

# 浙大2019暑期Android 课程方案第五讲——网络





# 目录

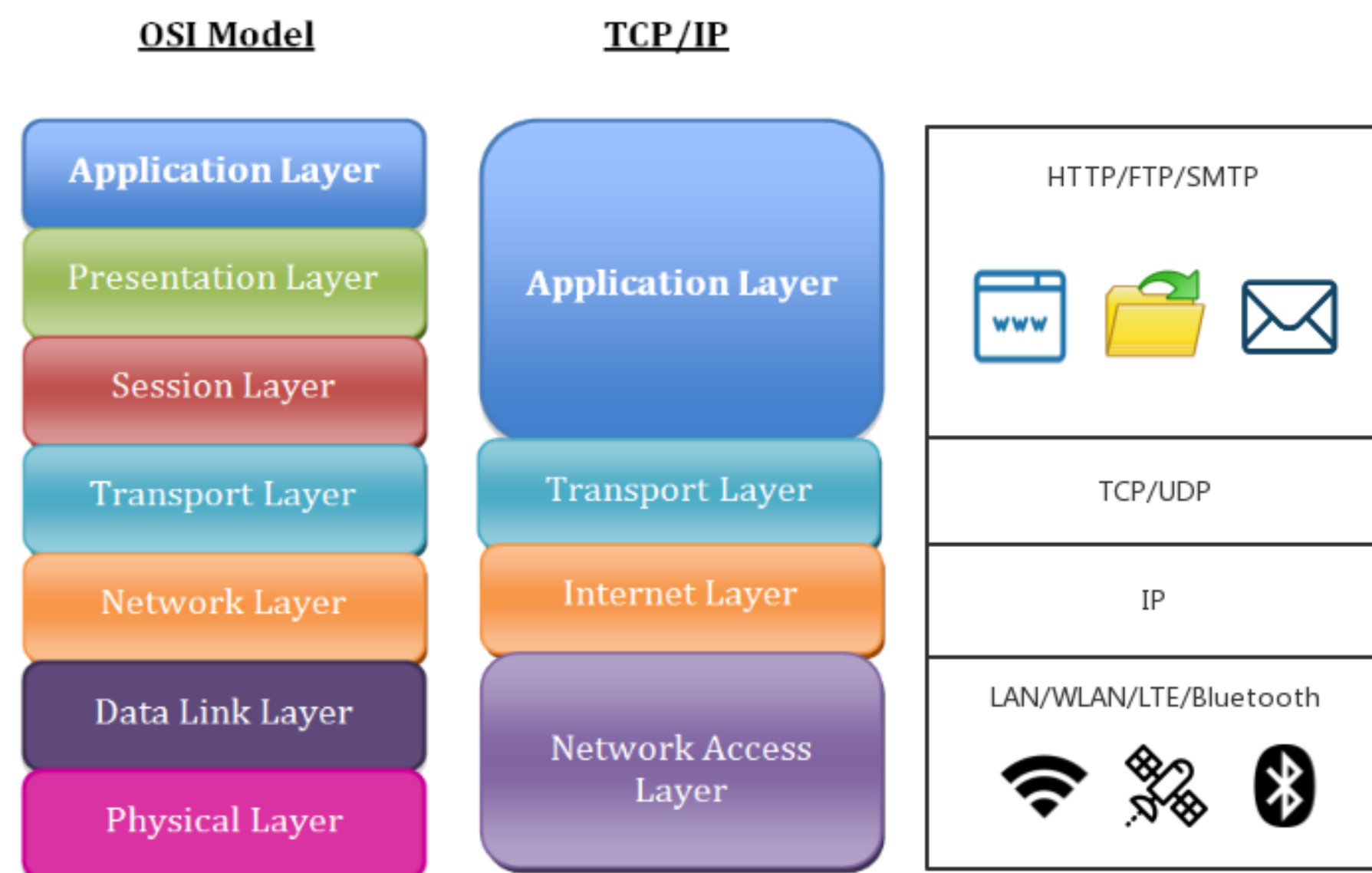
## 一、计算机网络基础

1. 经典网络分层
2. http协议格式
3. 调试一个RestFul API
4. 常用的数据交换格式

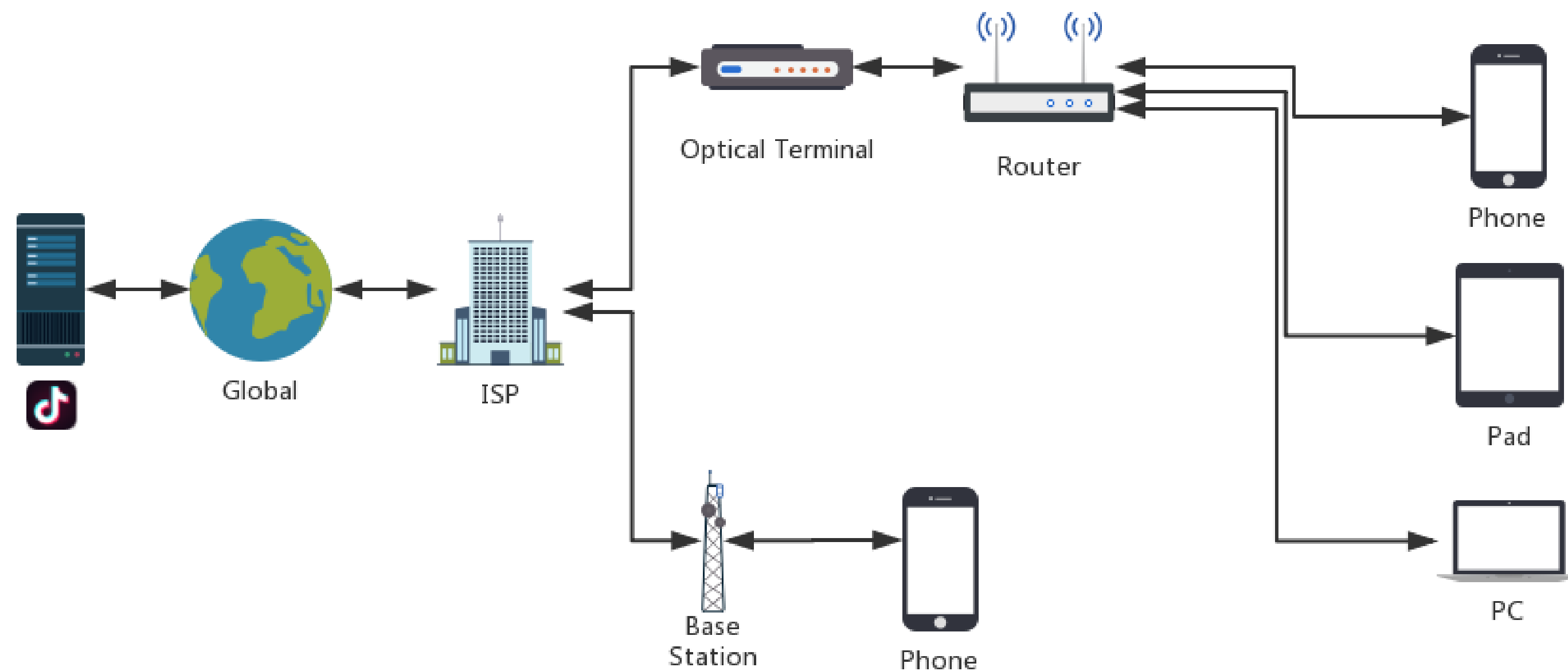
## 二、访问一个网络接口的安卓Demo

1. 网络权限申请
2. 网络请求
3. 结果解析
4. 数据展示

# 1.1、计算机网络基础



网络分层理论模型



实际使用物理模型

以太网交换机、基站、路由器、操作系统等，已经把物理层到TCP层封装完善；  
普通开发者，大部分只需要关心应用层的数据交换，如基于http、FTP、SMTP等协议做业务开发。

## 1.2、http协议格式

```
http://example.com/animal/mammal/primate/tarsier?diet=carnivore&active=night
```

The diagram shows the URL `http://example.com/animal/mammal/primate/tarsier?diet=carnivore&active=night` with green boxes highlighting its parts. Labels with green arrows point to these parts: `http://` is labeled "protocol/scheme"; `example.com` is labeled "host/domain/authority"; `/animal/mammal/primate/tarsier` is labeled "resource path"; and `?diet=carnivore&active=night` is labeled "query".

