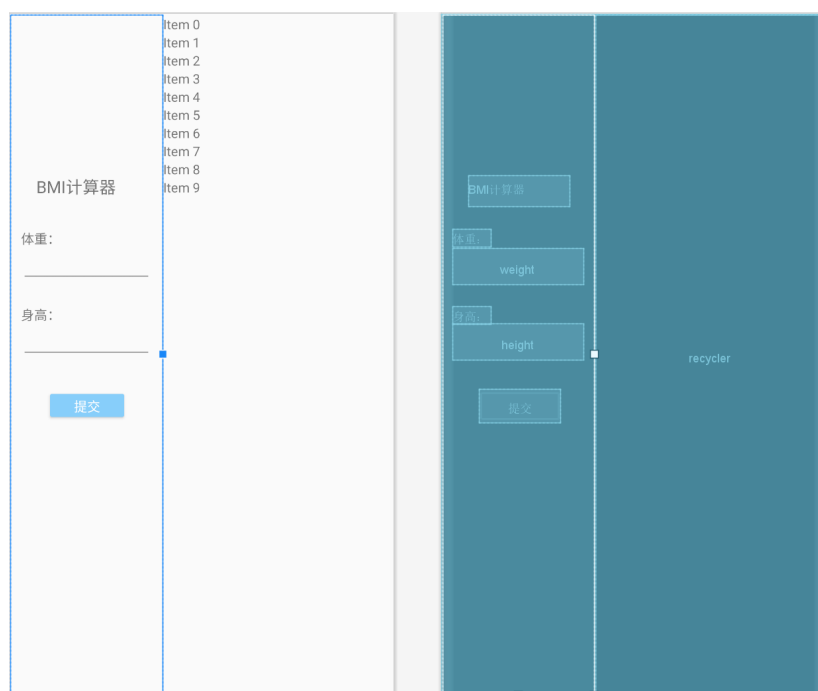
使用RecyclerView控件实现：

- 输入框输入身高体重值
- 计算得到这些值并在RecyclerView中显示
- 连接Web服务器，数据操作在远程服务器进行

二、实现过程

1、创建主活动页：

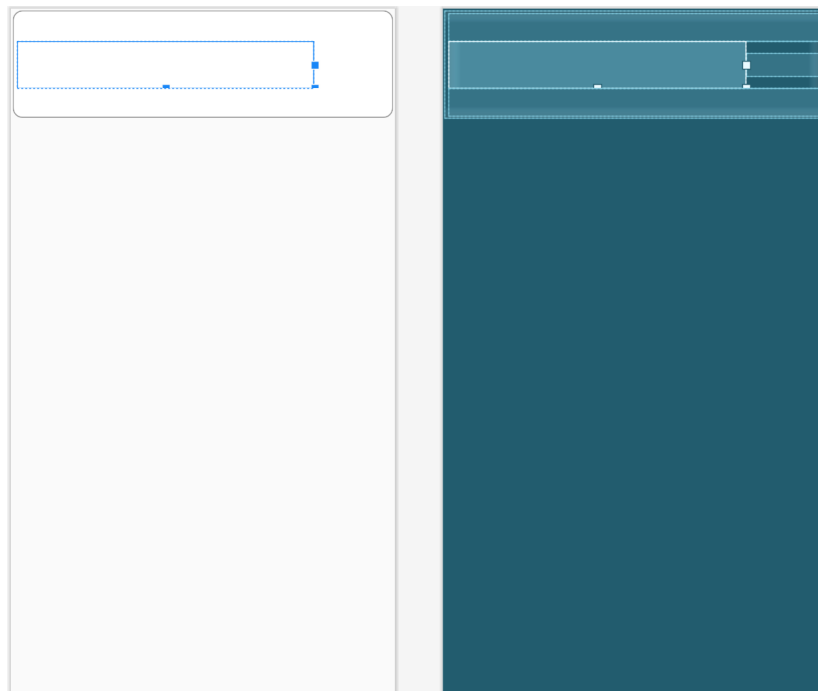
如下图：



首先创建排列方式为 `horizontal` 的 `LinearLayout` 以分开两半，权重分别为2，3。左侧放置控件较多，主要为身高体重的 `EditText` 和提交按钮。右侧放置 `RecyclerView`，用来显示bmi相关信息。

