

# Android 开发 · BMI计算器

22920182204218

李晓旭

## 一、实现内容

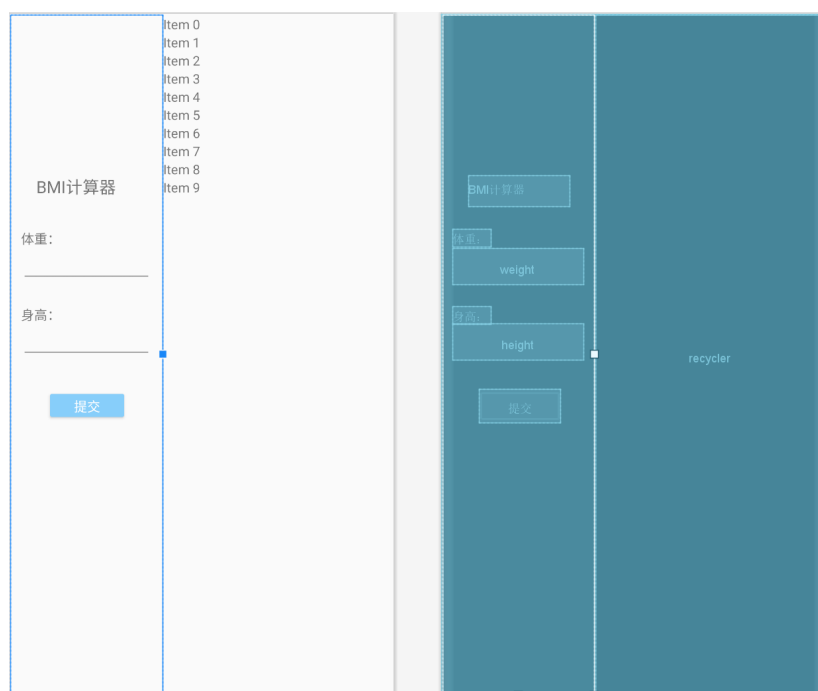
使用RecyclerView控件实现：

- 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：

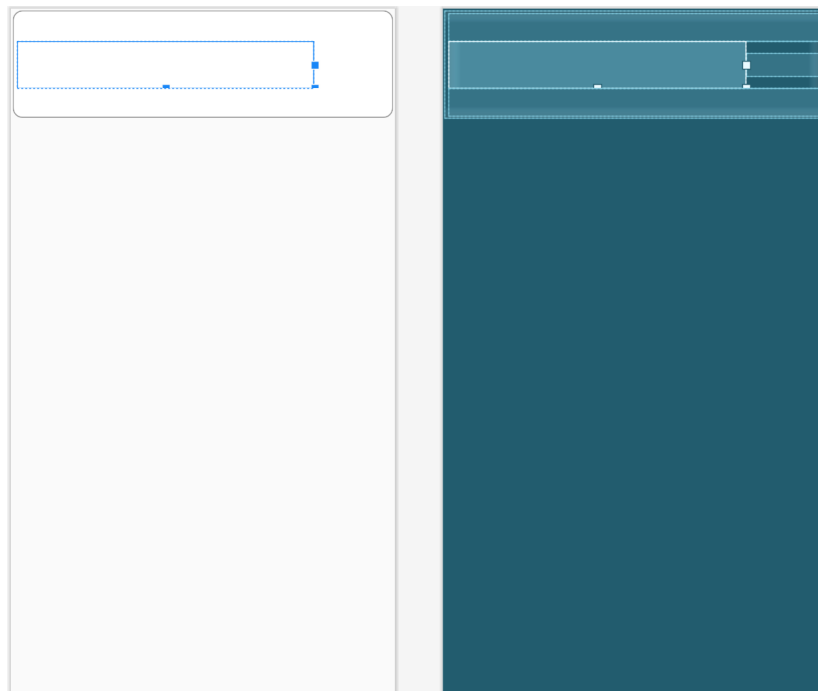
如下图：



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：

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android" >
    <solid android:color="#FFFFFF" />
    <corners android:topLeftRadius="10dp"
        android:topRightRadius="10dp"
        android:bottomRightRadius="10dp"
        android:bottomLeftRadius="10dp" />
    <stroke
        android:width="1dp"
        android:color="#999999" />
    <padding
        android:bottom="3dp"
        android:left="5dp"
        android:right="5dp"
        android:top="3dp" />
</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。

```
public class item {
    public float height, weight;
    public double bmi;
    public String time;
    public item(float h, float w, String t){
        height = h;
```

```

        weight = w;
        time = t;
        bmi = culBMI();
    }
    public double culBMI(){
        double bmiHere = weight * 1.0 / height / height;
        return bmiHere;
    }
}

```

## 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：

```
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。

```

public static class ViewHolder extends RecyclerView.ViewHolder{
    public TextView tv_time;
    public TextView tv_others;
    public TextView tv_bmi;
    public TextView tv_status;
    public ViewHolder(View view) {
        super(view);
        tv_time = (TextView)view.findViewById(R.id.time);
        tv_others = (TextView)view.findViewById(R.id.others);
        tv_bmi = (TextView)view.findViewById(R.id.bmi);
        tv_status = (TextView)view.findViewById(R.id.status);
    }
    public TextView getTextView(int i){
        switch (i){
            case 1: return tv_time;
            case 2: return tv_bmi;
            case 3: return tv_others;
            case 4: return tv_status;
        }
    }
}

```