host/domain/authority

query

protocol/scheme

resource path

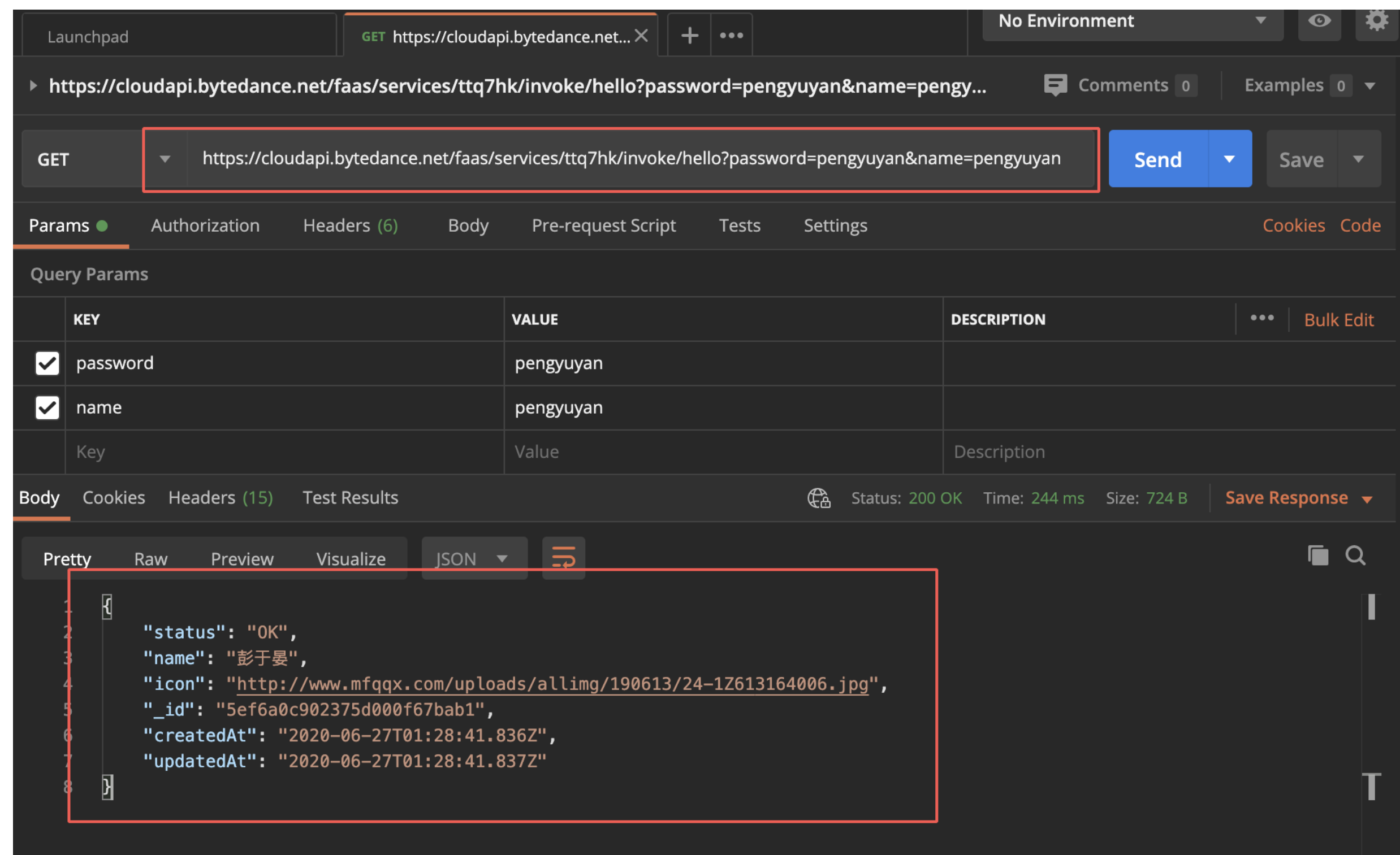
日常工作里，最常见的http请求协议格式

# 1.3、调试一个RestFul API

服务端部署地址：

<https://larkcloud.bytedance.net/dashboard/ttq7hk/function/hello?listSize=20&senderSize=25&logSize=30>