2、创建RecyclerView中各item的显示块

如下图：



该块首先用 `LinearLayout` 的 `vertical` 布局方式布局为三行，各自高度为30dp, 50dp, 24dp。其中第二行再次用 `LinearLayout` 分隔为左右两半，权重分别为4, 1。在此规定第一行显示时间信息，第二行显示bmi信息和身体状态信息，第三行是体重身高信息。

3、美化控件

原本创建了两个文件，但是可能因为主题的问题，按钮控件只能改颜色，圆角无法正常显示。

圆角边框item项：

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <shape xmlns:android="http://schemas.android.com/apk/res/android" >
3      <solid android:color="#FFFFFF" />
4      <corners android:topLeftRadius="10dp"
5              android:topRightRadius="10dp"
6              android:bottomRightRadius="10dp"
7              android:bottomLeftRadius="10dp" />
8      <stroke
9          android:width="1dp"
10         android:color="#999999" />
11     <padding
12         android:bottom="3dp"
13         android:left="5dp"
14         android:right="5dp"
15         android:top="3dp" />
16 </shape>
```

4、创建item类

该类用于存放每个身高体重等项的信息，在构造函数中顺便计算量bmi值，方便后面使用。

```
1  public class item {
2      public float height, weight;
3      public double bmi;
4      public String time;
5      public item(float h, float w, String t){
6          height = h;
```

```

7         weight = w;
8         time = t;
9         bmi = culBMI();
10    }
11    public double culBMI(){
12        double bmiHere = weight * 1.0 / height / height;
13        return bmiHere;
14    }
15 }

```

5、创建适配器类

Adapter主要需要完成下面几个方法：

`onCreateViewHolder()`：每当 `RecyclerView` 需要创建新的 `ViewHolder` 时，它都会调用此方法。此方法会创建并初始化 `ViewHolder` 及其关联的 `View`，但不会填充视图的内容。

`onBindViewHolder()`：调用此方法将 `ViewHolder` 与数据相关联。此方法会提取适当的数据，并使用该数据填充 `ViewHolder` 的布局。

`getItemCount()`：RecyclerView 调用此方法来获取数据集的大小。

由于还要实现动态添加的功能，在这里新增一个函数：

`getItemCount()`：更新Adapter的数据项并显示动画。

代码如下：

(1) 存储各项数据

在Adapter类中使用一个一维对象组存储所需数据：

```

1    public item[] I;

```

(2) ViewHolder类及其构造函数、辅助函数

在此类中，首先public列出四个TextView控件，后面需要使用到这四个控件来显示信息。

构造函数中首先调用父类的无参构造器，然后把public的四个控件对应到item.xml中的控件。下面是辅助函数getTextView，用来返回各个控件以备Adapter的函数调用设置text信息。

```

1    public static class ViewHolder extends RecyclerView.ViewHolder{
2        public TextView tv_time;
3        public TextView tv_others;
4        public TextView tv_bmi;
5        public TextView tv_status;
6        public ViewHolder(View view) {
7            super(view);
8            tv_time = (TextView)view.findViewById(R.id.time);
9            tv_others = (TextView)view.findViewById(R.id.others);
10           tv_bmi = (TextView)view.findViewById(R.id.bmi);
11           tv_status = (TextView)view.findViewById(R.id.status);
12        }
13        public TextView getTextView(int i){
14            switch (i){
15                case 1: return tv_time;
16                case 2: return tv_bmi;
17                case 3: return tv_others;
18                case 4: return tv_status;

```

```

19         default:break;
20     }
21     TextView tv = (TextView) itemView.findViewById(R.id.bmi);
22     tv.setText("error");
23     return tv;
24 }
25 }

```

(3) 适配器的构造函数

把参数值赋给 (1) 中的public变量

```

1 public RecyclerViewAdapter(item[] it){
2     I = it;
3 }

```

(4) onCreateViewHolder()

创建并初始化 ViewHolder 及其关联的 View

