

1、下单购买套餐包

打开此链接：<https://market.aliyun.com/products/57124001/cmapi00062195.html?#sku=yuncode5619500001>

根据需要购买套餐包。

购买完成后，进入阿里云控制台-云市场（链接：<https://market.console.aliyun.com>）可以看到已经购买的商品。

商品信息卡片上会显示AppCode。



The screenshot shows the Alibaba Cloud Marketplace console interface. At the top, there is a navigation bar with the Alibaba Cloud logo and a '工作台' (Dashboard) button. Below this, a sidebar on the left contains a menu with options: '云市场' (Cloud Market), '已购买的服务' (Purchased Services), '订单列表' (Order List), '未支付订单' (Unpaid Orders), '需求管理' (Requirement Management), '定制方案管理' (Customized Solution Management), '退款管理' (Refund Management), '发票管理' (Invoice Management), '优惠券管理' (Coupon Management), '合同管理' (Contract Management), and '我的关注' (My Favorites). The main content area is titled '已购买的服务' (Purchased Services) and features a search bar for '商品名称' (Product Name). Below the search bar, there is a table of purchased services. The first service is '【友盟+】智能认证 U-Verify (按量计价)' (U-Verify Pay-as-you-go). It includes a '设置备注' (Set Remark) button, a '付费方式: 按量付费' (Payment Method: Pay-as-you-go) label, and a '创建时间: 2020-02-20' (Creation Time: 2020-02-20) label. The 'AppKey: 203724081' and 'AppSecret:' fields are visible, with a '复制' (Copy) button next to the AppSecret. The 'AppCode:' field is highlighted with a red box, and a '复制' (Copy) button is also present next to it. The second service is '【友盟+】智能认证 U-Verify (一键登录)' (U-Verify One-click Login), which also includes a '设置备注' (Set Remark) button, a '付费方式: 按量付费' (Payment Method: Pay-as-you-go) label, and a '创建时间: 2019-11-27' (Creation Time: 2019-11-27) label.

2、请求构造

METHOD : POST

URL: <https://dmcommon.market.alicloudapi.com/api/v3/featureMatrix>

参数名	位置	类型	是否必填	描述	示例	可选值
idValues	body	[string]	是	32位小写md5 id	["8ba8cdf66c4969f7ca56e8c4034acaa6"]	
idTypes	body	[string]	是	对应idValues的id类型	["mobile_salt"]	"imei", "idfa", "oaid", "aaid", "mobile_salt", "caid"
os	body	string	是	id对应的系统, 不清楚是固定填"iOS"	"iOS"	"iOS", "Android"
Authorization	header	string	是	下单后获得的APPCODE	APPCODE 1234567890	

注意事项

1. idValues可填写指向同一个人的多个值（例如这个人既有oaid，又有mobile_salt），idTypes里的值需要和idValues一一对应，api随机返回一条id对应的结果
2. os不清楚时，可固定填写iOS
3. mobile_salt为加盐md5格式的手机号，使用如下方法得到

```
md5(concat(mobile, '38e14e97300177b6ec15eb12d9f8b19a'))
```

3、返回结果

1、成功匹配

```
{
  "success": true,
  "data": {
    "result": {
      "emb001": "a4e1aa8:0.198036",
      "emb002": "bbewaa1:0.851749",
      "emb003": "aycgaaBA1:0.058,aycgaa1AEC:0.232,aycgaa19E6:0.449",
      "emb004": "aufpaa2:0.629072",
      "emb005": "a5fvaa1:0.654627",
      "emb006": "asf9aa1:0.654627",
      "emb007": "bae3aa1:0.629933",
      "emb008":
"aga4aa19:1,aga4aa10:1,aganaa21:1,aga4aaDB:1,aga8aa21:100,aga4aa29:1,aga4aaC1:1,aga4aaC
:1,agagaa21:100,aga4aa41:1,aga4aa26E176:1,aga4aa50:1,aga4aa12:1,agbpaa21:100,aga9aa22:1
00,aga4aa411:1,agaeaa21:1,agaxaa22:100,aganaa43:1,agacaa21:1,aga4aa5328:1,aga4aa3870:1,
agajaa43:66,aga4aa4C:1,agacaa246:1,aga4aa31:1,agaeaa22:1,aga4aa16:1,aga4aa1A:1,agacaa22
:1,aga4aa11C:1,aga4aa3F:1,aga4aa139:1,aganaa0:3,aga4aa2:1,aga4aaE:1,aga4aa7:1,aga4aaD:1
,agajaa22:100,agadaa0:2,aga4aaB:1,aga4aa34:1,agawaa21:100,agacaa0:3,aga4aaD6BE:1,aga4aa
51:1,aga4aaB1:1,agawaa246:100,agagaa22:100,aga4aa11:1,aga5aa22:100,aga4aa43:1,agaxaa246
:100,aga4aa3:1,aga4aa54:1,aga4aa9:1,aga4aa47:1,agbpaa22:100,agblaa43:66,aga4aa55:1,aga4
aa62:1,aganaa22:1,aga4aa39:1,agaxaa21:100,agadaa21:1,agbmaa43:66,agaoaa0:1,aga4aa13:1,a
gaeaa0:3,aga4aa73:1,aga8aa246:100,agadaa22:1,agawaa22:100,aga5aa21:100,aga4aaF:1,agaoaa
43:1,agbpaa43:66,aga9aa21:100,aga4aa5A:1,aga4aa28:1,agaeaa246:1,aga4aa15:1,agajaa21:100
,aga4aa2C:1,aga4aaC3:1,aga8aa22:100,agagaa246:100,aga4aa6:1,aga4aa8:1",