提供服务的url： <https://cloudapi.bytedance.net/faas/services/ttq7hk/invoke/hello>



直接输入拼接好的地址会得到什么？

<https://cloudapi.bytedance.net/faas/services/ttq7hk/invoke/hello?name=pengyuyan&password=pengyuyan>



对应我们上一页的，  
**协议、域名、资源路径、参数分别是什么**

使用postman调试，可以看body部分和header部分

# 1.4、常用的数据交换格式

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<resources>
```

```
<color name="colorPrimary">#6200EE</color>
```

```
<color name="colorPrimaryDark">#3700B3</color>
```

```
<color name="colorAccent">#03DAC5</color>
```

```
</resources>
```

```
{  
  "name": "John Doe",  
  "age": 18,  
  "address": {"country" : "china", "zip-code": "10000"}  
}
```

```
message Person{  
  required string name = 1; //姓名  
  required int32 sex = 2; //性别  
  required int32 age = 3; //年龄  
}
```

XML：标签对

JSON：key-value

Protobuf：谷歌提出的一个  
更加精简的数据传输格式，

[https://en.wikipedia.org/wiki/Protocol\\_Buffers](https://en.wikipedia.org/wiki/Protocol_Buffers)

扩展阅读：

1、在浏览器地址栏输入一个URL后回车，背后会进行哪些技术步骤？

<https://www.zhihu.com/question/34873227>

2、https是如何保证数据传输安全，不被中间过程拦截&篡改的？

[https://blog.csdn.net/guolin\\_blog/article/details/104546558](https://blog.csdn.net/guolin_blog/article/details/104546558)



# 目录

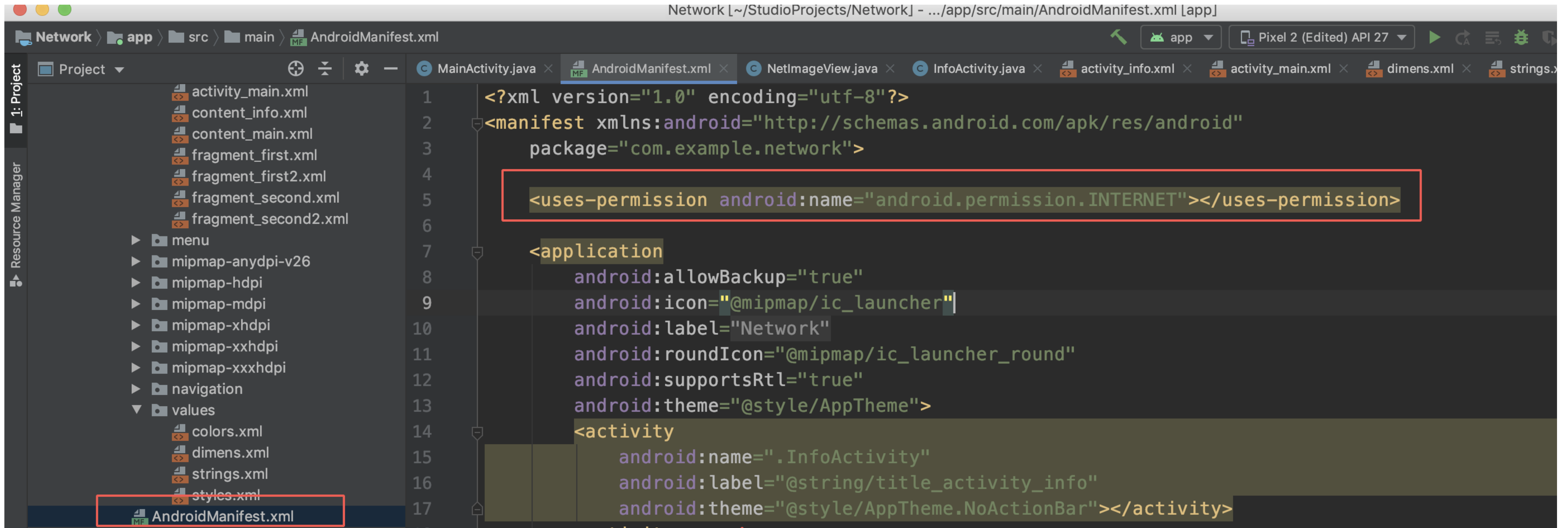
## 一、计算机网络基础

1. 经典网络分层
2. http协议格式
3. 调试一个RestFul API
4. 常用的数据交换格式

## 二、访问一个网络接口的安卓Demo

1. 网络权限申请
2. 网络请求
3. 结果解析
4. 数据展示

## 2.1、网络权限申请



Manifest定义里，可以尝试去掉网络权限申明，看看会发生什么？

安卓权限的说明，参见其他课程。



## 2.2、网络请求