```

1 public RecyclerViewAdapter.ViewHolder onCreateViewHolder(@NonNull
  @org.jetbrains.annotations.NotNull ViewGroup parent, int viewType) {
2
3     View view =
4     LayoutInflater.from(parent.getContext()).inflate(R.layout.item,parent,false);
5     return new ViewHolder(view);
6 }

```

(5) onBindViewHolder()

设置四个控件的文字内容。这里加了几个判断，根据不同判断能够获取不同的身体状态和状态字的背景色

```

1 @Override
2 public void onBindViewHolder(@NonNull @org.jetbrains.annotations.NotNull
  RecyclerViewAdapter.ViewHolder holder, int position) {
3     holder.getTextView(1).setText(I[position].time);
4     holder.getTextView(2).setText("BMI:" + String.format("%.2f",
  I[position].bmi));
5     holder.getTextView(3).setText(" 身高:" +
  String.valueOf(I[position].height) + "m 体
  重:" + String.valueOf(I[position].weight) + "kg");
6     double bmiHere = I[position].bmi;
7     String status = new String();
8     if(bmiHere <= 18.4) {
9         status = "偏瘦";
10        holder.getTextView(4).setText(status);
11
12        holder.getTextView(4).setBackgroundColor(Color.parseColor("#48D1CC"));
13    }
14    else if(bmiHere <= 23.9) {
15        status = "适中";
16        holder.getTextView(4).setText(status);
17
18        holder.getTextView(4).setBackgroundColor(Color.parseColor("#90EE90"));
19    }
20    else if(bmiHere <= 27.9) {
21        status = "过重";

```

```

20         holder.getTextView(4).setText(status);
21
22         holder.getTextView(4).setBackgroundColor(Color.parseColor("#FFA500"));
23     }
24     else {
25         status = "肥胖";
26         holder.getTextView(4).setText(status);
27
28         holder.getTextView(4).setBackgroundColor(Color.parseColor("#FF4500"));
29     }

```

(6) getItemCount()

获取item数。由于item顺序存放且初始化的item有40，后面未定义的都是null，返回长度出错，故挨个遍历找出长度

```

1  @Override
2  public int getItemCount() {
3      int i = 0;
4      for( ; i <= I.length; i++)
5      {
6          if(I[i] == null)
7              break;
8      }
9      return i;
10 }

```

(7) newItem()

插入新值并通过动画显示

```

1  public void newItem(item itemNew){
2      int i = getItemCount();
3      for( ; i > 0 ; i--){
4          I[i+1] = I[i];
5      }
6      I[0] = itemNew;
7      notifyItemInserted(0);
8  }

```

6、完成主活动

(1) 创建LayoutManager以设置RecyclerView

```

1  LinearLayoutManager manager = new LinearLayoutManager(this);
2  manager.setOrientation(RecyclerView.VERTICAL);
3  RE = (RecyclerView)findViewById(R.id.recycler);
4  RE.setLayoutManager(manager);

```

(2) 初始化数据

所有的数据都在Web服务器 <https://southstem.cloud> 中。要想访问服务器首先需要在Android Studio中添加网络权限：

```

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

```

由于当今的Android已经不支持简单的Http访问，必须要求https。故需要在nginx配置ssl（ssl可从腾讯云0元获得）。配置好后在服务器端(轻量级服务器Flask + NGINX + SSL)实现两个路由：

- /getBmiData：用于获取存储在服务器的BMI数据的json文件，并以json方式返回
- /addNewBmi?weight=*&height=*：首先将数据读出并解析为字典列表，然后根据GET请求的参数组建立一个新字典并加入字典列表，最后将字典列表转json并写入文件

