

---

# 速递易开放平台API说明

曾浩 <zenghao@isantai.cn>

## Table of Contents

1. 说明	2
2. POST <code>/\${version}/resv</code> 发起预约请求	2
2.1. 请求参数列表	2
2.2. 返回状态码	3
2.3. 返回参数列表	3
2.4. 请求消息示例	3
2.5. 返回消息示例	3
3. POST <code>notify_url</code> 回调预约结果	4
3.1. 回调消息格式	4
3.2. 回调消息示例	4
4. PUT <code>notify_url</code> 回调预约状态变化	5
4.1. 回调消息格式	5
4.2. 回调消息示例	5
5. DELETE <code>/\${version}/resv/:resv_order_id</code> 取消预约	6
5.1. 返回状态码	6
6. GET <code>/\${version}/resv/:resv_order_id</code> 查询预约订单状态	6
6.1. 返回消息格式	6
6.2. 返回消息示例	7
7. GET <code>/\${version}/area</code> 查询行政区划	7
7.1. 返回消息格式	7
7.2. 返回消息示例	7
8. GET <code>/\${version}/lattice?area=:area_id</code> 获取网点和设备信息	8
8.1. 返回消息格式	8
8.2. 返回消息示例	8
9. GET <code>/\${version}/boxStatus</code> 获取箱格状态	9
9.1. 返回消息格式	9
9.2. 返回消息示例	9
10. GET <code>/\${version}/boxStatus/online</code> 获取箱格状态和设备在线状态	10
10.1. 返回消息格式	10
10.2. 返回消息示例	10
11. GET <code>/\${version}/deadletter</code> 获取死信	10
11.1. 返回消息格式	11

11.2.	返回消息示例	11
12.	GET <code>/\${version}/resv</code>	11
12.1.	返回消息格式	12
12.2.	返回消息示例	12
13.	GET <code>/\${version}/closest</code>	13
13.1.	查询参数格式	13
13.2.	返回消息格式	13
13.3.	返回消息示例	13

## 1. #说明

本文涉及的全部API都采用 REST<sup>1</sup> 风格，请求域名为 `open.sudiyi.cn`，`version` 目前为 `v1`。例如发起预约的地址就是 `http://open.sudiyi.cn/v1/resv`。如果你的请求返回了403，请仔细阅读 [认证文档](#)<sup>2</sup>。

如果希望快速接入，也可以使用我们提供的 [SDK](#)<sup>3</sup>。

## 2. #POST `/${version}/resv` 发起预约请求

向服务器发起预约请求，注意该请求返回仅代表服务器收到了需要预约的请求，至于预约结果则是在之后的服务器回调中返回。

### 2.1. #请求参数列表

参数名	类型	是否必填	说明
<code>device_id</code>	String	是	速递易设备ID
<code>box_type</code>	Int	是	箱格类型 0:大箱 1:中箱 2:小箱 3:冰箱
<code>notify_url</code>	String	是	回调第三方系统的地址
<code>auto_upgd</code>	Boolean	否	是否可以自动升箱
<code>sender_name</code>	String	否	投递人姓名
<code>sender_mobile</code>	String	否	投递人手机
<code>order_no</code>	String	是	第三方订单号
<code>consignee_mobile</code>	String	是	取件人手机号

<sup>1</sup> <https://zh.wikipedia.org/wiki/REST>

<sup>2</sup> [authenticate.html](#)

<sup>3</sup> [sdk.html](#)

consignee_name	String	否	取件人姓名
payment	Integer	否	到付金额，如果不是到付件则为0
duration	Integer	否	预约时长，单位分钟，默认240分钟
pay_type	Integer	否	超期付费方式 0:快递柜付费 1:从商户账户扣除

## 2.2. #返回状态码

- 200 提交订单成功
- 420 请求过于频繁
- 421 超过预约上限

## 2.3. #返回参数列表

参数名	类型	说明
resv_order_no	Int	生成的预约订单号

## 2.4. #请求消息示例

```
{
  "device_id":1000081,
  "box_type":0,
  "notify_url":"http://localhost:8080",
  "auto_upgd":true,
  "sender_name": "testuser",
  "order_no": "12312312321",
  "consignee_mobile": "13535353535",
  "consignee_name": "收货人"
}
```

## 2.5. #返回消息示例

```
{
  "resv_order_no" : 999
}
```

### 3. #POST notify\_url 回调预约结果



该请求由速递易开放平台发起，第三方需要提供可接受调用的URL地址

预约成功或失败时，系统主动回调第三方提供的接口，通知预约结果，地址为提交预约订单时指定的 `notify_url`

#### 3.1. #回调消息格式

参数名	类型	说明
resv_order_no	Int	预约订单的编号
result	String	预约结果*
order_no	String	第三方订单, 为发起预约时传入的订单号
box_type	Int	实际取得的箱格的类型
upgraded	Boolean	是否进行了升箱
box_no	Int	箱格序号
resv_code	String	预约码

