

DR-API 安全登录使用手册

版本号	V0.1
作者	刘永波 (liuyongbo@baidu.com)
更新日期	2012-04-10

百度在线网络技术（北京）有限公司
(版权所有, 翻版必究)

目 录

DR-API 安全登录使用手册.....	1
目 录.....	1
1 概述.....	2
2 数据传输规范.....	2
2.1 请求规范.....	2
2.1.1 消息头规范.....	2
2.1.2 消息体规范.....	2
2.2 服务器响应结构说明.....	3
2.2.1 消息头规范.....	3
2.2.2 消息体规范.....	4
3 方法的数据结构规范.....	4
3.1 preLogin.....	4
3.2 doLogin.....	5
3.3 verifyQuestion.....	6
3.4 doLogout.....	6

1 概述

DR-API，即百度商业推广 API，是百度商业产品统一对公司外服务的开放 API，目前包括百度推广 API 和网盟推广 API。

本手册旨在明确 DR-API 使用方使用安全登录时需要遵循的规范和注意事项，作为 DR-API 的使用方的开发指南。

2 数据传输规范

2.1 请求规范

2.1.1 消息头规范

DR-API 挂接方发送的消息应该遵循如下规范：



每一个消息都包含了消息头和数据。消息头由 8 个字节组成，分成 4 个整数，每个整数包含 2 个字节高位在前。

第一个整数表示 client id，由 DR-API 统一分配。

第二个整数表示 encrypt version，由 DR-API 统一分配。

第三个和第四个整数为保留整数，暂不使用。

消息头后面紧跟着消息数据。数据采用先压缩后加密的形式：



挂接方需要首先将消息转换成 JSON 格式，编码采用 UTF8，然后用 GZIP 对消息 JSON 进行压缩。压缩完毕后，采用 RSA 算法的公钥进行加密。

公钥由 DR-API 统一分配，公钥暴露后可以再次向 DR-API 申请新的公钥，并增加 encrypt version 版本号。

2.1.2 消息体规范

消息体包含前置字段和方法真实请求体两个部分。

其整体格式如下：



消息体按照竖线|分割为 5 个部分，分别是：

用户名|权限代码|方法名|全局唯一性 ID|JSON（方法真实请求体）

使用方必须严格按照上述格式进行消息体组织。

2.2 服务器响应结构说明

2.2.1 消息头规范

服务器返回的消息结构和请求消息结构比较类似：



其中除了消息头的第一个字段是 RETURN CODE 外，其他结构完全与请求消息一致。

RETURN CODE 是服务器返回的总错误代码。当其不等于 0 时，消息体将不存在。也就是说使用方仅仅会收到 8 个字节的消息头。

RETURN CODE 详细说明：

RETURN CODE	详细说明
0	正常
1	CLIENT ID 错误
2	加密方法错误
3	数据体损坏
4	数据太大（超过 2K）
5	数据太小
6	请求消息体格式不正确

7	访问的方法不存在
8	方法处理出错
9	TOKEN 错误
10	用户名错误
11	方法执行过程中出现错误

2.2.2 消息体规范

服务器返回的消息体里面直接就是对应方法返回值的 JSON 字符串, 编码采用 UTF8, 然后用 GZIP 对消息 JSON 进行压缩。压缩完毕后, 采用 RSA 算法的私钥进行加密:



使用方收到后直接用由 DR-API 分配的公钥进行解密即可。

3 方法的数据结构规范

3.1 preLogin

方法作用:

当用户输入完毕用户名时, 由客户端调用此方法, 判断是否需要出验证码。

使用方在访问该方法时, 需要遵循如下规范:

1. 请求参数格式为:

```
PreLoginRequest {  
    // 客户端载体操作系统  
    private String osVersion;  
    // 客户端载体类型  
    private String deviceType;  
    // 客户端版本  
    private String clientVersion;  
}
```

2. 服务器返回消息格式为:

```
PreLoginResponse {
    // 是否需要验证码
    private boolean needAuthCode;
    // 验证码详细内容

    private AuthCode authCode;
}

AuthCode {
    // 图片的格式，比如JPG类型的图片，就是JPG三个字母
    private String imgtype;
    // 图片的二进制内容，base64编码
    private String imgdata;
    // 图片会话id
    private String imgssid;
```

3.2 doLogin

方法作用：

当用户输入完毕密码和验证码（可选）时调用该方法实现真正登录。

使用方在访问该方法时，需要遵循如下规范：

1. 请求参数格式为：

```
DoLoginRequest {
    // 用户输入密码
    private String password;
    // 验证码
    private String imageCode;
    // 验证码会话id
    private String imageSsid;
```

2. 服务器返回消息格式为：

```
DoLoginResponse {
    // 0：成功，131：验证码错误，134强制修改密码，135：该用户被锁定，191：该用户需要回答密
    // 保问题，3：登陆IP被封禁，133：用户不存在，132：用户名或密码错误，502：参数错误，600：业务
    // 系统不允许登录(返回信息可能为多种)，601：应用系统注册信息不完整
    private int retcode;
    // 错误信息
    private String retmsg;
    // 用户ucid
    private long ucid;
    // 会话ID
    private String st;
    // 是否是token登陆用户
    private int istoken;
    // 是否需要设置Pin码
```

```
private int setpin;
// 安全问题列表
private List<Question> questions;
Question {
    // 安全问题ID
    private int qid;
    // 安全问题字面
    private String content;
```

3.3 verifyQuestion

方法作用:

当用户触发了安全规则时，调用登录接口会要求回答安全问题，并同时会返回一个问题列表。

客户端自行在界面显示问题列表，由用户选择一个问题进行回答时调用本接口。

使用方在访问该方法时，需要遵循如下规范:

1. 请求参数格式为:

```
VerifyQuestionRequest {
    // 用户ucid
    private long ucid;
    // 会话ID
    private String st;
    // 用户回答的安全问题ID
    private int qid;
    // 用户输入安全问题答案
    private String answer;
```

2. 服务器返回消息格式为:

```
VerifyQuestionResponse {
    // 0: 成功, 3: 登陆IP被封禁, 192, 回答错误 193: 回答错误次数已超限制 190: 会话无效 502:
参数错误
    private int retcode;
    // error具体信息
    private String retmsg;
    // 记录已经回答错了几次
    private int errortime;
```

3.4 doLogout

方法作用:

登出。

使用方在访问该方法时，需要遵循如下规范:

1. 请求参数格式为:

```
DoLogoutRequest {  
    // 用户ucid  
    private long ucid;  
    // 会话ID  
    private String st;
```

2. 服务器返回消息格式为:

```
DoLogoutResponse {  
    // 0, 表示成功, 1: 失败, 502: 参数错误  
    private int retcode;  
    // error具体信息  
    private String retmsg;
```