```
public static String login(String username,String password){
    String msg = "";
    try {
        username = URLEncoder.encode(username, enc: "UTF-8");
        password = URLEncoder.encode(password, enc: "UTF-8");
    } catch ( UnsupportedEncodingException e1) {
        e1.printStackTrace();
    }
    String urlStr="https://cloudapi.bytedance.net/faas/services/ttq7hk/invoke/hello?name="+username+"&password="+password;
    try{
        URL url =new URL(urlStr);
        HttpURLConnection conn = (HttpURLConnection)url.openConnection();
        conn.setConnectTimeout(6000);
        InputStream in = conn.getInputStream();
        BufferedReader reader = new BufferedReader(new InputStreamReader(in, StandardCharsets.UTF_8));
        String line = null;
        while ((line = reader.readLine()) != null) {
            if(msg==null){
                msg=line;
            }else{
                msg += line;
            }
        }
        reader.close();
        in.close();//关闭数据流
        conn.disconnect();
    }catch(Exception e){
        e.printStackTrace();
        return null;
    }finally { //做清理操作 }
    return msg;
}
```

核心case:

- 1、明确要访问的服务API
- 2、填充访问参数
- 3、访问网络流
- 4、获得返回结果

现在有更丰富的开源库，如  
okhttp等对网络请求做了更好的  
封装和使用简化。

## 2.3、结果解析

```
try {
    JSONObject jsonObject = new JSONObject(resultJson);
    String status = jsonObject.getString( name: "status");
    Log.i(TAG, msg: "status:"+status);
    if( "OK".equals(status) ){
        String name = jsonObject.getString( name: "name");
        String iconUrl = jsonObject.getString( name: "icon");
        Log.i(TAG, msg: "name:"+name+" iconUrl:"+iconUrl);

        Intent intent = new Intent();
        intent.putExtra(InfoActivity.NAME_EXTRA, name);
        intent.putExtra(InfoActivity.ICON_URL_EXTRA, iconUrl);
        intent.setClass( packageContext: MainActivity.this, InfoActivity.class);
        startActivity(intent);