```

        default:break;
    }
    TextView tv = (TextView) itemView.findViewById(R.id.bmi);
    tv.setText("error");
    return tv;
}
}

```

### (3) 适配器的构造函数

把参数值赋给 (1) 中的public变量

```

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

```

### (4) onCreateViewHolder()

创建并初始化 `ViewHolder` 及其关联的 `View`

```

public RecyclerViewAdapter.ViewHolder onCreateViewHolder(@NonNull
@org.jetbrains.annotations.NotNull ViewGroup parent, int viewType) {

    View view =
    LayoutInflater.from(parent.getContext()).inflate(R.layout.item,parent,false);
    return new ViewHolder(view);
}

```

### (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

```

@Override
public void onBindViewHolder(@NonNull @org.jetbrains.annotations.NotNull
RecyclerViewAdapter.ViewHolder holder, int position) {
    holder.getTextView(1).setText(I[position].time);
    holder.getTextView(2).setText("BMI:" + String.format("%.2f",
I[position].bmi));
    holder.getTextView(3).setText(" 身高:" + String.valueOf(I[position].height)
+ "m 体重:" + String.valueOf(I[position].weight) + "kg");
    double bmiHere = I[position].bmi;
    String status = new String();
    if(bmiHere <= 18.4) {
        status = "偏瘦";
        holder.getTextView(4).setText(status);
        holder.getTextView(4).setBackgroundColor(Color.parseColor("#48D1CC"));
    }
    else if(bmiHere <= 23.9) {
        status = "适中";
        holder.getTextView(4).setText(status);
        holder.getTextView(4).setBackgroundColor(Color.parseColor("#90EE90"));
    }
    else if(bmiHere <= 27.9) {
        status = "过重";
        holder.getTextView(4).setText(status);
        holder.getTextView(4).setBackgroundColor(Color.parseColor("#FFA500"));
    }
}

```

```

        else {
            status = "肥胖";
            holder.getTextView(4).setText(status);
            holder.getTextView(4).setBackgroundColor(Color.parseColor("#FF4500"));
        }
    }
}

```

## (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

```

@Override
public int getItemCount() {
    int i = 0;
    for( ; i <= I.length; i++)
    {
        if(I[i] == null)
            break;
    }
    return i;
}

```

## (7) newItem()

插入新值并通过动画显示

```

public void newItem(item itemNew){
    int i = getItemCount();
    for( ; i > 0 ; i--){
        I[i+1] = I[i];
    }
    I[0] = itemNew;
    notifyItemInserted(0);
}

```

# 6、完成主活动

## (1) 创建LayoutManager以设置RecyclerView

```

LinearLayoutManager manager = new LinearLayoutManager(this);
manager.setOrientation(RecyclerView.VERTICAL);
RE = (RecyclerView)findViewById(R.id.recycler);
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

```
@app.route("/getBmiData", methods=['GET', 'POST'])
def getBmiData():
    with open("./static/bmiData.json", 'r') as f:
        res = f.read()
        config = json.loads(res)
        print(len(config))
        return res
    return "error"

@app.route("/addNewBmi", methods=['GET'])
def addNewBmi():
    height = request.args.get('height')
    weight = request.args.get('weight')
    config = None
    res = {}
    timeHere = time.strftime("%Y.%m.%d", time.localtime())
    with open("./static/bmiData.json", 'r') as f:
        config = json.load(f)
        newHere = {
            'height' : height,
            'weight' : weight,
            'time' : timeHere
        }
        print(config)
        config.append(newHere)
        f.close
    with open("./static/bmiData.json", 'w') as fw:
        fw.truncate()
        json.dump(config, fw)
        fw.close
    return "success"
```

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。

```
public void initItems() {
    new Thread(new Runnable() {
        @Override
        public void run() {
            item[] itemsHere = new item[maxNumb];
            int k = 0;
            BufferedReader reader;
            StringBuilder jsonData = new StringBuilder();
            HttpURLConnection connection;
            try {
                URL requestUrl = new URL("https://southstem.cloud/getBmiData");
                connection = (HttpURLConnection) requestUrl.openConnection();
                connection.setRequestMethod("GET");
                connection.setConnectTimeout(5000);
                connection.setReadTimeout(8000);
                if (connection.getResponseCode() == 200) {
                    InputStream in = connection.getInputStream();
                    reader = new BufferedReader(new InputStreamReader(in));
                    String line;
```

```

        while ((line = reader.readLine()) != null) {
            jsonData.append(line);
        }
    }
    JSONArray jsonArray = new JSONArray(jsonData.toString());
    for (int i = 0; i < jsonArray.length(); i++) {
        JSONObject jsonObject = (JSONObject) jsonArray.get(i);
        item newItemHere = new
item(Float.parseFloat(jsonObject.getString("height")),
Float.parseFloat(jsonObject.getString("weight")),
                                jsonObject.getString("time"));
        itemsHere[k] = newItemHere;
        k++;
    }
    connection.disconnect();
} catch (MalformedURLException | ProtocolException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
} catch (JSONException e) {
    e.printStackTrace();
}
items = itemsHere;
}
}).start();
}

