

# API 使用说明

Version 1.6 (2018-05-02)

**深圳市快飞鱼供应链有限公司**

地 址：深圳市龙岗区坂田街道杨美村杨美路27号1楼

邮 编：518000

网 站：<http://www.iquickfish.com>

APIUrl：<http://api.iquickfish.com>

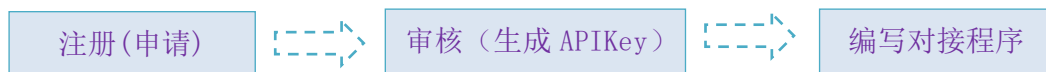
## 目录

1.	API 接口概要	- 1 -
2.	API 接口对接流程	- 1 -
2.1.	注册账号及 APIKey 申请	- 1 -
2.2.	审核通过	- 1 -
2.3.	编写程序对接	- 1 -
3.	API 接口函数说明	- 1 -
3.1.	API 身份验证规则	- 1 -
3.2.	运输区域查询[国家大全] (Country)	- 2 -
3.2.1.	获取所有国家信息函数(GetAll)	- 2 -
3.2.2.	根据 Id 获取国家信息函数(GetById)	- 3 -
3.2.3.	根据中文名称获取国家信息函数(GetByCnName)	- 3 -
3.2.4.	根据英文名称获取国家信息函数(GetByEnName)	- 4 -
3.2.5.	根据英文别名获取国家信息函数(GetByEnAlias)	- 4 -
3.3.	运输渠道查询 (Channel)	- 5 -
3.3.1.	获取所有运输方式信息函数(GetAll)	- 6 -
3.3.2.	根据 Id 获取运输方式信息函数(GetById)	- 6 -
3.3.3.	根据类别 Id 获取运输方式信息函数(GetByClassId)	- 6 -
3.3.4.	根据中文名称获取运输方式信息函数(GetByCnName)	- 7 -
3.3.5.	根据英文名称获取运输方式信息函数(GetByEnName)	- 7 -
3.3.6.	根据国家 Id 获取运输方式信息函数(GetByCountryId)	- 7 -
3.4.	运输方式类别查询 (ChannelClass)	- 8 -
3.4.1.	获取所有运输方式类别信息函数(GetAll)	- 8 -
3.4.2.	根据 Id 获取运输方式类别信息函数(GetById)	- 8 -
3.5.	订单信息查询及增加 (Order)	- 9 -
3.5.1.	根据订单参考号获取订单信息函数 (GetByReferCode)	- 12 -
3.5.2.	根据订单号获取订单信息函数(GetByOrderCode)	- 13 -
3.5.3.	增加订单信息函数(Create)	- 13 -
3.5.4.	修改订单信息函数(Update)	- 15 -
3.5.5.	修改订单预报重量函数(UpdateWeight)	- 16 -
3.5.6.	删除订单函数(DeleteById)	- 16 -
3.6.	跟踪号查询 (Order)	- 18 -
3.6.1.	GetTrackNoByReferCode	- 18 -
3.6.2.	GetTrackNoByOrderCode	- 18 -
4.	Label (标签) 获取	- 18 -
5.	跟踪轨迹	- 20 -
5.1.1.	单条订单跟踪信息查询	- 20 -
6.	API 接口使用建议	- 21 -
6.1.	使用本地程序缓存	- 21 -
6.2.	异常处理	- 21 -
6.3.	API 测试数据	- 21 -
6.4.	时间差	- 21 -

## 1. API 接口概要

API 使用 HTTP 访问方式。HTTP 访问方式是指用户通过 URL (GET) 或提交表单 (POST) 方式来获取系统信息。我们系统接收到用户按一定规则提交的请求后, 将会以 JSON 的形式返回相关信息。

## 2. API 接口对接流程



### 2.1. 注册账号及 APIKey 申请

1. 登录官网进行注册账号;
  2. 提交身份资料验证;
  3. APIKey 对接申请;
- 或**  
注册后直接联系您的业务经理

### 2.2. 审核通过

1. 我们收到您的申请后, 进行资料的审核;
2. 审核通过后, 根据您的用户账号生成一个 APIKey, 并交付给您;

### 2.3. 编写程序对接

1. 进行 API 接口调用;

## 3. API 接口函数说明

API 接口是物流运输订单数据的外部查询接口, 目前提供以下功能:

1. 运输区域查询;
2. 运输方式及类别查询;
3. 增加, 修改, 查询订单;

### 3.1. API 身份验证规则

调用 API 接口时, 涉及数据创建, 修改的 API, 需要进行身份验证:

签名规则:

1. `userStr = UserId + "_" + 当前时间(yyyyMMddHHmmss)` ; //
2. `signStr = SAFSign.Sign(userStr, apiKey, "utf-8")` ;

C# 请求验证代码Demo:

```

string apiKey = "1b35e1dd-4276-49c2-8c01-2cc5d0dd1ddf"; //提供给用户的密钥
string userStr = "10000" + "_" + DateTime.Now.ToString("yyyyMMddHHmmss"); //UserId + "_" + 14码时间 (yyyyMMddHHmmss)
string signStr = SAFSign.Sign(userStr, apiKey, "utf-8"); //utf-8编码进行签名
  
```

```

HttpClient client = new HttpClient();
client.BaseAddress = new Uri(APIUrl);
//设置Http请求验证信息
client.DefaultRequestHeaders.Add("kfy_userid", userStr);
client.DefaultRequestHeaders.Add("kfy_signdata", signStr);

