

代发系统



Yutaodaifa.com 制作

2021-07-07

| 日期 | 版本 | 说明 | 作者 |
|----------|--------|------|----|
| 2021/4/8 | V0.0.1 | 新增文档 | 张三 |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

目录

| | |
|-----------------------|----|
| 文档简介..... | 5 |
| 特别声明..... | 5 |
| 阅读对象..... | 5 |
| 产品说明..... | 5 |
| 名词解释..... | 5 |
| 接口工具测试..... | 5 |
| 使用流程..... | 6 |
| 全局参数列表..... | 6 |
| api 请求接口 | 6 |
| 寄发 api | 6 |
| 商品接口..... | 6 |
| 订单接口..... | 7 |
| 用户接口..... | 7 |
| 接口列表..... | 7 |
| api 请求接口 | 7 |
| 获取商品分类..... | 7 |
| 获取礼品分类..... | 8 |
| 获取快递价格..... | 9 |
| 获取所有仓库..... | 13 |
| 获取所有商品..... | 16 |
| 获取单个商品信息..... | 19 |
| 获取快递底单..... | 21 |
| 创建订单..... | 22 |
| 查看包裹信息..... | 26 |
| 查看订单信息..... | 28 |
| 获取用户信息..... | 30 |
| 寄发 api | 31 |

| | |
|------------------|----|
| 获取寄发商品信息..... | 31 |
| 新建接口..... | 31 |
| 生产环境资料..... | 32 |
| 如何获取服务器地址..... | 32 |
| 响应返回异常..... | 32 |
| 接入导入..... | 32 |
| 如何快速导入..... | 32 |
| 参数转码无效的解决方案..... | 33 |
| 附录..... | 34 |
| 返回码列表..... | 34 |

1 文档简介

1.1 特别声明

未得到本公司的书面许可，不得为任何目的、以任何形式或手段（包括但不限于机械的或电子的）复制或传播本文档的任何部分。对于本文档涉及的技术和产品，本公司拥有其专利（或正在申请专利）、商标、版权或其它知识产权。除非得到本公司的书面许可协议，本文档不授予这些专利、商标、版权或其它知识产权的许可。

本文档因产品功能示例和描述的需要，所使用的任何人名、企业名和数据都是虚构的，并仅限于本公司内部测试使用，不等于本公司有对任何第三方的承诺和宣传。

1.2 阅读对象

贵公司的技术部门的开发、维护及管理人员，应具备以下基本知识：

1. 了解 HTTPS/HTTP 协议等内容。
2. 了解信息安全的基本概念。
3. 了解计算机至少一种编程语言。

1.3 产品说明

本开发手册对该系统功能接口进行详细的描述，通过该指南可以对本系统有全面的了解，使技术人员尽快掌握本系统的接口，并能够在本系统上进行开发。

1.4 名词解释

| 名词缩写 | 名词定义 |
|------|------------------------|
| 客户端 | 本文档中的客户端为 ApiPost 客户端。 |
| 服务端 | 本文档中的服务端为用户端。 |
| 环境变量 | 本系统中动态的 KV 参数。 |

1.5 接口工具测试

1. 测试工具推荐 ApiPost 接口测试工具。官方地址：<https://www.apipost.cn>

2. ApiPost 测试工具使用方法如下：
 - ①. 将网关地址或者接口地址正确输入 url；
 - ②. 将所有参数按照下面目录和接口定义中参数从上到下顺序填写；
 - ③. 将目录的全局参数或者脚本正确写入；
 - ④. 将接口的参数和自定义脚本内容填入到对应位置；
3. 自定义脚本支持请求报文加密的数据拼装。

2 使用流程

1. 准备阶段：
 - A. 申请测试号等信息；
 - B. 取得开发手册（本文档）等资料；
2. 开发阶段：
 - A. 根据提供的 DEMO 结合开发文档快速熟悉对接接口；
 - B. 根据本系统提供的接口，在自己的系统上进行开发，实现所需要的业务功能；
 - C. 对自己系统的业务功能进行全面测试；
 - D. 与测试环境进行联调。
3. 生产使用：
 - A. 使用系统提供的正式资料。

3 全局参数列表

3.1 api 请求接口

目录描述：*暂无描述*

3.2 寄发 api

目录描述：*此 api 专门用于仓库寄发商品下单 如果您未在仓库寄发商品 无法使用此 api*

3.3 商品接口

目录描述：暂无描述

3.4 订单接口

目录描述：暂无描述

3.5 用户接口

目录描述：暂无描述

4 接口列表

4.1 api 请求接口

4.1.1 获取商品分类

接口地址：http://api.yutaodaifa.com/?action=get_product_category_lists

请求类型：*multipart/form-data*

请求方式：*post*

接口备注：此接口用于获取网站商品分类 例如/男装/女装/日用品 请根据自己的运营需求 网站结构 是否调用当前栏目

调试工具：*ApiPost*

URI 参数说明：

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------|----------------------------|--------|------|
| action | get_product_category_lists | String | 暂无描述 |