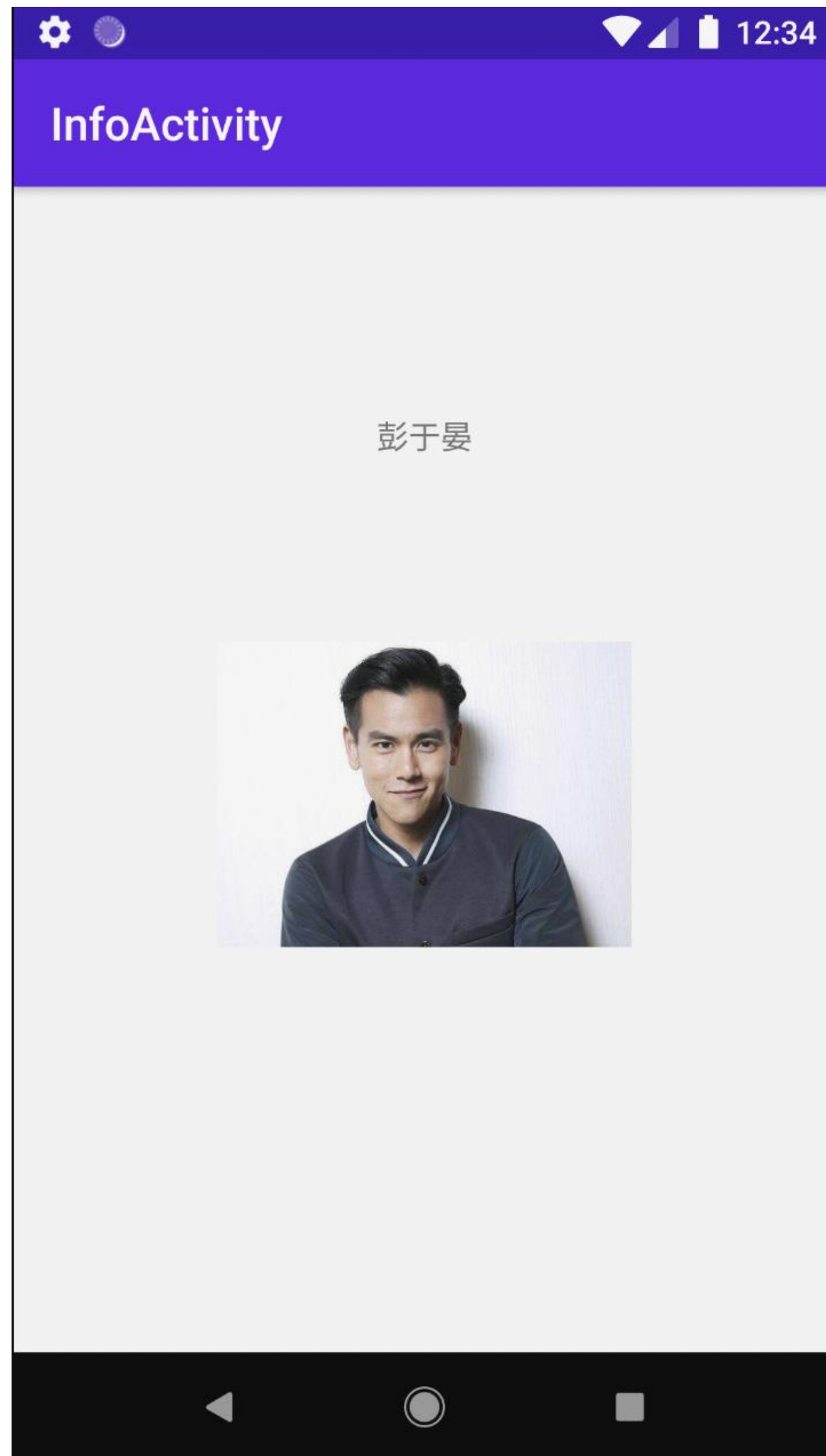
    }else{
        Log.i(TAG, msg: "wrong status");
        Toast.makeText( context: MainActivity.this, text: "错误的用户名或者密码", Toast.LENGTH_LONG).show();
    }
} catch (JSONException e) {
    e.printStackTrace();
}
```

本次约定的是JSON格式交互，因此拿到返回的string之后，先转换为jsonObject。

然后根据status，决定走什么逻辑。

```
{
  "status": "OK",
  "name": "彭于晏",
  "icon": "http://www.mfqqx.com/uploads/allimg/190613/24-1Z613164006.jpg",
  "_id": "5ef774483606ee000f4b3975",
  "createdAt": "2020-06-27T16:31:04.467Z",
  "updatedAt": "2020-06-27T16:31:04.467Z"
}
```

## 2.4、数据展示



用户名和密码正确时的跳转



用户名和密码错误时的提示



# 衍生问题

- 1、如果是抖音那种需要连续滑动、立即加载的，客户端与服务端的交互格式如何设计？  
——即如何设计预加载策略？
- 2、Demo里，为什么把网络请求放到一个独立的线程中？  
——如果放到主线程，是否会报错？  
——你觉得安卓系统如此设计的原因是什么？
- 3、你觉得json和XML相比，数据传输是否有优势？
- 4、Restful请求里，get和post区别？

# 课后练习

基于Demo，完成一个上传图片和视频的APK

# 附源码

字节服务部署：

<https://larkcloud.bytedance.net/share/tt7217xbo2wz3cem41h64fy8w>

Demo下载：

<https://github.com/pengshaohua1987/NetworkDemo.git>