```

### 3.2. 运输区域查询[国家大全] (Country)

函数表:

NO#	函数名称	接口分类	访问路径	方式	功能说明
1	GetAll()	Country	/V1/Country/GetAll	Get	获取所有国家信息
2	GetById(int id)	Country	/V1/Country/GetById?id=???	Get	根据 ID 获取国家信息
3	GetByCnName(string name)	Country	/V1/Country/GetByCnName?name=???	Get	根据中文名称获取国家信息
4	GetByEnName(string name)	Country	/V1/Country/GetByEnName?name=???	Get	根据英文名称获取国家信息
5	GetByEnAlias(string name)	Country	/V1/Country/GetByEnAlias?name=???	Get	根据英文别名获取国家信息 <b>注:</b>

Json 结构:

```

{"CountryId":1,"CountryCode":"US","CnName":"美国","CnAlias":"美國","EnName":"United States Of America","EnAlias":"United States Of America,USA,US,United States,USA,US"}

```

No#	属性名称	属性含义	类型	默认值	特别说明
1	CountryId	国家 Id	整型		
2	CountryCode	国家代码	字符		
3	CnName	中文名称	字符		
4	CnAlias	中文别名	字符		以半角逗号分隔, 一个国家可能有多个中文别名
5	EnName	英文名称	字符		
6	EnAlias	英文别名	字符		以半角逗号分隔, 一个国家可能有多个英文别名

#### 3.2.1. 获取所有国家信息函数(GetAll)

GetAll

函数名称	GetAll()
访问路径	/V1/Country/GetAll
验证需要	否
功能说明	获取所有国家信息
接口分类	Country
访问方式	Get
返回类型	Json 格式字符串 (Country 集合)

C#样例	<pre> HttpClient client = new HttpClient(); client.BaseAddress = new Uri(APIUrl); HttpResponseMessage response = client.GetAsync("V1/Country/GetAll").Result; if (response.IsSuccessStatusCode) {     var jsonStr = response.Content.ReadAsStringAsync().Result;     var list = JsonSerializer.Deserialize&lt;IEnumerable&lt;Country&gt;&gt;(jsonStr);     Console.WriteLine("{0} ({1})", (int)response.StatusCode, response.ReasonPhrase); } else {     Console.WriteLine("{0} ({1})", (int)response.StatusCode, response.ReasonPhrase); } </pre>
使用建议	定期获取一次（例如每天），缓存于调用程序，以提高响应性能。

### 3.2.2. 根据 Id 获取国家信息函数 (GetById)

GetById(**int** id)

函数名称	GetById( <b>int</b> id)
访问路径	/V1/Country/GetById?id=???
验证需要	否
功能说明	根据国家 Id 获取国家信息
接口分类	Country
访问方式	Get
参数类型	整型
返回类型	Json 格式字符串（Country 对象）
C#样例	<pre> HttpClient client = new HttpClient(); client.BaseAddress = new Uri(APIUrl); HttpResponseMessage response = client.GetAsync("V1/Country/GetById?id=3").Result; if (response.IsSuccessStatusCode) {     var jsonStr = response.Content.ReadAsStringAsync().Result;     var obj = JsonSerializer.Deserialize&lt;Country&gt;(jsonStr);     Console.WriteLine("{0} ({1})", (int)response.StatusCode, response.ReasonPhrase); } else {     Console.WriteLine("{0} ({1})", (int)response.StatusCode, response.ReasonPhrase); } </pre>

### 3.2.3. 根据中文名称获取国家信息函数 (GetByCnName)

GetByCnName(**string** name)

函数名称	GetByCnName( <b>string</b> name)
访问路径	/V1/Country/GetByCnName?name=???
验证需要	否
功能说明	根据中文名称获取国家信息

接口分类	Country
访问方式	Get
参数类型	字符型
返回类型	Json 格式字符串 (Country 对象)
C#样例	<pre> HttpClient client = new HttpClient(); client.BaseAddress = new Uri(APIUrl); HttpResponseMessage response = client.GetAsync("/V1/Country/GetByCnName?name=美国").Result; if (response.IsSuccessStatusCode) {     var jsonStr = response.Content.ReadAsStringAsync().Result;     var obj = JsonSerializer.Deserialize&lt;Country&gt;(jsonStr);     Console.WriteLine("{0} ({1})", (int)response.StatusCode, response.ReasonPhrase); } else {     Console.WriteLine("{0} ({1})", (int)response.StatusCode, response.ReasonPhrase); } </pre>

### 3. 2. 4. 根据英文名称获取国家信息函数 (GetByEnName)

GetByEnName (string name)

函数名称	GetByCnName (string name)
访问路径	/V1/Country/GetByEnName?name=???
验证需要	否
功能说明	根据英文名称获取国家信息
接口分类	Country
访问方式	Get
参数类型	字符型
返回类型	Json 格式字符串 (Country 对象)
C#样例	<pre> HttpClient client = new HttpClient(); client.BaseAddress = new Uri(APIUrl); HttpResponseMessage response = client.GetAsync("/V1/Country/GetByEnName?name=Canada").Result; if (response.IsSuccessStatusCode) {     var jsonStr = response.Content.ReadAsStringAsync().Result;     var obj = JsonSerializer.Deserialize&lt;Country&gt;(jsonStr);     Console.WriteLine("{0} ({1})", (int)response.StatusCode, response.ReasonPhrase); } else {     Console.WriteLine("{0} ({1})", (int)response.StatusCode, response.ReasonPhrase); } </pre>

### 3. 2. 5. 根据英文别名获取国家信息函数 (GetByEnAlias)

GetByEnAlias (string name)

函数名称	GetByEnAlias (string name)
访问路径	/V1/Country/GetByEnAlias?name=???

