Flutter 仿京东商城项目签名验证原理、签名验证算法

主讲教师：（大地）
合作网站：www.itying.com （IT 营）
我的专栏：https://www.itying.com/category-79-b0.html

目录

1、Flutter Md5 加密

```
import 'dart:convert';
import 'package:crypto/crypto.dart';
main() {
  print(md5.convert(utf8.encode("Hello")));
}
```

2、为什么要签名验证

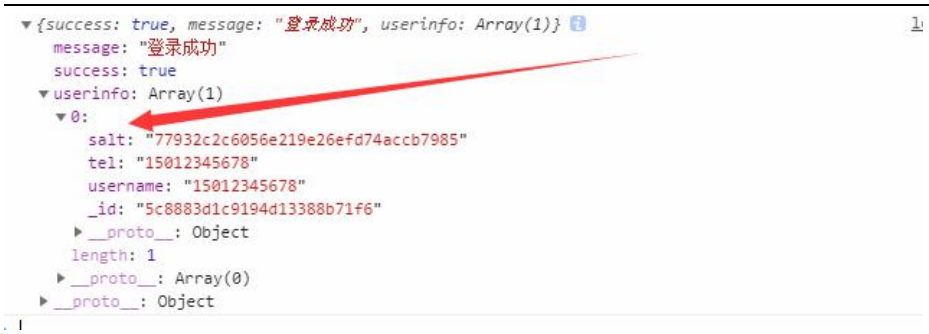
我们通过 http Post 或者 Get 方式请求服务器的时候，会面临着许多的安全性问题，**例如：**

1、请求来源(身份)是否合法？
2、请求参数被篡改？
3、请求的唯一性(不可复制)

项目中用户登录后以后才能访问的信息，请求 api 接口的时候为了安全，需要做签名验证。

3、签名验证实现原理

1、用户登录成功后服务器会返回用户信息以及 salt



salt 是用户注册的时候随机生成的字符串然后通过 md5 加密得到的，每个用户的 salt 不一样

2、请求接口的时候在接口中加入 sign 签名

如以前的请求方式：

http://jd.itying.com/api/addressList?uid=5a18fe9983796b0dc0542f99

现在的请求方式：

http://jd.itying.com/api/addressList?uid=5a18fe9983796b0dc0542f99&sign=fee452295f3a1d40ee90dc8e974885e9

3、sign 签名的生成算法

1、把请求接口的所有参数以及 salt 进行排序，然后拼接成字符串后用 Md5 加密。算法如下：

```
import 'dart:convert';
import 'package:crypto/crypto.dart';

getSign() {
        Map json = {
          "aid": 1,
          "name": 'zhangsan',
          "age": 20,
          "sex": '男',
          "salt":"xxxewrewqrqrwqrwqrwqr"    //私钥
       };
       List jsonKeys = json.keys.toList();
       //按照 ASCII 字符顺序进行升序排列（也就是所谓的自然顺序）
       jsonKeys.sort();
       var str = '';
       for (var i = 0; i < jsonKeys.length; i++) {
          str += "${jsonKeys[i]}${json[jsonKeys[i]]}";
```

```
    }

    print(md5.convert(utf8.encode(str)));
  }
```

4、请求接口传入 sign

http://jd.itying.com/api/addressList?uid=5a18fe9983796b0dc0542f99&sign=fee452295f3a1d40ee90dc8e974885e9

5、服务器端生成签名验证：

1. 获取客户端传过来的 sign 和 参数
2. 根据 uid 去数据库查询当前用户的 salt（32 位）
3. url 获取的数据和数据库查询的 salt 组合成 json 用同样的算法生成签名
4. 用服务器的签名和客户端的做对比如果一样表示没有篡改。

6、请求的唯一性解决方案：

为了防止别人重复使用请求参数问题，我们需要保证请求的唯一性，就是对应请求只能使用一次，这样就算别人拿走了请求的完整链接也是无效的。
唯一性的实现：在如上的请求参数中，我们加入时间戳 ：timestamp（yyyyMMddHHmmss），同样，时间戳作为请求参数之一，也加入 sign 算法中进行加密。

服务器获取到客户端传入的时间戳和本地时间做对比，如果两个时间的差值大于一个值，表示请求是无效的。

如何解决时间差问题：

1、第一次打开应用获取本地时间，然后请求接口获取服务器时间。
2、把时间差保存到本地存储
3、请求接口的时候把本地时间和时间差相加。

客户端请求的地址：

http://39.108.159.135/api/oneAddressList?uid=5a18fe9983796b0dc0542f99&sign=fee452295f3a1d40ee90dc8e974885e9

其中:sign=(32位服务器返回的salt)+uid

1.服务器获取客户端传过来的uid，去数据库查询当前uid对应的32位salt

2.服务器通过同样的算法生成 sign =（32位salt）+uid

3.服务器用获取的sign和自己生成的做对比，如果不一样表示签名不一样