其中`result`包括

- success 预约成功，将返回全部字段
- fail 预约失败，失败原因是箱格已满。将仅返回 `resv_order_no` 和 `result`
- timeout 预约超时

#### 3.2. #回调消息示例

成功时返回

```
{
  "resv_order_no":119,
  "result":"success",
  "box_type":2,
  "upgraded":true,
  "box_no":20,
  "resv_code":"qwe324"
```

```
}

```

#### 失败时返回

```
{
  "resv_order_no":119,
  "result":"fail"
}
```

## 4. #PUT notify\_url 回调预约状态变化



该请求由速递易开放平台发起，第三方需要提供可接受调用的URL地址

当用户投件或取件时，回调该接口，地址为提交预约订单是指定的 `notify_url`。

### 4.1. #回调消息格式

参数名	类型	说明
resv_order_no	Int	预约订单的编号
status	Int	订单的状态码
open_code	String	开箱码

其中状态码：

- 5 已投件，等待取件
- 6 已取件，流程结束
- 7 预约超过保留时长用户仍未投件，预约箱格自动释放

### 4.2. #回调消息示例

```
{
  "resv_order_no":119,
  "status":6,
  "open_code":"ABCDEF"
}
```

## 5. #DELETE \${version}/resv/:resv\_order\_id 取消预约



只能取消处于等待投件状态的订单

取消指定预约号的预约，其中 `resv_order_id` 为预约订单号。保留的箱格资源将被释放。将根据截止取消时箱格的保留时间来结算使用费用（当前版本未收费 2015-10-23）。

### 5.1. #返回状态码

- 200 取消成功
- 204 取消失败

## 6. #GET \${version}/resv/:resv\_order\_id 查询预约订单状态

其中 `resv_order_id` 为预约订单号

### 6.1. #返回消息格式

参数名	类型	说明
<code>resv_order_no</code>	String	预约订单号
<code>status</code>	Int	订单状态码

其中状态码

- 1: 预约订单创建, 等待响应
- 2: 预约成功, 等待投件
- 3: 预约失败
- 4: 预约超时
- 5: 已投件, 等待取件
- 6: 已取件

- 7: 投件超时, 用户未在保留时间内投件, 箱格自动释放
- 8: 取消预约
- 9: 取给取件人
- 10: 取给投件人
- 11: 值守暂存

## 6.2. #返回消息示例

```
{
  "resv_order_no": 199,
  "status": 5
}
```

## 7. #GET \${version}/area 查询行政区划

获取行政区划的信息, 该接口返回的数据量较大且变动不频繁, 建议用户缓存该接口返回的信息。

### 7.1. #返回消息格式

参数名	类型	说明
id	Int	行政区划的ID
n	String	name, 行政区划的名称
c	Array	children, 类型为本类型

### 7.2. #返回消息示例

```
[{
  "id": 510000,
  "n": "四川",
  "c": [
    {
      "id": 510100,
      "n": "成都",
      "c": [{
        "id": 510107,
        "n": "武侯区"
      }]
    }
  ]
}]
```

```

    },
    {
      "id": 510700,
      "n": "绵阳"
    }
  ], {
    "id": 110000,
    "n": "北京"
  }
}]

```

## 8. #GET \${version}/lattice?area=:area\_id 获取网点和设备信息

获取行政区划下所有的网点和设备信息，其中area\_id为第三级行政区划id且不能为空

### 8.1. #返回消息格式

参数名	类型	说明
id	Int	网点ID
name	String	网点名称
devices	Array[Device]	网点下属设备列表

其中 Device 定义如下

参数名	类型	说明
id	Int	设备ID
name	string	设备名称
address	String	设备地址

### 8.2. #返回消息示例

```

[ {
  "id": 1,
  "name": "网点名称",
  "devices": [ {
    "id": 1,
    "name": "设备名称",
    "address": "dizhi1"
  } ], {

```



```

    "id": 2,
    "name": "设备名称2",
    "address": "dizhi2"
  ]],
  {
    "id": 2,
    "name": "网点名称2"
  ]}

```

## 9. #GET \${version}/boxStatus 获取箱格状态

必须带有查询字符串 `device` 表示按设备查询, 或者 `lattice` 表示按网点查询. `device` 和 `lattice` 后跟设备或网点的id, 例如

```
GET /v1/boxStatus?device=999
```

或者

```
GET /v1/boxStatus?lattice=999
```

获取可用的箱格数量, 必须在查询参数中指定 `device` 或者 `lattice`. 如果指定了 `device` 则按照设备进行查询, 如果指定了 `lattice`, 则按照网点进行查询. `device` 和 `lattice` 分别代表设备ID和网点ID.