验证需要	否
功能说明	根据英文别名获取国家信息
接口分类	Country
访问方式	Get
参数类型	字符型
返回类型	Json 格式字符串 (Country 对象)
C#样例	<pre> HttpClient client = new HttpClient(); client.BaseAddress = new Uri(APIUrl); HttpResponseMessage response =client.GetAsync("V1/Country/GetByEnAlias?name=Russia").Result; if (response.IsSuccessStatusCode) {     var jsonStr = response.Content.ReadAsStringAsync().Result;     var obj = JsonSerializer.Deserialize&lt;Country&gt;(jsonStr);     Console.WriteLine("{0} ({1})", (int)response.StatusCode, response.ReasonPhrase); } else {     Console.WriteLine("{0} ({1})", (int)response.StatusCode, response.ReasonPhrase); } </pre>

### 3.3. 运输渠道查询 (Channel)

函数表:

NO#	函数名称	接口分类	访问路径	方式	功能说明
1	GetAll()	Channel	/V1/Channel/GetAll	Get	获取所有运输方式信息
2	GetById(int id)	Channel	/V1/ Channel /GetById?id=???	Get	根据运输方式 ID 获取运输方式信息
3	GetByClassId(int id)	Channel	/V1/ Channel/GetByClassId? id=???	Get	根据运输方式类别 ID 获取运输方式信息
4	GetByCnName(string name)	Channel	/V1/ Channel/GetByCnName?n ame=???	Get	根据运输方式中文名称获取运输方式信息
5	GetByEnName(string name)	Channel	/V1/ Channel/GetByEnName?n ame=???	Get	根据运输方式英文名称获取运输方式信息
6	GetByCountryId(int id)	Channel	/V1/Channel/GetByCoun tryId?id=???	Get	根据国家 Id 获取运输方式信息

Json 结构:

```

{"ChannelId":6,"ChannelClassId":5,"ChannelCode":"HKBRAM","CnName":"香港邮政小包挂号",
"EnName":"HK Post iMailR Economy Class Service","RefreTime":"7-20","ImagePath":null}

```

No#	属性名称	属性含义	类型	默认值	特别说明
1	ChannelId	运输方式 Id	整型		
2	ChannelClassId	运输方式类别 Id	整型		
3	ChannelCode	运输方式代码	字符		
4	CnName	中文名称	字符		
5	EnName	英文名称	字符		
6	RefreTime	参考时效	字符		

7	ImagePath	图片路径	字符	null	
---	-----------	------	----	------	--

### 3.3.1. 获取所有运输方式信息函数(GetAll)

#### GetAll

函数名称	GetAll()
访问路径	/V1/Channel/GetAll
验证需要	否
功能说明	获取所有国家信息
接口分类	Channel
访问方式	Get
返回类型	Json 格式字符串 (Channel 集合)
C#样例	请参考国家信息的获取代码

### 3.3.2. 根据 Id 获取运输方式信息函数(GetById)

#### GetById(int id)

函数名称	GetById(int id)
访问路径	/V1/Channel/GetById?id=???
验证需要	否
功能说明	获取所有国家信息
接口分类	Channel
访问方式	Get
参数类型	整型
返回类型	Json 格式字符串 (Channel 对象)
C#样例	请参考国家信息的获取代码

### 3.3.3. 根据类别 Id 获取运输方式信息函数(GetByClassId)

#### GetByClassId(int id)

函数名称	GetByClassId(int id)
访问路径	/V1/Channel/GetByClassId?id=???
验证需要	否
功能说明	根据运输方式类别 Id 获取运输方式信息
接口分类	Channel
访问方式	Get
参数类型	整型
返回类型	Json 格式字符串 (Channel 对象)
C#样例	请参考国家信息的获取代码



## 3.3.4. 根据中文名称获取运输方式信息函数 (GetByCnName)

GetByCnName(**string** name)

函数名称	GetByCnName( <b>string</b> name)
访问路径	/V1/Channel/GetByCnName?name=???
验证需要	否
功能说明	根据中文名称获取运输方式信息
接口分类	Channel
访问方式	Get
参数类型	字符型
返回类型	Json 格式字符串 (Channel 对象)
C#样例	<i>请参考国家信息的获取代码</i>

## 3.3.5. 根据英文名称获取运输方式信息函数 (GetByEnName)

GetByEnName(**string** name)

函数名称	GetByEnName( <b>string</b> name)
访问路径	/V1/Channel/GetByEnName?name=???
验证需要	否
功能说明	根据英文名称获取运输方式信息
接口分类	Channel
访问方式	Get
参数类型	字符型
返回类型	Json 格式字符串 (Channel 集合)
C#样例	<i>请参考国家信息的获取代码</i>

## 3.3.6. 根据国家 Id 获取运输方式信息函数 (GetByCountryId)

GetByCountryId(**int** id)

函数名称	GetByCountryId( <b>int</b> id)
访问路径	/V1/Channel/GetByCountryId?id=???
验证需要	否
功能说明	根据国家 Id 获取运输方式信息
接口分类	Channel
访问方式	Get
参数类型	整型
返回类型	Json 格式字符串 (Channel 集合)
C#样例	<i>请参考国家信息的获取代码</i>

### 3.4. 运输方式类别查询 (ChannelClass)

函数表:

No#	函数名称	接口分类	访问路径	方式	功能说明
1	GetAll()	ChannelClasses	/V1/ChannelClass/GetAll	Get	获取所有运输方式类别信息
2	GetById(int id)	ChannelClasses	/V1/ChannelClass/GetById?id=???	Get	根据 ID 获取运输方式类别信息

Json 结构:

```
{"ChannelClassId":1,"CnName":"国际快递","EnName":"International Express"}
```

No#	属性名称	属性含义	类型	默认值	特别说明
1	ChannelClassId	运输方式类别 Id	整型		
2	CnName	中文名称	字符		
3	EnName	英文名称	字符		

#### 3.4.1. 获取所有运输方式类别信息函数 (GetAll)

GetAll

函数名称	GetAll()
访问路径	/V1/ChannelClass/GetAll
验证需要	否
功能说明	获取所有运输方式类别信息
接口分类	ChannelClass
访问方式	Get
返回类型	Json 格式字符串 (ChannelClass 集合)
C#样例	请参考国家信息的获取代码

#### 3.4.2. 根据 Id 获取运输方式类别信息函数 (GetById)

GetById(int id)

函数名称	GetById(int id)
访问路径	/V1/ChannelClass/GetById?id=???
验证需要	否
功能说明	根据 Id 获取运输方式类别信息
接口分类	ChannelClass
访问方式	Get
参数类型	整型
返回类型	Json 格式字符串 (ChannelClass 对象)

C# 样例

请参考国家信息的获取代码

### 3.5. 订单信息查询及增加 (Order)

一个货运订单，包含订单主体信息 (OrderMain) 及物品明细信息 (OrderDetail) 两部分。在调用接口上传订单数据时，请按照规则完善数据的填写。

函数表：

NO#	函数名称	接口分类	访问路径	方式	功能说明
1	GetByReferCode (string code)	Order	/V1/Order/GetByReferCode?code=???	Get	根据订单参考号 (即用户自定义的唯一标识) 获取订单信息
2	GetByOrderCode (string code)	Order	/V1/Order/GetByOrderCode?code=???	Get	根据订单号 (系统生成的唯一标识) 获取订单信息
3	Create (OrderMain orderMain)	Order	/V1/Order/Create	Post	增加订单
4	CreateByList (List<OrderMain> orderMainList)	Order	/V1/Order/CreateByList	Post	增加多个订单
5	Update ([FromBody] OrderMain orderMain)	Order	/V1/Order/Update	Post	修改订单
6	UpdateWeight ([FromBody] OrderWeightModel model)	Order	/V1/Order/UpdateWeight	Post	修改预报重量

Json 结构：

#### 1. OrderMain:

```
{ "OrderMainId": 47537, "UserId": 8110, "ManagerCode": "KFY0002", "OrderType": 1, "OrderCode": "A81101307220019", "ReferCode": "", "TrackNo": "8969948002", "ToCountryId": 2, "ToCountryName": "US", "ChannelCode": "CNDHL", "PackageType": 1, "BillQTY": 1, "FactWeight": 6.000, "PriceWeight": 11.2728, "FactLength": 61.000, "FactWidth": 42.000, "FactHeight": 22.000, "Weight": null, "ToPostCode": "TW19 7NL ", "ToName": "Roy Douthwaite", "ToEmail": null, "ToAddress1": "Cirrus House, Bedford Road, Heathrow Airport Staines Middlesex", "ToAddress2": "", "ToProvince": "London", "ToCity": "London", "ToCompany": "AIR CARGO TRADER LTD", "ToContact": "+44 1784 266210", "IsSecure": 0, "SecureValue": 288.00, "IsSign": 0, "IsReject": 0, "IsBatory": 64, "FromName": "Dennis Tan", "FromCompany": "", "FromAddress": "No.103,1/F, 33rd Building Shahe Industrial Zone, FuTian District", "FromPhone": "0755-83580089-811", "FromCountryId": 64, "FromProvince": null, "FromCity": null, "FromZipCode": null, "Remark": "", "SalesAmount": 602.15, "Status": 6, "OrderDetailList": [{"OrderDetailId": 46127, "OrderMainId": 47537, "MaterialReferNo": "", "CnDesc": "", "EnDesc": "Please refer to the invoice", "MaterialQuantity": 36, "Price": 8.00, "CustomcCode": "", "ProductArea": ""}] }
```

#### 2. ReturnMessage:

```
{ "Status": "success", "OrderMainId": 10002, "OrderCode": "KFY001", "ReferCode": "Customr0001", "AssignTrackNo": 1, "TrackNo": "E938893737", "Message": "上传成功", "MessageDetailList": [{"FieldName": "ReferCode", "Description": ""}] }
```