```
1  @app.route("/getBmiData",methods=['GET', 'POST'])
2  def getBmiData():
3      with open("./static/bmiData.json",'r') as f:
4          res = f.read()
5          config = json.loads(res)
6          print (len(config))
7          return res
8      return "error"
9
10 @app.route("/addNewBmi", methods=['GET'])
11 def addNewBmi():
12     height = request.args.get('height')
13     weight = request.args.get('weight')
14     config = None
15     res = {}
16     timeHere = time.strftime("%Y.%m.%d", time.localtime())
17     with open("./static/bmiData.json",'r') as f:
18         config = json.load(f)
19         newHere = {
20             'height' : height,
21             'weight' : weight,
22             'time'    : timeHere
23         }
24         print(config)
25         config.append(newHere)
26         f.close
27     with open("./static/bmiData.json",'w') as fw:
28         fw.truncate()
29         json.dump(config,fw)
30         fw.close
31     return "success"
32
```

而后编写初始化数据的函数。这里由于网络通信是很费时的活动，需要新建线程。若不新建线程，程序报错。

```
1  public void initItems() {
2      new Thread(new Runnable() {
3          @Override
4          public void run() {
5              item[] itemsHere = new item[maxNum];
6              int k = 0;
7              BufferedReader reader;
8              StringBuilder jsonData = new StringBuilder();
9              HttpURLConnection connection;
10             try {
11                 URL requestUrl = new URL("https://southstem.cloud/getBmiData");
12                 connection = (HttpURLConnection) requestUrl.openConnection();
13                 connection.setRequestMethod("GET");
14                 connection.setConnectTimeout(5000);
```

```

15         connection.setReadTimeout(8000);
16         if (connection.getResponseCode() == 200) {
17             InputStream in = connection.getInputStream();
18             reader = new BufferedReader(new InputStreamReader(in));
19             String line;
20             while ((line = reader.readLine()) != null) {
21                 jsonData.append(line);
22             }
23         }
24         JSONArray jsonArray = new JSONArray(jsonData.toString());
25         for (int i = 0; i < jsonArray.length(); i++) {
26             JSONObject jsonObject = (JSONObject) jsonArray.get(i);
27             item newItemHere = new
item(Float.parseFloat(jsonObject.getString("height")),
Float.parseFloat(jsonObject.getString("weight")),
28                                     jsonObject.getString("time"));
29             itemsHere[k] = newItemHere;
30             k++;
31         }
32         connection.disconnect();
33     } catch (MalformedURLException | ProtocolException e) {
34         e.printStackTrace();
35     } catch (IOException e) {
36         e.printStackTrace();
37     } catch (JSONException e) {
38         e.printStackTrace();
39     }
40     items = itemsHere;
41 }
42 }).start();
43 }

```

(3) 创建适配器并显示

这里的While是由于实现进程间通信较为麻烦，这里的处理简单一些，直接循环判定是否为空即可

```

1     initItems();
2     while (items == null) ;
3     adapter = new RecyclerViewAdapter(items);
4     RE.setAdapter(adapter);

```

(4) 完成提交按钮点击事件

同上文，主要是创建线程和get请求。同时获取各输入信息后需要判断是否为空。

```

1     public void submit(View view) {
2         RE.scrollToPosition(0);
3         int k = 0;
4         new Thread(new Runnable() {
5             @Override
6             public void run() {
7                 EditText weightHereStr = (EditText) findViewById(R.id.weight);
8                 EditText heightHereStr = (EditText) findViewById(R.id.height);
9                 String weightHere = weightHereStr.getText().toString();
10                String heightHere = heightHereStr.getText().toString();
11                System.out.println(weightHere);
12                if(!weightHere.isEmpty() && !heightHere.isEmpty()) {
13                    float weight = Float.parseFloat(weightHere);

```

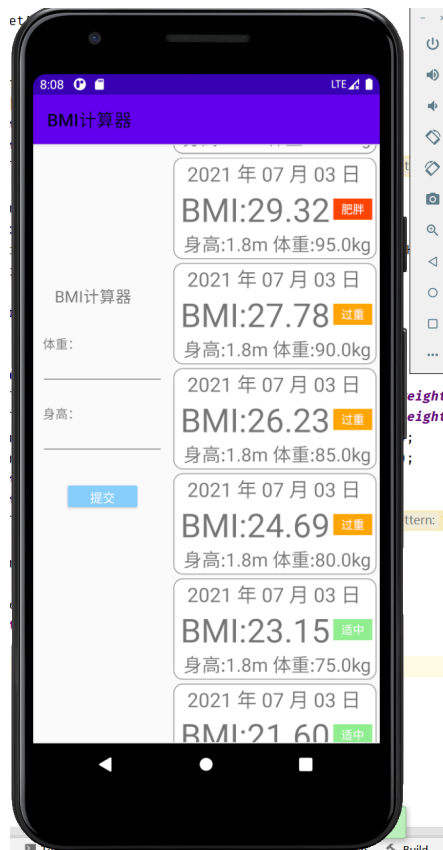
```