请求体参数说明：

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------------------|-----|----|------|
| 无请求参数 KEY/VALUE 类型 | | | |

返回参数说明：

| 参数名称 | 示例值 | 类型 | 参数说明 |
|------------|---|--------|-------|
| status | 1 | Number | 暂无描述 |
| msg | 获取成功 | String | 暂无描述 |
| data | | Object | 暂无描述 |
| data.id | 1 | Number | 分类 id |
| data.name | 母婴 | String | 分类名称 |
| data.image | https://daifa.oss-cn-shenzhen.aliyuncs.com/uploadfile/temp/20200910/1599728849NFU.png | String | 分类小图 |

返回示例:

```
{
  "status": 1,
  "msg": "获取成功",
  "data": [
    {
      "id": 1,
      "name": "母婴",
      "image": "https://daifa.oss-cn-shenzhen.aliyuncs.com/uploadfile/temp/20200910/1599728849NFU.png"
    }
  ]
}
```

4.1.2 获取礼品分类

接口地址: http://api.yutaodaifa.com/?action=get_product_type_lists

请求类型: *multipart/form-data*

请求方式: *post*

接口备注: 此接口用于获取网站的货品专区类型 例如 一毛专区 配重专区 促销专区 实际场景 建议根据网站自身情况选择是否调用当前栏目的商品 建议把商品获取回去之后入库自己创建的栏目当中

调试工具: *ApiPost*

URI 参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------|------------------------|--------|------|
| action | get_product_type_lists | String | 暂无描述 |

请求体参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------------------|-----|----|------|
| 无请求参数 KEY/VALUE 类型 | | | |

返回参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|------------------|----------|--------|-------|
| status | 1 | Number | 暂无描述 |
| msg | 获取成功 | String | 暂无描述 |
| data | | Object | 暂无描述 |
| data.id | 1 | Number | 栏目 id |
| data.name | 一毛专区 | String | 栏目名称 |
| data.description | 一毛低价礼品专区 | String | 栏目描述 |

返回示例:

```
{
  "status": 1,
  "msg": "获取成功",
  "data": [
    {
      "id": 1,
      "name": "一毛专区",
      "description": "一毛低价礼品专区"
    }
  ]
}
```

4.1.3 获取快递价格

接口地址: http://api.yutaodaifa.com/?action=get_express_info&depot_id=1

请求类型: *multipart/form-data*

请求方式: *post*

接口备注：此接口用于获取云仓所有支持的快递 以及当前用户组发快递的价格

调试工具：ApiPost

URI 参数说明：

| 参数名称 | 示例值 | 类型 | 参数说明 |
|----------|------------------|--------|----------------|
| action | get_express_info | String | 请求方法 |
| depot_id | 1 | String | 仓库 id, 默认为所有仓库 |

请求体参数说明：

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------------------|-----|----|------|
| 无请求参数 KEY/VALUE 类型 | | | |

返回参数说明：

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--|------|--------|------------------------|
| status | 1 | Number | 暂无描述 |
| msg | 获取成功 | String | 暂无描述 |
| data | | Object | 暂无描述 |
| data.express_price | | Object | 快递价格 |
| data.express_price.depot_id | 1 | Number | 仓库 id |
| data.express_price.depot_name | 成都仓储 | String | 仓库名称 |
| data.express_price.express | | Object | 支持快递 |
| data.express_price.express.express_id | 1 | Number | 快递 id |
| data.express_price.express.express_name | 申通快递 | String | 快递名称 |
| data.express_price.express.normal | | Object | 暂无描述 |
| data.express_price.express.normal.small | 3 | Number | 正常地区快递小件价格(0.5kg 以下) |
| data.express_price.express.normal.medium | 2 | Number | 正常地区快递中型件价格(0.5kg~1kg) |
| data.express_price.express.normal.large | 1 | Number | 正常地区快递大型件价格(超过 |

| | | | |
|---|--------|--------|-----------------------------|
| press.normal.large | | | 1kg 每按 kg 续重) |
| data.express_price.express.long | | Object | 暂无描述 |
| data.express_price.express.long.small | 3.5 | Number | 偏远地区快递小件价格(0.5kg 以下) |
| data.express_price.express.long.medium | 2.5 | Number | 偏远地区快递中型件价格(0.5kg~1kg) |
| data.express_price.express.long.large | 1.5 | Number | 偏远地区快递大型件价格(超过1kg 每按 kg 续重) |
| data.express_area | | Object | 地区 |
| data.express_area.id | 1 | Number | 快递 id |
| data.express_area.name | 申通快递 | String | 快递名称 |
| data.express_area.normal_area | | Object | 正常区域 |
| data.express_area.normal_area.code | 110000 | Number | 正常区域代码 |
| data.express_area.normal_area.name | 北京市 | String | 正常区域名称 |
| data.express_area.long_area | | Object | 偏远地区 |
| data.express_area.long_area.code | 540000 | Number | 偏远地区代码 |
| data.express_area.long_area.name | 西藏自治区 | String | 偏远地区名称 |
| data.express_area.disabled_area | | Object | 禁发地区 |
| data.express_area.disabled_area.province | | Object | 禁发省 |
| data.express_area.disabled_area.province.code | 220000 | Number | 禁发省代码 |
| data.express_area.disabled_area.province.name | 吉林省 | String | 禁发省名称 |
| data.express_area.disabled_area.reason | | Object | 禁发原因 |

