

《达达商城》订单模块API文档

模型类

一、订单模块概述

二、事件定义

1. 生成【确认订单】页
2. 生成订单并支付
3. 订单查询
4. 确认收货
5. 订单支付

三、API说明

1. 确认订单API

- 1.1 事件触发
- 1.2 请求
- 1.3 响应

2. 生成订单API

- 2.1 事件触发
- 2.2 请求
- 2.3 响应

3. 订单查询API

- 3.1 事件触发
- 3.2 请求
- 3.3 响应

4. 确认收货API

- 4.1 事件触发
- 4.2 请求
- 4.3 响应

《达达商城》订单模块API文档

模型类

```
from django.db import models
from goods.models import SKU
from users.models import UserProfile

STATUS_CHOICES = (
    (1, "待付款"),
    (2, "待发货"),
    (3, "待收货"),
    (4, "订单完成")
)

class OrderInfo(models.Model):
    # 订单表
    # 用户表:订单表 ---> 1:n
    # 订单编号、订单总金额、支付方式、运费、订单状态、收货地址相关字段[反范式]
    user_profile = models.ForeignKey(UserProfile, on_delete=models.CASCADE)
    # 20220301143930007
    order_id = models.CharField(max_length=64, primary_key=True, verbose_name="订单编号")
```

```

# 订单总金额、商品总数量
total_amount = models.DecimalField(max_digits=10, decimal_places=2,
verbose_name="订单总金额")
total_count = models.IntegerField(verbose_name="商品总数")
# 支付方式:1代表支付宝
pay_method = models.SmallIntegerField(default=1, verbose_name="支付方式")
# 运费
freight = models.DecimalField(max_digits=10, decimal_places=2,
verbose_name="运费")
# 订单状态
status = models.SmallIntegerField(verbose_name="订单状态",
choices=STATUS_CHOICES)

# 冗余地址字段
receiver = models.CharField(verbose_name="收件人", max_length=10)
address = models.CharField(verbose_name="收件地址", max_length=100)
receiver_mobile = models.CharField(verbose_name="手机号", max_length=11)
tag = models.CharField(verbose_name="标签", max_length=10)

created_time = models.DateTimeField(auto_now_add=True)
updated_time = models.DateTimeField(auto_now=True)

class Meta:
    db_table = "orders_order_info"

class OrderGoods(models.Model):
    """
    订单商品表
    订单表:订单商品表 ---> 1:n
    SKU表:订单商品表 ---> 1:n
    """
    order_info = models.ForeignKey(OrderInfo, on_delete=models.CASCADE)
    sku = models.ForeignKey(SKU, on_delete=models.CASCADE)

    # 数量、单价
    count = models.IntegerField(default=1, verbose_name="数量")
    price = models.DecimalField(max_digits=10, decimal_places=2, verbose_name="单价")

    created_time = models.DateTimeField(auto_now_add=True)
    updated_time = models.DateTimeField(auto_now=True)

    class Meta:
        db_table = "orders_order_goods"

```

一、订单模块概述

此为达达商城订单功能模块API说明文档,此文档对于常规业务逻辑的流程进行分析和针对订单模块的技术点进行分析。订单模块提供以下功能:

生成【确认订单】页

生成订单并支付

支付订单

二、事件定义

1. 生成【确认订单】页

用户在购物车页面点击确认**去结算**按钮，前端跳转到确认订单页(orderconfirm.html)发送Ajax请求(get)到后端，后端从redis中获取**商品信息**和从mysql中获取**用户地址信息**响应给到订单确认页展示。

2. 生成订单并支付

用户在【确认订单】页面点击**确认并付款**按钮，前端跳转到支付订单页(payment.html)发送Ajax请求(post)到后端，后端生成订单且将**订单编号、订单状态（默认为待支付）、订单金额**等信息响应给前端订单支付页展示。此次请求带回第三方支付的URL。用户在支付订单页点击**确认支付**按钮，前端获取事件使用**window.location**去访问第三方支付**pay_URL**，之后由支付宝提供支付业务。支付完成支付宝重定向到订单支付结果页面，并异步通知到商城后端。双重验证后返回支付结果给前端展示。