### 订单主表对象

No#	属性名称	属性含义	类型	默认值	规则	特别说明
1	OrderMainId	订单主表ID	整型		不须	系统生成的订单唯一编号
2	UserId	用户ID	整型		不须	
3	ManagerCode	业务经理代码	字符		不须	
4	OrderType	订单类型	整型	1	不须	以下枚举值之一： 1. 普通订单
5	OrderCode	订单代码	字符		不须	系统生成的订单唯一代码
6	ReferCode	订单参考号	字符		选填	用户自定义的唯一标识
7	TrackNo	跟踪号	字符		选填	
8	ToCountryId	收件人国家ID	整型		2 选 1	可从Country接口获取国家ID, 如果设置, 则优先于ToCountryName
9	ToCountryName	收件人国家名称	字符		2 选 1	如果ToCountryId没有设置值, 则可填写国家二字母 或 中文国家名称 或 英文国家名称
10	ChannelCode	运输方式代码	字符		必须	请从Channel接口获取产品代码 (或向您的业务经理咨询了解), 需要验证这个运输方式是否到达收件人国家
11	PackageType	包裹类型	整型	1	选填	以下枚举值之一： 1. 包裹 2. 文件
12	BillQTY	票数	整型	1	选填	包裹个数
13	FactWeight	实际重量	货币		不须	单位 KG
14	PriceWeight	计费重量	货币		不须	单位 KG, 如果体积重大于实际重量, 则取体积重
15	FactLength	实长	货币		不须	单位 CM
16	FactWidth	实宽	货币		不须	单位 CM
17	FactHeight	实高	货币		不须	单位 CM
18	Weight	预计重量	货币		选填	单位 KG
19	ToPostCode	收件人邮编	字符		必须	
20	ToName	收件人名称	字符		必须	
21	ToEmail	收件人邮箱	字符		必须	
22	ToAddress1	收件人地址1	字符		必须	
23	ToAddress2	收件人地址2	字符		选填	
24	ToProvince	收件人省/洲	字符		必须	
25	ToCity	收件人城市	字符		必须	
26	ToCompany	收件人公司名称	字符		必须	
27	ToContact	收件人联系信息	字符		必须	填写收件人的手机, 电话等信息
28	IsSecure	是否保险 0否, 1是	整型	0	选填	
29	SecureValue	申请保险价值	货币		不须	由物品明细的价格总数计算得出
30	IsSign	是否电子签名 0否, 1是	整型	0	选填	0 否, 1 是
31	IsReject	是否退件	整型	0	选填	0 否, 1 是
32	IsBatory	是否带电池 0否, 1是	整型		选填	0 否, 1 是
33	FromName	发件人姓名	字符		选填	

34	FromCompany	发件人公司	字符		选填	
35	FromAddress	发件人地址	字符		选填	
36	FromPhone	发件人电话	字符		选填	
37	FromCountryId	发件人国家ID	整型	64 (中国)	选填	
38	FromProvince	发件人省/洲	字符		选填	
39	FromCity	发件人城市	字符		选填	
40	FromZipCode	发件人邮编	字符		选填	
41	Remark	订单备注	字符		选填	
42	SalesAmount	订单费用	货币		不须	
43	Status	订单状态	整型	3	不须	<ol style="list-style-type: none"> <li>待修改</li> <li>待交寄</li> <li>已交寄</li> <li>已收未付</li> <li>已收已付</li> <li>已发货</li> <li>已完成</li> <li>已退件</li> <li>已删除(逻辑)</li> </ol>
44	CODAmount	COD代收金额	货币		选填	COD 渠道必填
45	CODCurrency	COD代收币种	字符		选填	COD 渠道填对应国家币种
46	ToFCEShortCode	收件FCE服务地址短代码	字符		选填	FCE 渠道必填
47	DutyPaymentType	关税付款方	整型		选填	1. 收件人付款, 2 发件人付款, 3. 第三方付款。目前暂不支持第三方付款
48	OrderDetailList []	订单物品明细	对象集合		不须	请参考<订单物品明细对象>描述

### 订单物品明细对象

No#	属性名称	属性含义	类型	默认值	规则	特别说明
1	OrderDetailId	订单明细ID	整型		不须	
2	OrderMainId	订单主表ID	整型		不须	
3	MaterialReferNo	物品参考号	字符		必须	
4	CnDesc	中文描述	字符		选填	
5	EnDesc	英文描述	字符		选填	
6	MaterialQuantity	数量	整型		必须	
7	Price	单个价格	货币		必须	不能为 0
8	CustomCode	海关编码	字符		选填	部分运输方式必填
9	ProductArea	产地	字符		选填	默认是为 China
10	ProductWeight	物品单件重量	货币		选填	
11	ParcelNumber	一票多件包裹、箱序号	整型		选填	部分 FBA、快递渠道必填, 从开始, 如物品在第一箱填 1
12	ParcelShipmentCode	箱运单号, 如FBA运单号	字符		选填	选题, 如"FBA44BNR46U001"
13	ECommerceUrl	物品电商销售链接	字符		选填	

### 订单修改预报重量对象 OrderWeightModel

No#	属性名称	属性含义	类型	默认值	特别说明
1	OrderCode	订单号	字符		创建订单时返回的订单号
2	Weight	预报重量	货币		要更新的预报重量

## ReturnMessage

No#	属性名称	属性含义	类型	默认值	特别说明
1	Status	状态	字符		枚举值: success, error 说明: success (成功), error (失败)
2	Message	信息内容	字符		信息描述内容
3	OrderCode	订单号	字符		系统订单号, 如果创建失败, 些属性为空
4	ReferCode	订单参考号	字符		用户自定义的订单标识
5	Index	序号	整型		订单的序号
6	MessageDetailList	信息明细	对象集合	null	信息明细列表
7	OrderMainId	订单主键	整型	null	系统的订单主表主键值
8	TrackNo	订单跟踪号	字符		系统的跟踪号,

## MessageDetail

No#	属性名称	属性含义	类型	默认值	特别说明
1	FieldName	字段名称	字符		对应订单主表的属性名称
2	Description	信息描述	字符		信息描述内容