返回示例:

```
{
  "status": 1,
```

```
"msg": "获取成功",
"data": {
  "express_price": [
    {
      "depot_id": 1,
      "depot_name": "成都仓储",
      "express": [
        {
          "express_id": 1,
          "express_name": "申通快递",
          "normal": {
            "small": "3",
            "medium": "2",
            "large": "1"
          },
          "long": {
            "small": "3.5",
            "medium": "2.5",
            "large": "1.5"
          }
        }
      ]
    }
  ],
  "express_area": [
    {
      "id": 1,
      "name": "申通快递",
      "normal_area": [
        {
          "code": "110000",
          "name": "北京市"
        }
      ],
      "long_area": [
        {
          "code": "540000",
          "name": "西藏自治区"
        }
      ],
      "disabled_area": [
        {
          "province": {
            "code": "220000",
            "name": "吉林省"
          }
        }
      ]
    }
  ]
}
```

```

        "reason": ""
      }
    ]
  }
}

```

4.1.4 获取所有仓库

接口地址: http://api.yutaodaifa.com/?action=get_depot_lists

请求类型: *multipart/form-data*

请求方式: *post*

接口备注: *接口描述*

调试工具: *ApiPost*

URI 参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------|-----------------|--------|------|
| action | get_depot_lists | String | 暂无描述 |

请求体参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------------------|-----|----|------|
| 无请求参数 KEY/VALUE 类型 | | | |

返回参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|------------------|-----------|--------|-------|
| status | 1 | Number | 暂无描述 |
| msg | 获取成功 | String | 暂无描述 |
| data | | Object | 暂无描述 |
| data.id | 1 | Number | 仓库 id |
| data.name | 成都仓储 | String | 仓库名称 |
| data.description | 坐落于成都武侯区 | String | 仓库介绍 |
| data.address | 四川省成都市青羊区 | String | 仓库地址 |

| | | | |
|----------------------------|------|--------|-------------------------|
| data.default_express | | Object | 默认快递 |
| data.default_express.id | 1 | Number | 默认快递 id |
| data.default_express.name | 申通快递 | String | 默认快递名称 |
| data.express | | Object | 支持快递信息 |
| data.express.id | 1 | Number | 支持快递 id |
| data.express.name | 申通快递 | String | 支持快递名称 |
| data.is_express_limit | 1 | Number | 是否限制单包裹商品种类 1 限制 0 不限制 |
| data.express_limit | 2 | Number | 单包裹商品种类限制数量 |
| data.is_single_goods_limit | 1 | Number | 是否限制单商品可添加数量 1 限制 0 不限制 |
| data.single_goods_add_num | 10 | Number | 单商品可添加数量 |

返回示例:

```
{
  "status": 1,
  "msg": "获取成功",
  "data": [
    {
      "id": 1,
      "name": "成都仓储",
      "description": "坐落于成都武侯区",
      "address": "四川省成都市青羊区",
      "default_express": {
        "id": 1,
        "name": "申通快递"
      },
      "express": [
        {
          "id": "1",
          "name": "申通快递"
        }
      ],
      "is_express_limit": 1,
      "express_limit": 2,
      "is_single_goods_limit": 1,
      "single_goods_add_num": 10
    }
  ]
}
```

```
    }  
  ]  
}  
原文返回数组示例:  
Array  
(  
  [status] => 1  
  [msg] => 获取成功  
  [data] => Array  
    (  
      [0] => Array  
        (  
          [id] => 1  
          [name] => 成都仓储  
          [description] => 坐落于成都武侯区  
          [address] => 四川省成都市青羊区  
          [default_express] => Array  
            (  
              [id] => 1  
              [name] => 申通快递  
            )  
          [express] => Array  
            (  
              [0] => Array  
                (  
                  [id] => 1  
                  [name] => 申通快递  
                  [logo] => /uploadfile/temp/20200716/1594889671Wr3.png  
                )  
              [1] => Array  
                (  
                  [id] => 2  
                  [name] => 圆通快递  
                  [logo] => /uploadfile/temp/20200716/1594889700ooN.jpg  
                )  
              [2] => Array  
                (  
                  [id] => 6  
                  [name] => 韵达  
                  [logo] => /uploadfile/temp/20200716/1594889815UhX.png  
                )  
            )  
          [sources] => Array  
            (  
              [0] => 1 //支持淘宝  
              [1] => 3 //支持京东面单  
            )  
        )  
      )  
    )  
)
```

```

[2] => 4 //支持拼多多面单

)
[is_express_limit] => 1
[express_limit] => 1
[is_single_goods_limit] => 1
[single_goods_add_num] => 1
)
)