3. 订单查询

用户在页面头部点击**我的订单**图标，前端跳转到**我的订单页**(Myorder.html)发送Ajax请求(get)到后端,后端查询订单数据相应给前端。前端展示依据订单的状态**全部订单、待付款、待发货、待收货、已完成**进行展示数据即**订单编号、订单金额、成交时间和交易状态**。

4. 确认收货

用户在我的订单页面点击确认收货,前端获取点击事件向后端发送请求(get).后端将订单状态改为**已完成**。

5. 订单支付

用户在我的订单页面点击**去支付**按钮，前端跳转到支付订单页(payment.html)发送Ajax请求(post)到后端,后端生成订单且将**订单编号、订单状态（默认为待支付）、订单金额**等信息响应给前端订单支付页展示。此次请求带回第三方支付的URL。

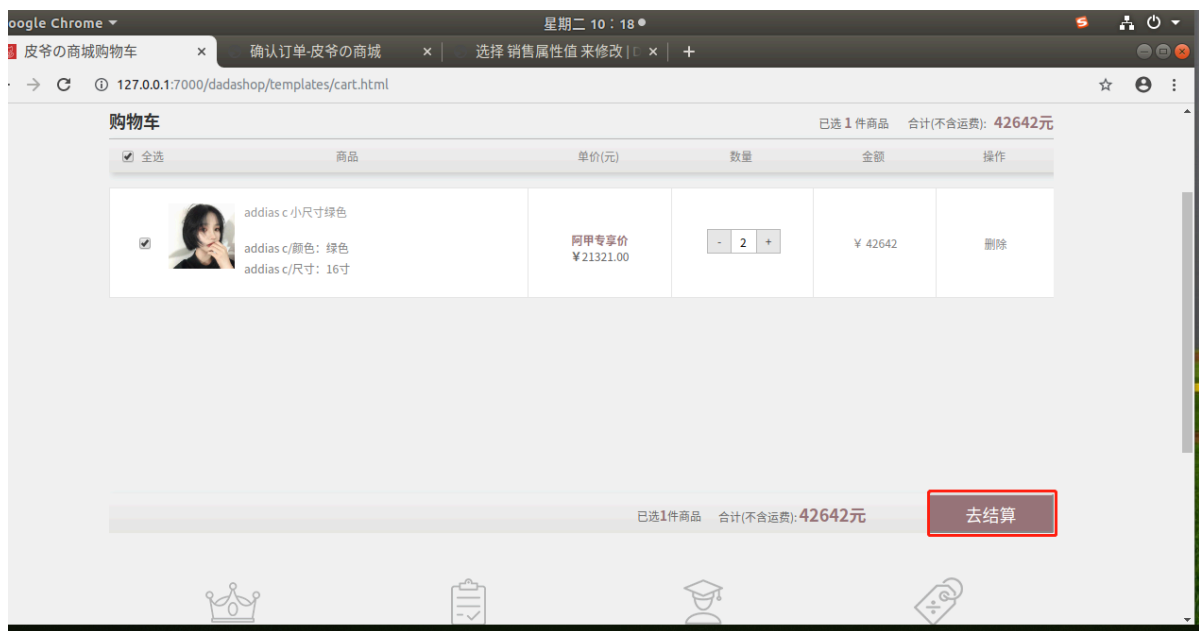
三、API说明

1. 确认订单API

用户在登录情况下才能够点击去结算并跳转到确认订单页。

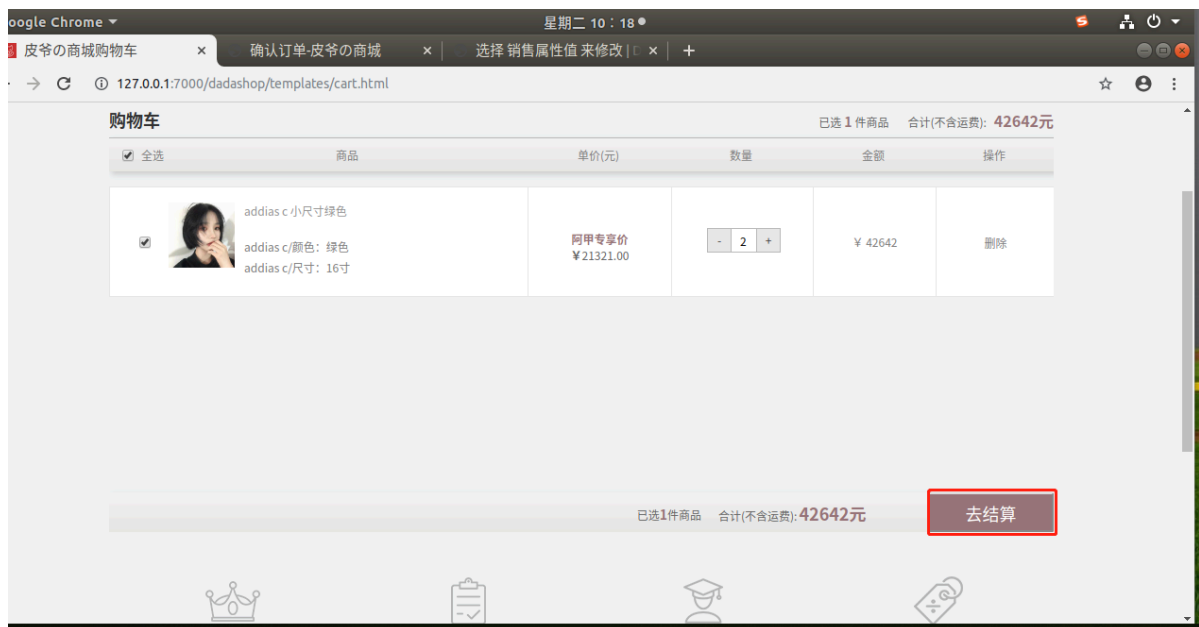
1.1 事件触发

点击去结算按钮



事件触发(异常):用户无收货地址

用户点击去结算按钮



点击确定按钮

127.0.0.1 显示
地址为空,请添加地址

确定

添加地址展示

127.0.0.1:7000/dadashop/templates/orderConfirm.html

确认订单信息 支付订单 支付完成

收货人信息

kun

更多地址 >>

确认商品信息

addias c 小尺寸绿色
addias c/颜色:绿色
尺寸

<<返回购物车

新增收货地址

* 收货人: 琨琨

* 省份: 北京市 城市: 北京市市辖区 区/县: 朝阳区

* 详细地址: 十里堡别墅群

* 手机号码: 18812341234

* 邮政编码: 000000

* 地址名称: 公司 如: 家 公司 宿舍

保存收货人信息

必填信息必须填写, 点击保存收货信息

金额

42642

确认并付款

确认订单页展示数据

皮皮科技 Py Technology

请输入您要搜索的内容

注册 | sad

确认订单信息 支付订单 支付完成

收货人信息

kun 北京市北京市市辖区通州区天安门 18812341234

更多地址 >>

确认商品信息

商品信息	单价(元)	数量	金额