注: 新建订单时, 系统会对订单的属性内容进行验证, 如果验证不通过, 产生一个 MessageDetail 描述。

## 3.5.1. 根据订单参考号获取订单信息函数 (GetByReferCode)

## GetByReferCode

函数名称	GetByReferCode(string code)
访问路径	/V1/Order/GetByReferCode?code=???
验证需要	须要
功能说明	根据订单参考号获取订单信息
接口分类	Order
访问方式	Get
返回类型	Json 格式字符串 (OrderMain 对象)
C#样例	<pre> HttpClient client = new HttpClient(); client.BaseAddress = new Uri(APIUrl); string apiKey = "1b35e1dd-4271-49c2-3c01-2cc5s0de1ddf"; //用户Id + 14码时间 (yyyyMMddHHmmss) string userStr = "8110" + "_" + DateTime.Now.ToString("yyyyMMddHHmmss"); string signStr = SAFSign.Sign(userStr, apiKey, "utf-8"); //设置HttpRequest验证信息 client.DefaultRequestHeaders.Add("kfy_userid", userStr); client.DefaultRequestHeaders.Add("kfy_signdata", signStr);  HttpResponseMessage response = client.GetAsync("V1/Order/GetByReferCode?code=1306290028").Result; if (response.IsSuccessStatusCode) {     var jsonStr = response.Content.ReadAsStringAsync().Result;     var orderMain = JsonSerializer.Deserialize&lt;OrderMain&gt;(jsonStr);     Console.WriteLine("{0} ({1}), {2}", (int)response.StatusCode, response.ReasonPhrase, orderMain); } else { </pre>

```
Console.WriteLine("{0} ({1})", (int)response.StatusCode, response.ReasonPhrase);
}
```

### 3.5.2. 根据订单号获取订单信息函数 (GetByOrderCode)

#### GetByOrderCode

函数名称	GetByOrderCode( <a href="#">string</a> code)
访问路径	/V1/Order/GetByOrderCode?code=???
验证需要	须要
功能说明	根据订单参考号获取订单信息
接口分类	Order
访问方式	Get
返回类型	Json 格式字符串 (OrderMain 对象)
C#样例	请参考 GetByReferCode( <a href="#">string</a> code) 函数

### 3.5.3. 增加订单信息函数 (Create)

**提示:** 增加订单, 需要构造一个订单信息, 包含 OrderMain(主体信息) 及 OrderDetail(物品明细信息), 以下字段的赋值需要符合系统的规则:

1. ToCountryId(收件人国家 Id): *必须与系统的国家匹配;*
2. ChannelId(运输方式 Id): *必须与系统的运输方式匹配;*
3. ReferCode(参考号): *唯一;*

#### Create

函数名称	Create( <a href="#">OrderMain</a> orderMain)
访问路径	/V1/Order/Create
验证需要	须要
功能说明	增加系统货运订单
接口分类	Order
访问方式	Post
返回类型	Json 格式字符串 (ReturnMessage 对象)
C#样例	<pre>HttpClient client = new HttpClient(); client.BaseAddress = new Uri("http://localhost:1405/"); string apiKey = "1b35e1dd-4276-49c2-8c01-2cc5d0dd1ddf"; // 用户Id + 14码时间 (yyyyMMddHHmmss) string userStr = "8110" + "_" + DateTime.Now.ToString("yyyyMMddHHmmss"); string signStr = SAFSign.Sign(userStr, apiKey, "utf-8"); //设置HttpRequest验证信息 client.DefaultRequestHeaders.Add("kfy_userid", userStr); client.DefaultRequestHeaders.Add("kfy_signdata", signStr);  // 获取到达法国的国家信息, France为法国英文别名 HttpResponseMessage response = client.GetAsync("V1/Country/GetByEnAlias?name=France").Result; Country country = null; Channel channel = null; if (response.IsSuccessStatusCode) {</pre>

```

var jsonStr = response.Content.ReadAsStringAsync().Result;
country = JsonSerializer.Deserialize<Country>(jsonStr);
}

if (country != null)
{
    // 根据国家Id获取到达这个国家的运输方式
    response = client.GetAsync(string.Format("V1/Channel/GetByCountryId?id={0}",
country.CountryId)).Result;
    IEnumerable<Channel> channels = null;
    if (response.IsSuccessStatusCode)
    {
        var jsonChannelStr = response.Content.ReadAsStringAsync().Result;
        channels = JsonSerializer.Deserialize<IEnumerable<Channel>>(jsonChannelStr);

        // 选择一个运输方式
        if (channels != null)
            channel = channels.FirstOrDefault();
    }
}

if (country != null && channel != null)
{
    OrderMain entity = new OrderMain();
    entity.ToCountryId = country.CountryId; // <-- (从此API获取国家Id)
    entity.ChannelId = channel.ChannelId; // <-- (从此API获取可到达此国家的运输方式)

    entity.BillQTY = 1;
    entity.FromAddress = "No.103,1/F, 33rd Building Shahe Industrial Zone ,
Baishizhou Nanshan District.";
    entity.FromCity = "Shenzhen";
    entity.FromCompany = "";
    entity.FromCountryId = 64;
    entity.FromName = "Dennis Tan";
    entity.FromPhone = "";
    entity.FromProvince = "Guangdong";
    entity.FromZipCode = "518030";

    entity.IsBatory = 0;
    entity.IsReject = 0;
    entity.IsSecure = 0;
    entity.IsSign = 0;
    entity.ManagerCode = "";
    entity.PackageType = 0;

    entity.ReferCode = DateTime.Now.ToString("yyyyMMddHHmmss");
    entity.Remark = "";
    entity.ToAddress1 = "Mohaliver 85";
    entity.ToAddress2 = "Mohaliver 85 Mohaliver 85";
    entity.ToCity = "Yehud";
    entity.ToCompany = "";
    entity.ToContact = "(050)336-1005";
    entity.ToEmail = "bislli17@walla.com";
    entity.ToName = "Orly Eldar";
    entity.ToProvince = "Israel";
    entity.ToPostCode = "56208";
    entity.TrackNo = "";
}