```

4.1.5 获取所有商品

接口地址:

http://api.yutaodaifa.com/?action=get_product_lists&product_type_id=&depot_id=&catid=&express_id=&order=

请求类型: *multipart/form-data*

请求方式: *post*

接口备注: 此接口用于获取云仓所有在线商品列表 2020-01-09 新增 *end_lists* 数组返回已下架的云仓商品 ID 如果已入库该商品 可以使用此接口下架对应商品

调试工具: *ApiPost*

URI 参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|-----------------|-------------------|--------|--|
| action | get_product_lists | String | 暂无描述 |
| product_type_id | | Number | 根据活动类型筛选, 活动类型 ID 自行通过接口获取 |
| depot_id | | Number | 根据仓库进行筛选, 仓库 ID 自行通过接口获取 |
| catid | | Number | 根据所属栏目筛选, 栏目信息通过接口获取 |
| express_id | | Number | 根据快递进行筛选, 快递 ID 通过接口获取 |
| order | | String | 排序方式 id:按照最新排序,sales_asc:销量从低到高,sales_desc:销量从高到低,price_asc:价格从低到高,price_desc:价格从高到低,weight_desc:重量从重到 |

| | | | |
|------|--|--------|---------------------|
| | | | 轻,weight_asc:重量从轻到重 |
| page | | String | 当前页码 默认第一页 |
| num | | String | 每页数量 默认 15 条 最大 50 |

请求体参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------------------|-----|----|------|
| 无请求参数 KEY/VALUE 类型 | | | |

返回参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|----------------------------------|---|--------|--------|
| status | 1 | Number | 暂无描述 |
| msg | 获取成功 | String | 暂无描述 |
| data | | Object | 暂无描述 |
| data.all_page | 3 | Number | 总页数 |
| data.page | 1 | Number | 当前页数 |
| data.lists | | Object | 暂无描述 |
| data.lists.id | 42 | Number | 产品 id |
| data.lists.title | USB 节能灯 | String | 产品名称 |
| data.lists.thumb | https://daifa.oss-cn-shenzhen.aliyuncs.com/uploadfile/temp/20201015/16027534230wb.jpeg | String | 产品缩略图 |
| data.lists.price | 0.42 | Number | 价格 |
| data.lists.weight | 0.03kg | Number | 重量 |
| data.lists.depots | | Object | 仓库信息 |
| data.lists.depots.name | 成都仓储 | String | 仓库名称 |
| data.lists.depots.depot_id | 1 | Number | 仓库 id |
| data.lists.depots.surplus_number | 311 | Number | 剩余数量 |
| data.lists.depots.express | | Object | 仓库快递信息 |

| | | | |
|--------------------------------------|---|--------|----------|
| data.lists.depots.express.name | 申通快递 | String | 快递名称 |
| data.lists.depots.express.logo | /uploadfile/temp/20200716/1594889671Wr3.png | String | 快递 logo |
| data.lists.depots.express.express_id | 1 | Number | 快递 id |
| data.lists.is_cat | 0 | Number | 是否存在于购物车 |

返回示例:

```
{
  "status": 1,
  "msg": "获取成功",
  "data": {
    "all_page": 3,
    "page": 1,
    "lists": [
      {
        "id": 42,
        "title": "USB 节能灯",
        "thumb": "https://daifa.oss-cn-shenzhen.aliyuncs.com/uploadfile/temp/20201015/1602753423Owb.jpeg",
        "price": "0.42",
        "weight": "0.03kg",
        "depots": [
          {
            "name": "成都仓储",
            "depot_id": "1",
            "surplus_number": 311,
            "express": [
              {
                "name": "申通快递",
                "logo": "/uploadfile/temp/20200716/1594889671Wr3.png",
                "express_id": "1"
              }
            ]
          }
        ],
        "is_cat": 0
      }
    ]
  }
}
```

4.1.6 获取单个商品信息

接口地址: `http://api.yutaodaifa.com/?action=get_product_info&product_id=42`

请求类型: `multipart/form-data`

请求方式: `post`

接口备注: `接口描述`

调试工具: `ApiPost`

URI 参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|------------|------------------|--------|-------|
| action | get_product_info | String | 暂无描述 |
| product_id | 42 | Number | 活动 ID |

请求体参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------------------|-----|----|------|
| 无请求参数 KEY/VALUE 类型 | | | |

