

最近在写一个小项目，需要涉及到前端向后端发送一个jsonArray，然后我们后端采用的是node的koa框架，所以需要http发送，然后这个http包含着一个content即可

移动端的发送代码

```
JSONArray jsonArray = new JSONArray();
for (int i = 0; i < 5; i++) {
    JSONObject object = new JSONObject();
    try {
        object.put("user_id", "1111111111");
        object.put("user_name", "2222222222");
        object.put("user_phone", "3333333333");
        object.put("user_address", "44444444444444");
        object.put("product_id", "5555555555");
        object.put("product_name", "66666666666666");
        object.put("product_price", "777777777777");
        object.put("product_count", "888888888888");
    } catch (JSONException e) {
        Log.i("lin", "---lin's log---> 进入 catch");
        e.printStackTrace();
    }
    jsonArray.put(object);
}
String content = jsonArray.toString();
OkHttpUtils
    .postString()
    .url("http://172.20.10.4:3008/testjson")
    .content(content)
    .build()
    .execute(new StringCallback() {
        @Override public void onError(Request request, Exception e) {
            Toast.makeText(TestActivity.this, "error", Toast.LENGTH_SHORT).show();
            Log.i("lin", "---lin's log---> error " + e.toString());
        }

        @Override public void onResponse(String response) {
            Toast.makeText(TestActivity.this, "onResponse", Toast.LENGTH_SHORT).show();
            Log.i("lin", "---lin's log---> response " + response);
        }
    });
```

后端接收的代码:

```
var koa = require('koa');
var controller = require('koa-route');
var parse = require('co-body');
var app = koa();

app.use(controller.post('/testjson', function* () {
    console.log("接收到请求~");
    var item = yield parse(this);
    var jsonList = eval(item);
    for (var i = 0; i < jsonList.length; i++) {
        console.log(jsonList[i].user_id);
        console.log(jsonList[i].user_name);
    }

    this.set('Cache-Control', 'no-cache');
    this.body = "100";
}));
```

```
Run app.js
/usr/local/Cellar/node/6.3.1/bin/node /Users/mac/WorkSpace/毕设/温馨后端/app.js
Koa server is started!
(node:5030) DeprecationWarning: `DEBUG_FD` is deprecated. Override `debug.log` if you want to use a different log function (https://git.io/vMlYr)
111222333444
11111111111
22222222222
11111111111
22222222222
11111111111
22222222222
11111111111
22222222222
11111111111
22222222222
11111111111
22222222222
```

以上便可以实现，前端向后端发送jsonArray并将其解析的过程啦~