```

### (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

```

initItems();
while (items == null) ;
adapter = new RecyclerViewAdapter(items);
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。

```

public void submit(View view) {
    RE.scrollToPosition(0);
    int k = 0;

    EditText weightHereStr = (EditText) findViewById(R.id.weight);
    EditText heightHereStr = (EditText) findViewById(R.id.height);
    String weightHere = weightHereStr.getText().toString();
    String heightHere = heightHereStr.getText().toString();
    float weight = Float.parseFloat(weightHere);
    float height = Float.parseFloat(heightHere);
    SimpleDateFormat formatter = new SimpleDateFormat(" yyyy.MM.dd");
    new Thread(new Runnable() {
        @Override
        public void run() {
            if (!weightHere.isEmpty() && !heightHere.isEmpty() && flag == false) {
                try {
                    String uu = "添加记录: " + weightHere + "kg , " + heightHere + "m
成功。";

```

```

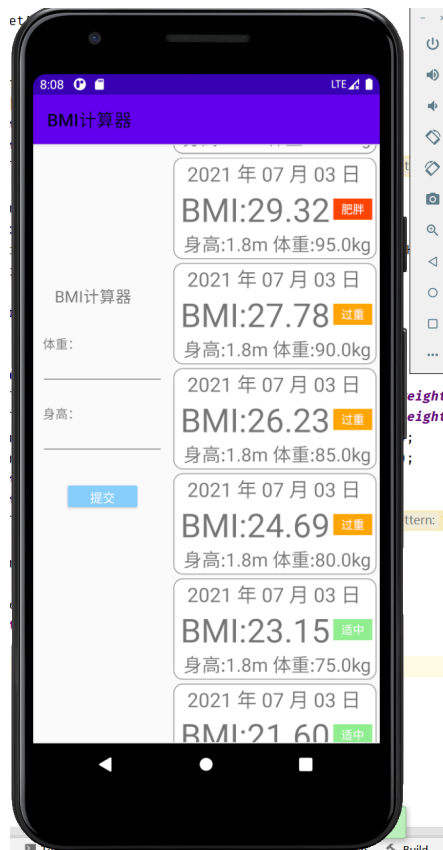
        URL reqUrl = new URL("https://southstem.cloud/addNewBmi?
height=" + heightHere + "&weight=" + weightHere);
        HttpURLConnection connection2 = (HttpURLConnection)
reqUrl.openConnection();
        connection2.setRequestMethod("GET");
        connection2.setConnectTimeout(5000);
        connection2.setReadTimeout(8000);
        if (connection2.getResponseCode() == 200) {
            resultStr = uu;
        } else {
            resultStr = "ERROR!";
        }
        connection2.disconnect();
    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (ProtocolException protocolException) {
        protocolException.printStackTrace();
    } catch (IOException ioException) {
        ioException.printStackTrace();
    }
    // adapter.notifyDataSetChanged();
} else {
    resultStr = "ERROR";
}
flag = true;
}
}).start();
Date curDate = new Date(System.currentTimeMillis());
String timeHere = formatter.format(curDate);
item itemNew = new item(height, weight, timeHere);
adapter newItem(itemNew);
}

```

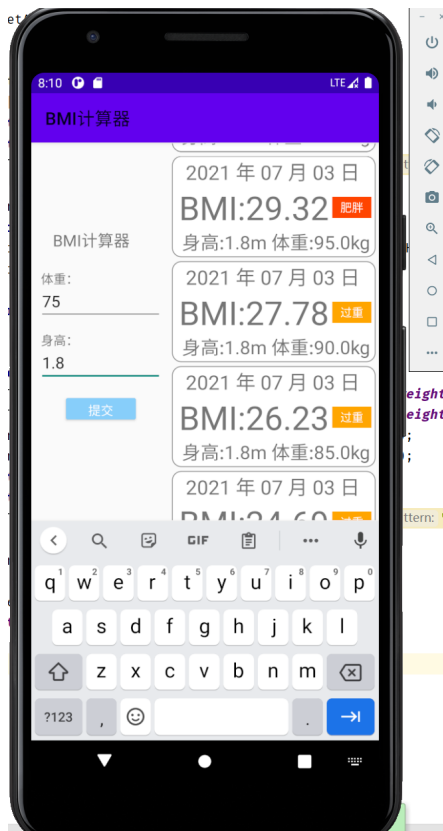
代码部分完成

### 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。查看服务器返回json数据，已经成功添加，测试完成。

```
[pid: 53974|app: 0|req: 2/2] 106.122.185.214 () {42 vars in 638 bytes} [Tue Jul 6 23:33:37 2021] GET /addNewBmi?height=1.8&weight=75 => generated 7 bytes in 3 msecs (HTTP/1.1 200) 2 headers in 78 bytes (1 switches on core 0)
```

源码：[GitHub : Southstem/ BMICalculator](#)

## 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。