1.创建托运人账户

URI: /v1/consignors

方法: POST

有效载荷:

请求header

**标签 参数类型 描述**

API-Key 否

Content-Type string application/json

请求body

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 参数描述 |
| name | String | 用户名 |
| Email | String | 用户邮箱 |
| password | String | 用户密码 |

响应参数

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 参数描述 |
| Code | String | 成功为0 |
| data | object | 数据字典，包括id（用户id(string)），name（用户名(string)），email（用户邮箱(string)） |

例子：

{

"name": "John Doe",

"email": "john.doe@example.com",

"password": "mypassword"

}

响应成功：

{

Code：0，

Data：{

"id": 123,

"name": "John Doe",

"email": "john.doe@example.com"

}

}

失败：

{

Code：1，

}

2. 创建一个航运订单

请求

URI: /v1/shipping-orders

方法: POST

有效载荷:

**标签 参数类型 描述**

API-Key 否

Content-Type string application/json

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 参数描述 |
| consignor\_id | String | 托运人id |
| destination\_address | String | 目的地 |
| shipping\_items | List | 托运货物清单 |
| name  quantity | String | 货物名称 |
| String | 货物数量 |
| weight | Double | 货物重量（单位：kg） |

响应参数

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 参数描述 |
| Code | String | 成功为0 |
| data | object | 数据字典，包括下方所有参数 |
| consignor\_id | String | 托运人id |
| destination\_address | String | 目的地 |
| Id | String | 订单编号 |
| Status | String | 订单当前状态 |
| shipping\_items | List | 托运货物清单（包括下方） |
| shipping\_items .name  shipping\_items .quantity | String | 货物名称 |
| string | 货物数量 |
| shipping\_items .Id | String | 该货物批次里的编号 |
| shipping\_items weight | Double | 货物重量（单位：kg） |

例子：

{

"consignor\_id": 123,

"destination\_address": "123 Main St, Anytown USA",

"shipping\_items": [

{

"name": "item1",

"quantity": 10,

"weight": 20

},

{

"name": "item2",

"quantity": 5,

"weight": 30

}

]

}

响应成功：

{

"id": 456,

"consignor\_id": 123,

"destination\_address": "123 Main St, Anytown USA",

"shipping\_items": [

{

"id": 1,

"name": "item1",

"quantity": 10,

"weight": 20

},

{

"id": 2,

"name": "item2",

"quantity": 5,

"weight": 30

}

],

"status": "new"

}

失败：

{

"error": "Invalid payload"

}

3. 检索承运人对不同货物的帐单

请求

URI: /v1/bills

方法: GET

查询参数：

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 参数描述 |
| consignor\_id: | String | 托运人id |
| id | String | 订单编号(可选) |
| from\_date: | Date | 开始日期（可选） |
| to\_date: | Date | 结束日期（可选） |

**标签 参数类型 描述**

API-Key 否

Content-Type string application/json

例子：

响应成功：

{

"bills": [

{

"id": 1,

"shipping\_order\_id": 456,

"consignor\_id": 123,

"amount": 1000.00,

"date": "2021-01-01"

},

{

"id": 2,

"shipping\_order\_id": 789,

"consignor\_id": 123,

"amount": 2000.00,

"date": "2021-02-01"

}

]

}

失败状态码：400 Bad Request

{

"error": "Invalid query parameters"

}

4. 追踪货物和运输情况

请求

URI: /v1/shipping-orders/tracking

方法: Post

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 参数描述 |
| Id | String | 托运订单编号 |

:id是航运订单的ID。

**标签 参数类型 描述**

API-Key 否

Content-Type string application/json

响应参数:

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 参数描述 |
| Code | String | 成功为0 |
| data | object | 数据字典，包括下方所有参数 |
| consignor\_id | String | 托运人id |
| destination\_address | String | 目的地 |
| Id | String | 订单编号 |
| Status | String | 订单当前状态 |
| shipping\_items | List | 托运货物清单（包括下方） |
| shipping\_items .name  shipping\_items .quantity | String | 货物名称 |
| string | 货物数量 |
| shipping\_items .Id | String | 该货物批次里的编号 |
| shipping\_items weight | Double | 货物重量（单位：kg） |

例子：

响应成功：

{

Code:0,

Data:{

"id": 456,

"consignor\_id": 123,

"destination\_address": "123 Main St, Anytown USA",

"shipping\_items": [

{

"id": 1,

"name": "item1",

"status": "已发货",

"origin": "起始地1",

"destination": "目的地1",

"currentLocation":"当前位置"，

"arrival\_date": "2022-01-01"

}

}

}