addias c 小尺寸绿色 addias c/颜色:绿色 addias c/尺寸:16 寸	皮粉专属价 ¥21321.00	2	¥42642

<<返回购物车

共1件商品 合计:42642元

确认并付款

1.2 请求

- **URL:** [http://127.0.0.1:8000/v1/orders/\[username\]/advance](http://127.0.0.1:8000/v1/orders/[username]/advance)
- **请求方式:** GET
- **请求参数:**

参数	类型	是否必须	说明
settlement_type	int	是	结算类型

- **请求示例:**

```
http://127.0.0.1:8000/v1/orders/sad/advance?settlement_type=0
```

1.3 响应

返回值类型:JSON

响应内容:

字段	含义	类型	备注
code	状态码	int	默认正常为200，异常见状态码
data	具体数据	dict	与error二选一

响应格式:

```
{"code": 200, "data": data, "base_url": ""}
```

data参数信息:

参数	类型	是否必须	说明
addresses	list	是	用户收货地址列表
sku_list	list	是	商品信息

data数据示例:

```
{
  "addresses":
    [
      {
        "id":1,
        "name":"guoxiao",
        "mobile":"13488873110",
        "title":"家",
        "address":"北京市北京市市辖区东城区啊啊啊"
      }
    ],
  "sku_list":
    [
```

```

{
  "id":2,
  "name":"安踏A红色大尺寸",
  "count":1,
  "selected":1,
  "default_image_url":"2_i2NMNkA.png",
  "price":"200.00",
  "sku_sale_attr_name":["安踏A/尺寸","安踏A/颜色"],
  "sku_sale_attr_val":["18寸","红色"]
},
{
  "buy_count": buy_num,
  "sku_id": sku_id
}