```

```
"emb050": "0.020333,0.131285,-0.003566,-0.025359,-0.020994,-0.079587,-0.002986,-0.07679
3,0.043425,0.044873,-0.013856,0.013480,-0.005446,0.094951,-0.017466,-0.032542,-0.077701
,0.021758,-0.052249,-0.053622,0.247052,-0.019724,0.009413,-0.137654,-0.019316,0.096402,
-0.139922,-0.009739,0.087534,0.026346,-0.012827,0.174347,0.063401,-0.010701,0.028131,0.
064910,0.005111,-0.095708,0.025809,0.041618,-0.735710,0.060773,0.005222,-0.077392,0.049
395,0.087206,0.000459,-0.031782,-0.047220,0.030850,0.034555,-0.046890,0.052481,0.018358
,-0.040946,-0.025597,0.013875,0.002403,0.045600,-0.041250,0.041652,0.002611,-0.069658,-
0.019575,-0.036048,0.015771,-0.051786,0.024761,-0.077371,0.021941,0.004357,-0.023370,0.
042612,0.075690,0.054230,0.151215,0.080530,0.032406,-0.000021,-0.062288,0.027240,-0.039
251,0.005973,-0.062103,-0.030945,0.027527,0.018306,-0.023700,-0.010959,0.027699,-0.0392
51,0.055747,-0.061881,-0.033866,0.036029,0.080869,-0.050549,-0.001701,-0.035838,-0.0648
80,-0.027691,0.008220,0.021276,-0.126681,-0.032930,0.038086,-0.032783,0.004173,-0.06426
0,0.000483,-0.143332,-0.020761,0.041999,0.041678,0.078238,-0.064271,0.041672,0.043300,-
0.063548,-0.081289,0.093816,-0.053116,0.023519,-0.073783,-0.039379,-0.023666,0.012714,-
0.044558"
  },
  "idTypes": [
    "mobile_salt"
  ],
  "idValues": [
    "8ba8cdf66c4969f7ca56e8c4034acaa6"
  ],
  "os": "iOS",
  "matchStatus": true
},
"code": 200,
"message": null,
"requestId": null
}
```

2、未匹配

```
{
  "success": false,
  "data": null,
  "code": 501,
  "message": "未匹配成功",
  "requestId": "17FF9772-5BDE-4C41-8280-EC3C6687CB9C"
}
```

3、注意事项

1. 只有在http状态码为200时，能匹配并返回结果data，进行扣费
2. http状态码非200时，data为null，不会扣费
3. 按次成功计费，即使多次请求同一个id，也会多次计费

4、java代码示例

1、pom依赖

向pom.xml引入相关依赖

```
<dependency>
  <groupId>com.alibaba.fastjson2</groupId>
  <artifactId>fastjson2</artifactId>
  <version>2.0.24</version>
</dependency>
<dependency>
  <groupId>org.apache.httpcomponents</groupId>
  <artifactId>httpclient</artifactId>
  <version>4.5.13</version>
</dependency>
```

2、请求代码

```
package com.umeng.datamet.api.demo;

import com.alibaba.fastjson2.JSONObject;
import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.entity.StringEntity;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.impl.conn.PoolingHttpClientConnectionManager;
import org.apache.http.util.EntityUtils;

import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.util.Objects;
import java.util.concurrent.TimeUnit;

public class Test2 {

    public static CloseableHttpClient httpClient = null;
```

```

static {
    PoolingHttpClientConnectionManager connectionManager = new
PoolingHttpClientConnectionManager(60, TimeUnit.SECONDS);
    connectionManager.setMaxTotal(1000);
    connectionManager.setDefaultMaxPerRoute(500);

    httpClient = HttpClients.custom()
        .setConnectionManager(connectionManager)
        .disableAutomaticRetries()
        .build();
}

public static void main(String[] args) {
    //接口的url、appCode、请求body待填写
    // TODO
    String url = "";
    // TODO
    String appCode = "";
    // TODO
    JSONObject body = new JSONObject();

    JSONObject responseBody = postOnce(url, appCode, body);
    assert responseBody != null;
    System.out.println(responseBody.toJSONString());
}

private static JSONObject postOnce(String url, String appCode, JSONObject body) {
    CloseableHttpResponse response = null;
    HttpPost httpPost = new HttpPost(url);

    StringEntity entity = new StringEntity(body.toJSONString(),
StandardCharsets.UTF_8);
    httpPost.setEntity(entity);

    httpPost.setHeader("Content-type", "application/json");
    httpPost.setHeader("Authorization", "APPCODE " + appCode);

    try {
        response = httpClient.execute(httpPost);
        String responseBodyString = EntityUtils.toString(response.getEntity(),
StandardCharsets.UTF_8);
        if (!Objects.equals(response.getStatusLine().getStatusCode(), 200)) {
            System.out.println("本次请求返回状态码非200, 您可能需要在此处打印log");
        }
        return JSONObject.parseObject(responseBodyString);
    } catch (IOException ignored) {
        System.out.println("请求接口出现错误, 您可能需在此处打印log");
    }
}

```

```

    } finally {
        if (null != response) {
            try {
                EntityUtils.consume(response.getEntity());
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
    return null;
}
}

```

5、python代码示例

```

import requests

session = requests.Session()
adapter = requests.adapters.HTTPAdapter(
    pool_connections=100,
    pool_maxsize=300)

def post_once(url, app_code, body):
    headers = {
        'Content-Type': 'application/json',
        'Authorization': 'APPCODE ' + app_code
    }
    response = session.post(url=url, headers=headers, json=body)
    if response.status_code != 200:
        print('本次请求返回状态码非200,您可能需要在此处打印log')
    return response.json()

if __name__ == '__main__':
    # 接口的url、appCode、请求body待填写

    # TODO
    url = ''
    app_code = ''
    body = {

    }

    response_data = post_once(url, app_code, body)
    print(response_data)

```