```



```

entity.UserId = 8110;

IList<OrderDetail> list = new List<OrderDetail>();
OrderDetail detail = new OrderDetail();
detail.Price = Convert.ToDecimal(0.1);
detail.MaterialQuantity = 1;
detail.EnDesc = "gift";
detail.CnDesc = "礼物";
list.Add(detail);

entity.OrderDetailList = list;

response = client.PostAsJsonAsync<OrderMain>("V1/Order/Create",
entity).Result;
if (response.IsSuccessStatusCode)
{
    var returnValue = response.Content.ReadAsAsync<ReturnMessage>().Result;
    Console.WriteLine("调用状态为{0}, 订单创建状态: {1}", response.StatusCode,
returnValue.Status);
    if (returnValue.MessageDetailList != null)
    {
        foreach (MessageDetail ex in returnValue.MessageDetailList)
        {
            Console.WriteLine("字段为{0}, 错误原因: {1}", ex.FieldName,
ex.Description);
        }
    }
    else
    {
        Console.WriteLine("订单号{0}", returnValue.OrderCode);
    }
}
else
{
    Console.WriteLine("{0} ({1}), {2}", (int)response.StatusCode,
response.ReasonPhrase, "Calling Error...");
}
}

```

### 3.5.4. 修改订单信息函数 (Update)

#### Update

函数名称	Update (OrderMain orderMain)
访问路径	/V1/Order/Update
验证需要	须要
功能说明	修改系统货运订单
接口分类	Order
访问方式	Post
返回类型	Json 格式字符串 (ReturnMessage 对象)
C#样例	请参考 SAF.Api.ClientSample 代码 OrderSample
说明	修改订单时, 部分信息不能变更, 如果订单状态已收货 (Status>3), 则整个订单信息都不能修改。

	部分主要信息不能变更: UserId 不能修改; OrderMainId 不能修改;; OrderCode 不能修改; Status 不能修改;
--	--

### 3.5.5. 修改订单预报重量函数(UpdateWeight)

#### Update

函数名称	UpdateWeight (OrderWeightModel model)
访问路径	/V1/Order/UpdateWeight
验证需要	须要
功能说明	修改系统订单预报重
接口分类	Order
访问方式	Post
返回类型	Json 格式字符串 (ReturnMessage 对象)
C#样例	请参考 SAF.Api.ClientSample 代码 OrderSample
说明	修改订单时, 部分信息不能变更, 如果订单状态已收货 (Status>3), 则整个订单信息都不能修改。
请求 Json	<pre>{   "OrderCode": "A100001711020001",   "Weight": 0.5 }</pre>

### 3.5.6. 删除订单函数(DeleteById)

#### DeleteById

函数名称	DeleteById (int orderMainId)
访问路径	/V1/Order/DeleteById?orderMainId=xxx
验证需要	须要
功能说明	删除系统货运订单
接口分类	Order
访问方式	Get
返回类型	bool
C#样例	请参考 SAF.Api.ClientSample 代码 OrderSample
说明	订单状态 Status > 3, 即包裹已入库之后, 不能删除订单. 执行顺序: 1. 根据参考号或 KFY 单号查询到订单信息: GetByReferCode GetByOrderCode, 并获得 OrderMainId.

## 2. 再调用/V1/Order/DeleteById?orderMainId=xxx

C# Example:

```
var order = OrderExample.GetByReferCode("C16022716852");
if (order != null)
{
    OrderExample.DelOrder(order.OrderMainId);
}
```

```
public static OrderMain GetByReferCode(string referCode)
{
    try
    {
        HttpClient client = new HttpClient();
        client.BaseAddress = new Uri(APIConfig._ApiURL);

        // 设置验证信息
        SignHelper.BuildHttpRequestSinData(ref client);

        HttpResponseMessage response =
client.GetAsync("V1/Order/GetByReferCode?code=" + referCode).Result;
        if (response.IsSuccessStatusCode)
        {
            var jsonStr = response.Content.ReadAsStringAsync().Result;
            var orderMain = JsonHelper.Deserialize<OrderMain>(jsonStr);

            return orderMain;
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    return null;
}

public static void DelOrder(int id)
{
    HttpClient client = new HttpClient();
    client.BaseAddress = new Uri(APIConfig._ApiURL);

    // 设置验证信息
    SignHelper.BuildHttpRequestSinData(ref client);

    HttpResponseMessage response =
client.GetAsync("V1/Order/DeleteById?orderMainId=" + id).Result;
    if (response.IsSuccessStatusCode)
    {
        var val = response.Content.ReadAsStringAsync().Result;

        Console.WriteLine(string.Format("删除状态: {0}", val));
    }
    else
    {
        Console.WriteLine("{0} ({1})", (int)response.StatusCode,
response.ReasonPhrase);
    }
}
```

## 3. 6. 跟踪号查询 (Order)

### 3. 6. 1. GetTrackNoByReferCode

函数名称	GetTrackNoByReferCode ( <b>string</b> code)
访问路径	/V1/Order/GetTrackNoByReferCode?code=???
验证需要	须要
功能说明	根据订单参考号获取跟踪号
接口分类	Order
访问方式	Get
返回类型	<b>string</b>
C#样例	请参考 GetByReferCode( <b>string</b> code) 函数