```

状态码参考:

状态码	响应信息
200	正常
错误状态码	错误原因

2. 生成订单API

2.1 事件触发

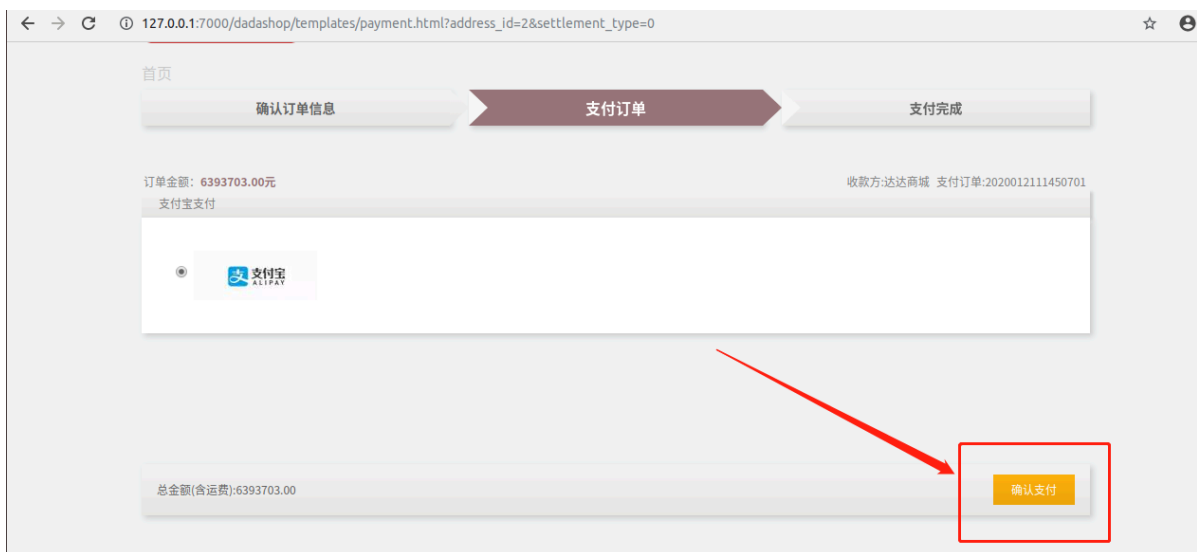
点击确认并付款按钮



订单支付页展示



点击确认并支付



2.2 请求

URL: [http://127.0.0.1:8000/v1/orders/\[username\]](http://127.0.0.1:8000/v1/orders/[username])

请求方式: POST

请求参数:

参数	类型	是否必须	说明
address_id	int	是	收货地址id
settlement_type	int	是	类型(0-购物车 1-立即购买)
buy_count	int	否	商品数量-立即购买链条
sku_id	int	否	商品id-立即购买链条

前端js代码


```
if(buy_count!=0&&sku_id!=0){
    var data={
        "address_id":address_id,
        "settlement_type":settlement_type,
        "buy_count":buy_count,
        'sku_id':sku_id
    }
}else {
    var data={
        "address_id":address_id,
        "settlement_type":settlement_type
    }
}
```

请求示例:

```
http://127.0.0.1:8000/v1/orders/sad
```

2.3 响应

返回值类型:JSON

响应内容:

字段	含义	类型	备注
code	状态码	int	默认正常为200，异常见状态码
data	具体数据	dict	与error二选一
error	错误信息	char	与data二选一

响应格式:

```
{"code": 200, "data": data}
```

data参数信息:

参数	类型	是否必须	说明
saller	str	是	商家名称
total_amount	decimail	是	总价格
order_id	str	是	订单号
pay_url	str	是	第三方支付路由

data数据示例:

```
{
  'saller': '达达商城',
  'total_amount': 1314,
  'order_id': 2020021601,
  'pay_url': 'http://alipay.com/xxx/xx/',
  'carts_count': 0
}
```

error参数信息:

参数	类型	是否必须	说明
errmsg	str	是	错误信息说明

error数据示例:

```
{"code": 错误状态码, "error": "错误原因"}
```

状态码参考:

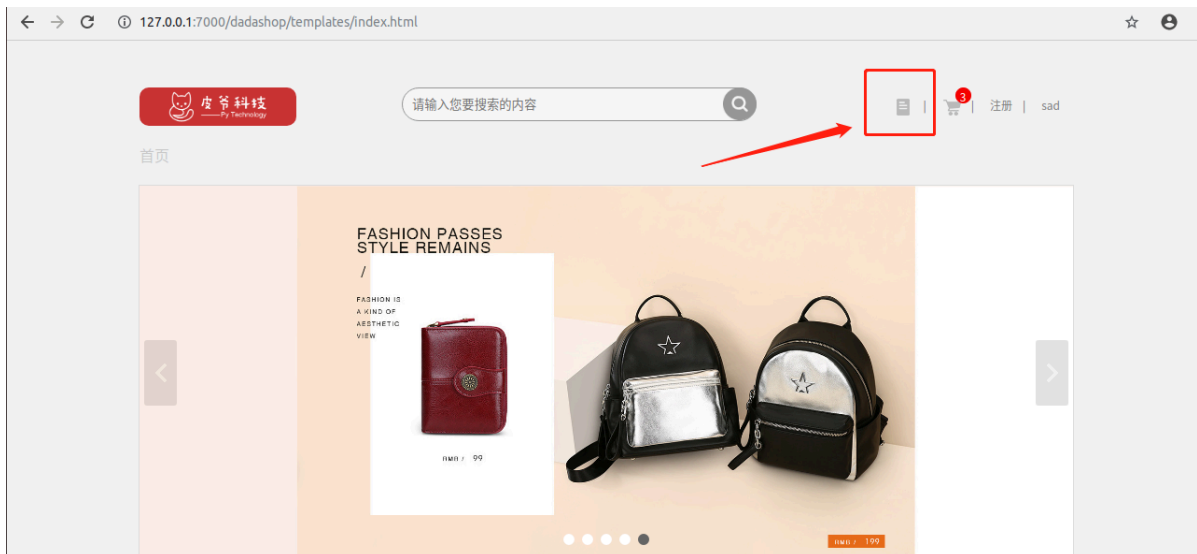
状态码	响应信息
200	正常
错误状态码	错误原因

3. 订单查询API

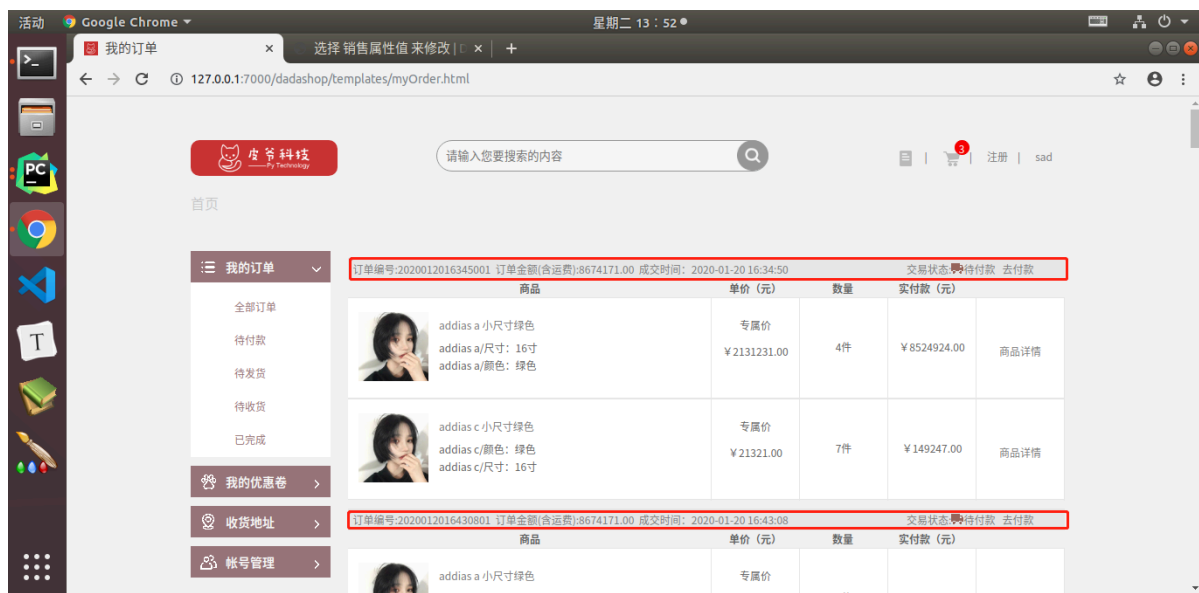
3.1 事件触发

在登录状态下才可查看订单信息

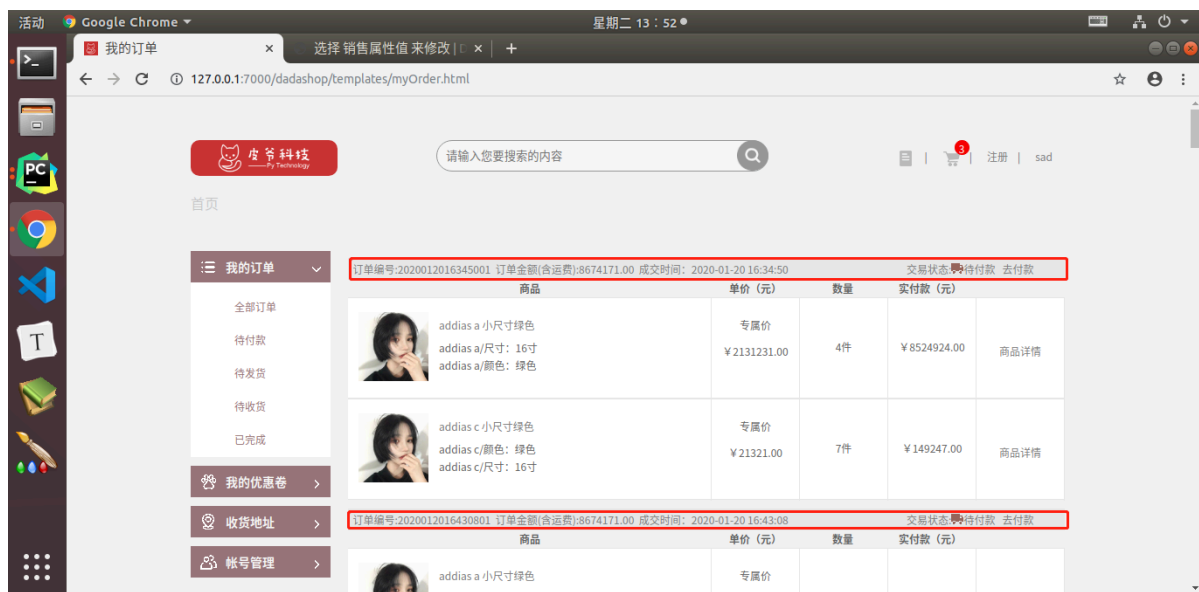
任意页面header点击订单图标



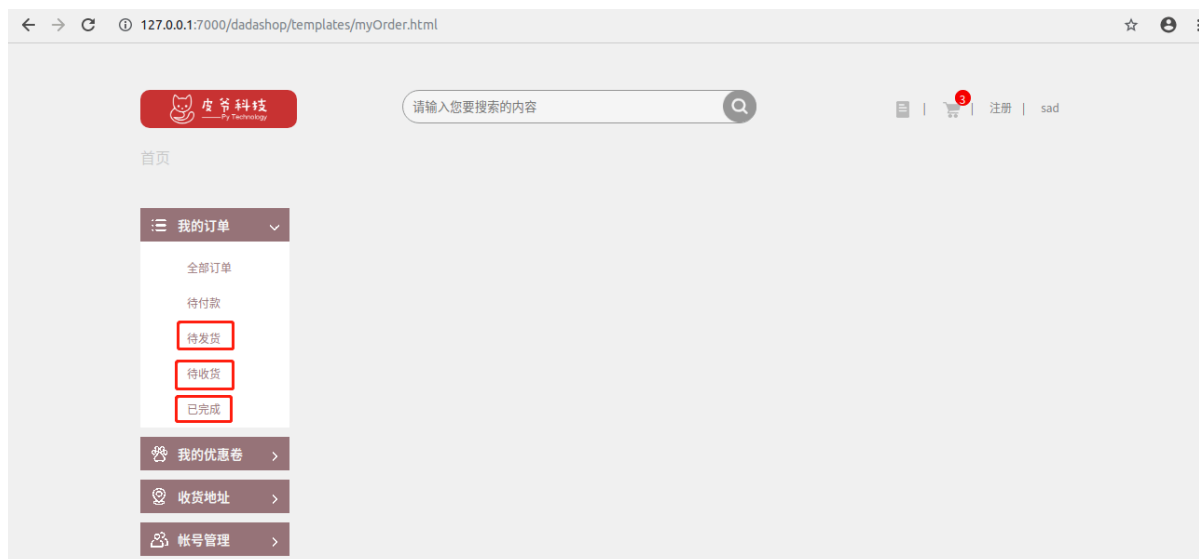
点击全部订单，即可查看全部订单



订单页点击待付款按钮，即可查看待付款的订单



订单页点击待发货(待收货,已完成)按钮，即可查看待发货(待收货,已完成)的订单



3.2 请求

URL: [http://127.0.0.1:8000/v1/orders/\[username\]](http://127.0.0.1:8000/v1/orders/[username])

请求方式:GET

请求参数:

参数	类型	是否必须	说明
type	str	是	订单状态

请求示例:

```
http://127.0.0.1:8000/v1/orders/sad?type=0
```

3.3 响应

返回值类型:JSON

响应内容:

字段	含义	类型	备注
code	状态码	int	默认正常为200，异常见状态码
data	具体数据	dict	与error二选一