返回参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------------------|---|--------|-------|
| status | 1 | Number | 暂无描述 |
| msg | 获取成功 | String | 暂无描述 |
| data | | Object | 暂无描述 |
| data.id | 42 | Number | 商品 ID |
| data.thumb | https://daifa.oss-cn-shenzhen.aliyuncs.com/uploadfile/tem p/20201015/1602753423Owb.jpeg | String | 商品大图 |
| data.more_thumb | | Object | 更多图片 |
| data.goods_content | <p></p> | | |
| data.catid | 156 | Number | 所属分类 |
| data.price | 0.42 | Number | 礼品价 |
| data.goods_price | 0.60 | Number | 商品原价 |
| data.status | 1 | Number | 商品状态 1: 进行中, 2: 已下架, 3: 等待上架 |
| data.weight | 0.03 | Number | 重量 |
| data.product_type_id | 5 | Number | 所属活动栏目 |
| data.total_number | 7431 | Number | 总库存 |
| data.short_name | 节能灯 | String | 简称 |
| data.depots | | Object | 仓库信息 |
| data.depots.name | 成都仓储 | String | 仓库名称 |
| data.depots.depot_id | 1 | Number | 仓库 ID |
| data.depots.surplus_number | 311 | Number | 库存 |
| data.depots.express | | Object | 快递信息 |
| data.depots.express.name | 申通快递 | String | 快递名称 |
| data.depots.express.logo | /uploadfile/temp/20200716/1594889671Wr3.png | String | 快递 logo |
| data.depots.express.express_id | 1 | Number | 快递 id |

返回示例:

```
{
  "status": 1,
  "msg": "获取成功",
  "data": {
    "id": 42,
    "thumb": "https://daifa.oss-cn-shenzhen.aliyuncs.com/uploadfile/temp/20201015/16027534230wb.jpeg",
    "more_thumb": null,
    "goods_content": "",
    "catid": 156,
```

```

    "price": "0.42",
    "goods_price": "0.60",
    "status": 1,
    "weight": 0.03,
    "product_type_id": 5,
    "total_number": 7431,
    "short_name": "节能灯",
    "depots": [
        {
            "name": "成都仓储",
            "depot_id": "1",
            "surplus_number": 311,
            "express": [
                {
                    "name": "申通快递",
                    "logo":
"/uploadfile/temp/20200716/1594889671Wr3.png",
                    "express_id": "1"
                }
            ]
        }
    ]
}

```

4.1.7 获取快递底单

接口地址: http://api.yutaodaifa.com/?action=get_express_bills&yun_express_sn=

请求类型: *multipart/form-data*

请求方式: *post*

接口备注: 接口是用于获取快递电子面单底单信息 实时返回底单照片 请传入本平台返回的正确快递单号 目前仅支持 明文类型的订单 获取底单 暂不支持: 通过店铺同步订单形式获取过来的密文订单 获取快递底单

调试工具: *ApiPost*

URI 参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|----------------|-------------------|--------|------------|
| action | get_express_bills | String | 请求方法 |
| yun_express_sn | | String | 由云仓返回的快递单号 |

请求体参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------------------|-----|----|------|
| 无请求参数 KEY/VALUE 类型 | | | |

返回参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------------------|-----|----|------|
| 无请求参数 KEY/VALUE 类型 | | | |

返回示例:

```
//成功之后返回示例
Array
(
    [status] => 1
    [msg] => 底单获取成功
    [data] => http://upload.kuaidihelp.com/CloudPrint/2020/12/15/17/06/24390966703232417489954926555020.jpg
)
```

4.1.8 创建订单

接口地址: http://api.yutaodaifa.com/?action=submit_order_trade

请求类型: *multipart/form-data*

请求方式: *post*

接口备注: 此接口用于向云仓提交订单数据 请严格按照示例传入数据 订单创建成功之后 可在商家中心看见已创建的订单信息 然后使用查看订单信息接口 获取订单的出单状态和单号

调试工具: *ApiPost*

URI 参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------|--------------------|--------|------|
| action | submit_order_trade | String | 暂无描述 |

请求体参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|----------------|-----------|--------|------------------------------------|
| appkey | | String | 申请的 api 密钥 |
| product_lists | | Array | 需下单的商品列表 请按照示例传入 |
| order_lists | | Array | 需下单的包裹列表 请按照示例传入 最大 1000 条 |
| yun_depot_id | 1 | Number | 云端仓库 ID 可以用仓库 api 获取 |
| yun_express_id | 1 | Number | 云端快递 ID 可以用快递 api 获取 |
| source | 1 | Number | 1 淘宝/天猫 3.京东 4 拼多多 5 抖音 6 快手 10.其他 |
| trade_sn | 123456789 | Number | 本地唯一订单号 为避免下单重复 每次下单请传入您本地唯一订单编号 |
| is_check | 0 | String | 1 为提交测试 0 为正式下单 |