### 3. 6. 2. GetTrackNoByOrderCode

函数名称	GetTrackNoByOrderCode ( <b>string</b> code)
访问路径	/V1/Order/GetTrackNoByOrderCode?code=???
验证需要	须要
功能说明	根据订单号获取跟踪号
接口分类	Order
访问方式	Get
返回类型	<b>string</b>
C#样例	请参考 GetByReferCode( <b>string</b> code) 函数

## 4. Label (标签) 获取

### 4. 1. 1. 获取 HTML 标签

标签地址: <http://uc.iquickfish.com/p/print/PrintAPiLabel.aspx>

标签返回的是 HTML 内容, 具体设置请参考以下方式:

// 标签打印地址:

<http://uc.iquickfish.com/p/print/PrintAPiLabel.aspx?PrintOrderCodeList=单个订单号>

//热敏打印(10\*10) 地址标签 参数:

"&PrintPageStyle=2&PrintLabelType=1&PrintOrderByType=1&PrintIsPrintDetail=on&Print

```
IsPrintRetAddress=on"
```

```
//热敏打印(10*10) 地址标签 CN22 报关单
```

```
"&DocType=1&PrintPageStyle=2&PrintLabelType=3&PrintOrderByType=1&PrintIsPrintDetail=on&PrintIsPrintRetAddress=on"
```

```
//A4 地址标签
```

```
"&DocType=1&PrintPageStyle=1&PrintLabelType=1&PrintOrderByType=1&PrintIsPrintDetail=on&PrintIsPrintRetAddress=on"
```

```
//A4 地址标签 CN22 报关单
```

```
"&DocType=1&PrintPageStyle=1&PrintLabelType=3&PrintOrderByType=1&PrintIsPrintDetail=on&PrintIsPrintRetAddress=on"
```

如订单号为 A100001603210001, 要打印热敏 10\*10 地址标签 URL 为:

```
http://uc.iquickfish.com/p/print/PrintAPiLabel.aspx?PrintOrderCodeList=A100001603210001&PrintPageStyle=2&PrintLabelType=1&PrintOrderByType=1&PrintIsPrintDetail=on&PrintIsPrintRetAddress=on
```

#### 4.1.2. 获取 PDF 标签

通过订单号获取标签: GetLabelByOrderCode

函数名称	GetLabelByOrderCode ( <a href="#">string</a> code)
访问路径	<a href="#">/V1/Order/GetLabelByOrderCode?code=???</a>
验证需要	须要
功能说明	根据订单号获取标签
接口分类	<a href="#">Order</a>
访问方式	Get
返回类型	Json 格式字符串 ( <a href="#">LabelRespModel</a> 类型) 返回 Json 示例: <pre>{   "OrderCode": "A100001801250001",   "LabelUrl": "http://xxx.com/A100001801250001_180126150015_117.pdf" }</pre>
C#样例	请参考 <a href="#">GetLabelByOrderCode (string code)</a> 函数

#### LabelRespModel

No#	属性名称	属性含义	类型	默认值	特别说明
1	<a href="#">OrderCode</a>	订单号	字符		订单号
2	<a href="#">LabelUrl</a>	标签地址	字符		PDF 标签下载地址

## 5. 跟踪轨迹

### 5.1.1. 单条订单跟踪信息查询

函数名称	GetTrackInfo (string code)
访问路径	/v1/Order/GetTrackInfo?code=???
验证需要	须要
功能说明	根据订单号获取跟踪信息
接口分类	Order
访问方式	Get
返回类型	Json 格式字符串 (string)
C#样例	请参考 GetTrackInfo (string code) 函数

请求参数:

名称	类型	是否必需	描述
code	string	Y	单号, 订单号或者跟踪号

返回结果:

名称	类型	描述
trackNo	String	跟踪号
orderCode	String	订单号
toCountry	String	目的国家
channelCode	String	渠道代码
referCode	String	参考号
success	String	是否成功 true 成功, false 失败
msg	String	返回信息, 成功为空, 失败为错误信息
trackingData	List<object>	返回 List 类型跟踪信息, object 中包含 track_time, track_string
track_time	String	时间
track_string	String	跟踪内容

```
{
  "success": true,
  "msg": "",
  "orderCode": "KFYBR7081800675YQ",
  "trackNo": "LS916111998CH",
  "toCountry": "巴西",
  "channelCode": "BREXP",
  "referCode": "T201708180001",
  "trackingData": [
    {
      "track_time": "2017-08-18T10:15:00+08:00",
      "track_string": "SERVICE REQUEST 2017-08-18 10:15:00"
    }
  ],
  "status": 1
}
```

## 6. API 接口使用建议

### 6.1. 使用本地程序缓存

区域，运输方式，运输方式类别信息等内容，建议定期获取并放置本地应该程序缓存，以提高程序效率。

### 6.2. 异常处理

调用远程 API 接口服务，很难保证一如既往的正常返回结果信息，建议在调用 API 时，编写程序异常处理逻辑。（注：文档中附带的代码样例，未进行异常信息处理）

### 6.3. API 测试数据

在对接 API 完成时，需要一些测试的数据以调试程序，请在 Remark 属性的赋值“APITest”等内容。

### 6.4. 时间差

在调用 API 时，需要验证的 Token 有效期(有效时间差为 1 小时), 请保正调用主机时间正确。