error	错误信息	char	与data二选一

响应格式:

```
{"code":200,"data":data,"base_url": PIC_URL}
```

data参数信息:

参数	类型	是否必须	说明
orders_list	list	是	订单信息列表

data数据示例:

```
"data":{
  "orders_list":[
    {
      "order_id":"2019111811504601",
      "order_total_count":1,
      "order_total_amount":"209.00",
      "order_freight":"1.00",
      "address":{
        "title":"家",
        "address":"北京市北京市市辖区东城区珍贝大厦",
        "mobile":"13691433520",
        "receiver":"习瓜瓜"
```

```
{
  "status": 1,
  "order_sku": [
    {
      "id": 4,
      "default_image_url": "2_940nDrI.jpg",
      "name": "adidas a",
      "price": "199.00",
      "count": 1,
      "total_amount": "199.00",
      "sku_sale_attr_names": [ "颜色", "尺寸" ],
      "sku_sale_attr_vals": [ "红色", "15寸" ]
    }
  ],
  "order_time": "2019-11-18 11:50:46"
}
```

error参数信息:

参数	类型	是否必须	说明
error_msg	str	是	错误信息说明

error数据示例:

```
{"code": 错误状态码, "error": "错误原因"}
```

状态码参考:

状态码	响应信息
200	正常

4. 确认收货API

4.1 事件触发

在登录状态下才可使用此功能

用户在订单页点击确认收货按钮

我的订单

全部订单

待付款

待发货

待收货

已完成

我的优惠券

收货地址

帐号管理

订单编号:2019111821363401 订单金额(含运费):219.00 成交时间: 2019-11-18 21:36:34

交易状态: 已发货 确认收货

商品	单价 (元)	数量	实付款 (元)	
 红袖添香 颜色: 红色 尺寸: 18寸	专属价 ¥209.00	1件	¥209.00	商品详情

页面展示



点击确定后展示



4.2 请求

URL: [http://127.0.0.1:8000/v1/orders/\[username\]](http://127.0.0.1:8000/v1/orders/[username])

请求方式: PUT

请求参数:

参数	类型	是否必须	说明
order_id	str	是	订单状态

请求示例:

```
127.0.0.1:8000/v1/orders/sad
```

4.3 响应

返回值类型:JSON

响应内容:

字段	含义	类型	备注
code	状态码	int	默认正常为200，异常见状态码

响应格式:

```
{"code":200}
```