返回参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------------------|-----|----|------|
| 无请求参数 KEY/VALUE 类型 | | | |

返回示例:

json 请求示例:

```
{
  "action": "submit_order_trade",
  "product_lists": [{
    "num": 1,
    "yun_product_id": 1
  }],
  "order_lists": [{
    "address": "61017808637494,雪**,133****726,重庆市渝北区**两江新区阿里云创新中心**",
    "order_sn": 61017808637494,
    "encrypt_datas": {
      "name": "~AgAAAAAqcX8FqYksbQAEQMZY46g08wk0mF5modUbN8=~kNen~2~~",
      "phone": "$jmGuiNewL0jy$AgAAAAAqcX8GqYksbQC5oQipz2HnRYzEKlz6Z3fCE8s=$2$$",
      "address":
      "~AgAAAAAqcX8lqYksbQCWsaZR/OVtXOM4OQFOHQslQ1rfEszHVZRuovlyoGGFvL9A~E/cW4TfJqXl7Cxo+IKX/bh01rDlaqyBLolEw~2~~"
    }
  }],
}
```

```

    "is_check": 0,
    "yun_depot_id": 1,
    "yun_express_id": 6,
    "source": 4,
    "appkey": "m43IPgnV13QW44zWqfPf3QaFqg115555",
    "trade_sn": "61017808637494001"
  }
}
请求说明 Array
(
  [action] => submit_order_trade
  [product_lists] => Array
  (
    [0] => Array
    (
      [yun_product_id] => 25 //云仓商品 ID
      [num] => 1 //商品数量
    )
    //如果云仓支持同一包裹多件商品同时下单 就传入多件
    [1] => Array
    (
      [yun_product_id] => 26 //云仓商品 ID
      [num] => 1 //商品数量
    )
  )

  [order_lists] => Array
  (
    [0] => Array
    (
      [order_sn] => //如果是检测订单可不传 正式提交订单请传入本地网站的唯一订单包裹号 后期可用此包裹号查询包裹状态
      [address] => 120552000000000000,范先生,13500000000,浙江省杭州市黄山大道 1 号 //格式为 第三方订单号,收件人姓名,收件人手机号,详细收件地址 第三方订单号如果没有可以随意传入
      [oaid] => //可选 如果是淘宝密文订单 请必须传入 oaid

      //如果是拼多多密文订单且订购服务商不一致 请必须传入以下全加密数据
      [encrypt_datas] =>array(

        [name] => ~AgAAAAAqcX8FqYksbQAEQMZY46g08wk0mFSmodUbN8=~kNen~2~~ //收件人姓名 全加密格式
        [phone] => $jmGuiNewL0jy$AgAAAAAqcX8GqYksbQC5oQipz2HnRYzEKlz6Z3fCE8s=$2$$ // 收件人姓名 全加密格式
        [address] =>

```



```
~AgAAAAqcX8lqYksbQCWsaZR/OVtXOM4OQFOHQsiQ1rfEsZHVZRuovlyoGGFvL9A~E/cW4TfjqXI7Cxo+IKX/bh01rDlaqy
BLolEw~2~~ //收件人地址 全加密格式
```

```

    )
  }

  )
  [1] => Array
  (
    [order_sn] =>
    [address] => 120624000000000000,谭先生,13300000000,浙江省杭州市黄山大道 2 号
    [oid] => //可选 如果是淘宝订单 请传入 oaid
  )
)
[is_check] => 1
[yun_depot_id] => 2
[yun_express_id] => 1
[source] => 1
[appkey] => FUphl8z8Pvkvv5Kw6pNWKH8PqPpp8qu5
[trade_sn] => 120116068172436935 //本地网站的唯一订单标识 后续可通过此订单编号 查看订单信息
)

```

响应数据:

如果 检测提交订单返回数据:

```
(
  [status] => 1
  [msg] => 检测通过
  [data] => Array
  (
  )
)
```

如果是创建订单则返回以下数据

```
(
  [status] => 1
  [msg] => 订单创建成功 订单 ID:1203160698259048
  [data] => Array
  (
    [yun_order_sn] => 1203160698259048 //云端订单唯一的 order_sn 请保存后续通过此获取订单状态
    [total_order_pay_price] => 6.14 //本次总支付费用
    [status] => 2 //订单给状态 1 待提交 2 已付款待出单 3.已全部出单 待确认
    [order_lists] => Array //订单列表
    (
      [0] => Array
      (
        [status] => 1 //包裹状态 1 待出单 2 已出单 3 售后中 4.已重发 5.售后被驳回 6.已完成 -1 已退款
        [order_sn] => 12031606982590480 //云端包裹唯一号 后续可通过此包裹号 获取某个包裹状态
      )
    )
  )
)
```

```

[express_sn] => //快递单号 获取成功才有 提交订单号 一般 30 分钟内全部出单
[express_id] => 1 //快递 id 可以使用获取快递列表查看
[waybill_status] => 1 //出单状态 1 待出单 2 已出单 -1 出单失败
[api_order_sn] => 12051607158232250 //通过 api 传入的本地网站包裹唯一号 后期可通过此 api 包裹号 获取某个包裹状态
)
[1] => Array
(
    [status] => 1
    [order_sn] => 12031606982590481
    [express_sn] =>
    [express_id] => 1
    [waybill_status] => 1
)
)
)
)
)