### 9.1. #返回消息格式

参数名	类型	说明
small	Int	可用的小箱的数量
medium	Int	可用的中箱的数量
big	Int	可用的大箱的数量

### 9.2. #返回消息示例

```

{
  "small": 1,
  "medium": 2,
  "big": 3
}

```

## 10. #GET \${version}/boxStatus/online 获取箱格状态和设备在线状态

和 GET `${version}/boxStatus` 类似，必须带有查询字符串 `device` 表示按设备查询，或者 `lattice` 表示按网点查询。 `device`和`lattice`后跟设备或网点的id。不同的是如果快递柜不在线返回结果为空。

### 10.1. #返回消息格式

返回结果为以下类型的数组：

参数名	类型	说明
<code>device</code>	<code>Int</code>	设备ID
<code>status</code>	<code>BoxStatus</code>	可预约箱格数量，类型同 <code>BoxStatus</code> ，如果设备不在线，该字段为空

### 10.2. #返回消息示例

```
[
  {
    "device": 1000076,
    "status":
      {
        "small": 17,
        "medium": 0,
        "big": 8
      }
  },
  {
    "device": 1000077
  },
  {
    "device": 1000088
  }
]
```

## 11. #GET \${version}/deadletter 获取死信

获取回调失败的消息。调用结束后已经被读取的消息将被清空，下次调用返回的将是新的没有读取过的消息。

## 11.1. #返回消息格式

参数名	类型	说明
post	Array	失败的预约成功的消息
put	Array	失败的箱格状态变化的消息

## 11.2. #返回消息示例

```
{
  "post": [
    {
      "result": "success",
      "upgraded": false,
      "resv_order_no": 20151015000119,
      "box_type": 1,
      "box_no": 1,
      "resv_code": "123456",
      "order_no": "131313"
    }
  ],
  "put": [
    {
      "status": 5,
      "resv_order_no": 20151015000119,
      "open_code": "123456"
    }, {
      "status": 6,
      "resv_order_no": 20151015000119,
      "open_code": "123456"
    }
  ]
}
```

## 12. #GET \${version}/resv

查询订单当前的详细状态，比如指定 `order_no`（第三方订单号）或者 `resv_order_no`（预约号作为查询参数），例如：

GET [http://open.sudiyi.cn/v1/resv?order\\_no=TEST001](http://open.sudiyi.cn/v1/resv?order_no=TEST001)

或者

GET [http://open.sudiyi.cn/v1/resv?resv\\_order\\_no=20151125000001](http://open.sudiyi.cn/v1/resv?resv_order_no=20151125000001)

结果按 id 倒序进行排列。

## 12. 1. #返回消息格式

参数名	类型	说明
resv_order_no	number	预约号
order_no	String	第三方运单号
send_mobile	String	投件人电话号码
consignee_mobile	String	取件人电话号码
create_at	number	创建时间的时间戳
status	number	状态
resv_code	String	预约码
open_code	String	开箱码

## 12. 2. #返回消息示例

```
[
  {
    "status": 6,
    "resv_order_no": 20151125000001,
    "order_no": "111",
    "send_mobile": "18040304193",
    "consignee_mobile": "18040304193",
    "create_at": 1448420310000,
    "resv_code": "CH4YMS",
    "open_code": "1111"
  },
  {
    "status": 7,
    "resv_order_no": 20151117000088,
    "order_no": "111",
    "send_mobile": "18040304193",
    "consignee_mobile": "18040304193",
    "create_at": 1447756625000,
    "resv_code": "WG6SHR"
  },
  {
    "status": 4,
    "resv_order_no": 20151117000087,
    "order_no": "111",
    "send_mobile": "18040304193",
```

```

    "consignee_mobile": "18040304193",
    "create_at": 1447756595000,
    "open_code": "1111"
  }
]

```

## 13. #GET \${version}/closest

获取附近设备，包括以给定坐标为中心，中心与四边相距各五公里的正方形内的所有设备。

### 13.1. #查询参数格式

参数名	类型	说明
lat	number	中心点的纬度
lng	number	中心点的经度

例如：GET <http://open.sudiyi.cn/v1/closest?lat=30.5452&lng=104.065>

### 13.2. #返回消息格式

返回为以下数据类型的列表

参数名	类型	说明
id	number	设备id
name	string	设备名称
address	string	设备地址
lat	number	设备纬度
lng	number	设备经度
distance	number	设备距离当前点位的距离

### 13.3. #返回消息示例

```

[ {
  "id": 1000018,
  "name": "SDY成都武侯区-香年广场-01",
  "address": "4503A办公室近门口",
  "lat": 30.5453,
  "lng": 104.066,

```

```
"distance": 300
},
{
  "id": 1000115,
  "name": "香年办公室测试T-2",
  "address": "香年广场4503A最里面",
  "lat": 30.548,
  "lng": 104.064,
  "distance": 205
}]
```

---