14         float height = Float.parseFloat(heightHere);
15         SimpleDateFormat formatter = new SimpleDateFormat("
yyyy.MM.dd");
16         try {
17             String uu = "添加记录: " + weightHere + "kg ,"
+ heightHere + "m 成功。";
18             System.out.println(uu);
19             URL reqUrl = new URL("https://southstem.cloud/addNewBmi?
height="
+ weightHere);
20             HttpURLConnection connection2 = (HttpURLConnection)
reqUrl.openConnection();
21             connection2.setRequestMethod("GET");
22             connection2.setConnectTimeout(5000);
23             connection2.setReadTimeout(8000);
24             if (connection2.getResponseCode() == 200) {
25                 resultStr = uu;
26             } else {
27                 resultStr = "ERROR!";
28             }
29             connection2.disconnect();
30         } catch (MalformedURLException e) {
31             e.printStackTrace();
32         } catch (ProtocolException protocolException) {
33             protocolException.printStackTrace();
34         } catch (IOException ioException) {
35             ioException.printStackTrace();
36         }
37         Date curDate = new Date(System.currentTimeMillis());
38         String timeHere = formatter.format(curDate);
39         item itemNew = new item(height, weight, timeHere);
40         adapter.newItem(itemNew);
41     }
42     else{
43         resultStr = "ERROR";
44     }
45     flag = true;
46 }
47 }).start();
48 while(!flag) ;
49 Toast toast = Toast.makeText(getApplicationContext(),
resultStr, Toast.LENGTH_LONG);
50 toast.show();
51 flag = false;
52 resultStr = "";
53 }

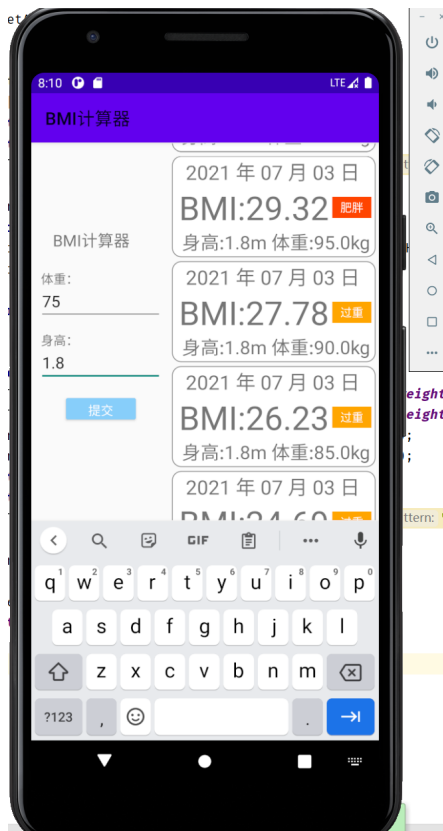
```

代码部分完成

三、运行结果



可见初始化结果显示正常，计算正确且颜色显示正确。下面输入数据测试：



点击提交：



可见最上方出现我们需要的结果，测试完成。

四、总结

本次BMI计算器程序使用RecyclerView控件实现了一个BMI计算器，能够动态的添加数据并能够正确的显示。由于RecyclerView 会回收单个元素。当列表项滚动出屏幕时，RecyclerView 不会销毁其视图而会对屏幕上滚动的新列表项重用该视图，则可以显著提高性能，改善应用响应能力并降低功耗。RecyclerView使用起来较为简便，在现代安卓开发中发挥着重要作用。结合Web服务，RecyclerView可以更简便地开发Web应用。