```

4.1.9 查看包裹信息

接口地址: http://api.yutaodaifa.com/?action=get_order_info

请求类型: *multipart/form-data*

请求方式: *post*

接口备注: 此接口是用于 如果要获取订单下某一个包裹信息 可以用此接口 返回包裹信息和状态

调试工具: *ApiPost*

URI 参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------------|-----------------------|--------|---|
| action | get_order_info | String | 暂无描述 |
| yun_order_sn | 120616072545966 60 | String | 云端返回的包裹唯一编号 |
| api_order_sn | | String | 本地网站传入的包裹唯一编号 yun_order_sn 或 api_order_sn 必须传入其中一个 |

请求体参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|------|-----|----|------|
|------|-----|----|------|

无请求参数 KEY/VALUE 类型

返回参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|---------------------------------|---|--------|------------------|
| status | 1 | Number | 暂无描述 |
| msg | 包裹信息获取成功 | String | 暂无描述 |
| data | | Object | 暂无描述 |
| data.status | -1 | Number | 包裹状态 |
| data.order_sn | 120616072545966 60 | Number | 云仓包裹唯一编号 |
| data.express_sn | | Object | 快递单号 |
| data.express_id | 1 | Number | 快递 ID |
| data.waybill_status | 1 | Number | 出单状态 |
| data.api_order_sn | 120616072545768 30 | Number | 本地网站传入的包裹编号 |
| data.refund_cause | 不支持 jd | String | 退款原因 如果订单发生退款才显示 |
| data.status_name | 已退款 | String | 包裹状态 |
| data.express_name | 申通快递 | String | 快递名称 |
| data.order_log_lists | | Object | 订单日志 |
| data.order_log_lists.cause | 订单管理员已手动退款 退款原因: 不支持 jd, 退款金额: 2.42 元到您账户 | String | 暂无描述 |
| data.order_log_lists.input_time | 1607254640 | Number | 暂无描述 |

返回示例:

```
{
  "status": 1,
  "msg": "包裹信息获取成功",
  "data": {
    "status": -1,
    "order_sn": "12061607254596660",
    "express_sn": null,
  }
}
```

```

    "express_id": 1,
    "waybill_status": 1,
    "api_order_sn": "12061607254576830",
    "refund_cause": "不支持 jd",
    "status_name": "已退款",
    "express_name": "申通快递",
    "order_log_lists": [
      {
        "cause": "订单管理员已手动退款 退款原因: 不支持 jd, 退款金额: 2.42 元
到您账户",
        "input_time": 1607254640
      }
    ]
  }
}

```

4.1.10 查看订单信息

接口地址: http://api.yutaodaifa.com/?action=get_order_trade_info

请求类型: *multipart/form-data*

请求方式: *post*

接口备注: 获取订单信息接口 创建订单之后,间隔一段时间之后 可以使用此接口获取订单出单情况和出单状态 集成建议: 当订单状态 `status = 3` 的时候为全部出单 此时在对本地订单进行更新操作 订单下的部分包裹 可能会因为地址不全和快递禁发 或者其他原因 到导致无法获取单号 发生退款 请集成的时候特别注意: 参考以下包裹状态:

1 => '已付款 待出单', 2 => '已出单 待确认', 3 => '售后中', 4 => '已重发',
5 => '已驳回', 6 => '已完成', 0 => '已取消', -1 => '已退款',

调试工具: *ApiPost*

URI 参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------------|----------------------|--------|----------------------------------|
| action | get_order_trade_info | String | 暂无描述 |
| yun_trade_sn | | String | 云仓的订单编号 |
| api_trade_sn | | String | 本地网站的订单编号 云仓本地编号/本地订单编号 必须传入其中一个 |

请求体参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------------------|-----|----|------|
| 无请求参数 KEY/VALUE 类型 | | | |

返回参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------------------|-----|----|------|
| 无请求参数 KEY/VALUE 类型 | | | |

返回示例:

请求信息示例;

Array

```
(
  [action] => get_order_trade_info
  [appkey] => FUphl8z8Pvkvv5Kw6pNWkh8PqPpp8qu5
  [yun_trade_sn] => 1204160708593926 //云仓订单编号
  [api_trade_sn] => //本地订单编号 云仓订单编号和本地订单编号 必须传入其中一个
)
```

//成功之后响应信息

Array

```
(
  [status] => 1
  [msg] => 订单信息获取成功
  [data] => Array
  (
    [api_trade_sn] => 1204160708589772
    [yun_trade_sn] => 1204160708593926
    [status] => 2
    [status_name] => 待出单
    [order_lists] => Array
    (
      [0] => Array
      (
        [status] => 1
        [order_sn] => 12041607085939260
        [express_sn] => //快递单号 出单之后才会显示
        [express_id] => 1 //快递 ID
        [waybill_status] => 1 //出单状态
        [api_order_sn] =>
        [status_name] => 已付款 待出单 //包裹状态
        [express_name] => 申通快递
      )
    )
  )
)
```

```

[1] => Array
(
    [status] => 1
    [order_sn] => 12041607085939261
    [express_sn] =>
    [express_id] => 1
    [waybill_status] => 1
    [api_order_sn] =>
    [status_name] => 已付款 待出单
    [express_name] => 申通快递
)
)
)
)

```

4.1.11 获取用户信息

接口地址: http://api.yutaodaifa.com/?action=get_user_info

请求类型: *multipart/form-data*

请求方式: *post*

接口备注: 此接口用于获取 *api* 用户在网站的账户财务信息

调试工具: *ApiPost*

URI 参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------|---------------|--------|------|
| action | get_user_info | String | 请求方法 |

请求体参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------------------|-----|----|------|
| 无请求参数 KEY/VALUE 类型 | | | |

返回参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------|------|--------|------|
| status | 1 | Number | 暂无描述 |
| msg | 获取成功 | String | 暂无描述 |
| data | | Object | 暂无描述 |

| | | | |
|------------|-------------|--------|---------|
| data.phone | 155*****315 | Number | 所属用户手机号 |
| data.money | 716.80 | Number | 用户账户余额 |
| data.gold | 0.00 | Number | 用户账户金币 |

返回示例:

```
{
  "status": 1,
  "msg": "获取成功",
  "data": {
    "phone": "155*****315",
    "money": "716.80",
    "gold": "0.00"
  }
}
```

4.2 寄发 api

4.2.1 获取寄发商品信息

接口地址:

请求类型: *multipart/form-data*

请求方式: *post*

接口备注: *接口描述*

调试工具: *ApiPost*

请求体参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------------------|-----|----|------|
| 无请求参数 KEY/VALUE 类型 | | | |

返回参数说明:

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------------------|-----|----|------|
| 无请求参数 KEY/VALUE 类型 | | | |

4.2.2 新建接口

接口地址：*寄发商品下单*

请求类型：*multipart/form-data*

请求方式：*post*

接口备注：*接口描述*

调试工具：*ApiPost*

请求体参数说明：

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------------------|-----|----|------|
| 无请求参数 KEY/VALUE 类型 | | | |

返回参数说明：

| 参数名称 | 示例值 | 类型 | 参数说明 |
|--------------------|-----|----|------|
| 无请求参数 KEY/VALUE 类型 | | | |

5 生产环境资料

5.1 如何获取服务器地址

以阿里云为例，登录阿里云后台找到 ECS 控制台，找到公网地址，复制保存即可。

5.2 响应返回异常

对比 Postman 和 ApiPost 实际发出的请求报文（curl、参数、body、header）是否完全吻合。

（关于密码的保管：贵公司一定要保证密码仅能被少数可靠的授权人知晓，严防密码被不可信的人获取，如密码泄露需立即进行修改同时替换程序中的密码。）

6 接入导入

6.1 如何快速导入

- ①. 获取 DEMO 压缩包：

- ②. 将请求地址改为测试环境请求地址；
- ③. 目前仅有 PHP 范例；
- ④. 将范例部署到您的应用服务器，并运行；
- ⑤. 在测试环境上请调通接口；
- ⑥. 请求和响应都调通后，便可在范例中加入您系统本身的业务逻辑，并再次在测试环境进行调试，直至通过；
- ⑦. 将您正式的密钥配置到程序中，并将请求地址改为正式环境请求地址后便可上线。

6.2 参数转码无效的解决方案

解决方法：

首先检查配置中的参数是否与系统上提供的参数一致,是否有中文。

其次处理中文转码问题,有两个需要正确转码的环节：

- ①. 涉及中文的参数在传入生成 url 的方法时，注意编码问题。
- ②. 生成请求参数后，涉及中文的参数的值不能是乱码，当前仅支持 UTF-8 的编码格式。

7 附录

7.1 返回码列表

| 返回码 | 说明 |
|--------------|--|
| SYS_API_0000 | 成功! |
| SYS_API_999 | 注册参数校验失败/抱歉,白名单参数有误! / 抱歉,冻结天数超出范围! / 传入的参数有误! |
| SYS_API_998 | 调用接口失败。 |
| SYS_API_997 | 附件大小不符合规范。 |
| | |
| | |
| | |
| | |