

用户模块-接口说明

一.用户模块概述：

用户模块主要提供对于用户使用本产品的基础。用户模块提供以下功能：

- 找回密码
- 修改密码
- 用户激活
- 用户收货地址新增/修改/删除/设为默认
- 第三方登录(微博)

二、事件定义

找回密码

用户在登陆页可以通过点击找回密码按钮来实现找回密码功能，主要分为三步。首先通过用户发送注册时的邮箱，此时会向邮箱中发送一份关于找回密码的验证码，此验证码会有时效。通过输入对应的验证码，验证没问题之后，显示设置新密码界面，提交之后就可以使用新密码进行登录。

修改密码

修改密码功能需要在登录情况下进行，在个人账号管理页面中选择，需要输入旧密码以及新密码。通过ajax进行与后端交互。

用户激活

用户在注册的同时会同步发送用户激活的邮件。此激活邮件中存在用户激活的链接，用户通过点击此链接之后，在此页面内进行跳转到激活页面，此页面中有两个按钮，是否激活。用户点击激活之后，通过ajax发送激活请求。进行激活。

用户地址

地址查询功能：在用户访问到地址页面时，此时会发送用户绑定的地址请求，全量返回有效的用户地址。

增加地址功能：用户可以提交新增地址，包括收货人、收件地址、地址标签、以及用户的联系方式，通过ajax提交。如果正常存储的话，此时回全量返回用户绑定的地址。

修改地址功能：用户可以修改有效地址中的收件人、收件地址、练习方式、以及地址标签。提交成功的话，回全量返回用户的地址。

删除地址功能：用户可以在界面上触发删除地址按钮，此时会将地址的id通过ajax传入到后台，通过设置该地址的活跃属性来实现删除功能，此功能也是全量返回，然后通过渲染进行在前端展示。

设为默认地址：用户可以将任意一个非默认地址改为默认地址。通过ajax提交到后端，此功能提交的是地址的id，将此id的地址设置为默认，然后进行全量返回。进行渲染。

第三方登录：

用户可以在登录页面进行点击微博登录按钮，跳转到微博登录页面，用户输入微博账号密码，提交之后获得用户授权，此时如果用户已经存储在我们的数据库中，然后进行登录。如果用户在后台数据库中不存在，那么就让用户绑定个人信息。

三.API 说明

修改密码

我的订单

全部订单

待付款

待收货 1

待评价

退货退款

我的优惠券

收货地址

帐号管理

我的信息

安全管理

更改密码

输入旧密码：

输入新密码：

确认新密码：

保存更改

在登录的状态下才能够修改用户登录密码。

修改密码API：

URL: http://127.0.0.1:8000/v1/users/<username>/password

请求方法：POST

请求参数：JSON

字段	含义	类型	备注
oldpassword	旧密码	char	必填
password1	新密码	char	必填
password2	确认密码	char	必填

请求示例：

```
{
  'oldpassword': '123456',
  'password1': '123321',
  'password2': '654321'
}
```

字段	含义	类型	备注
code	状态	int	默认正常为200。异常见附录
error	错误信息	char	与data二选一
data	正确信息	dict	与error二选一

返回值：JSON

响应格式：

```
# 正确示例
{
  'code': 200,
  'data': {
    'message': '修改成功'
  }
}

# 错误示例
{
  'code': xxx,
  error: {
    'message': 'error_reason'
  }
}
```

找回密码功能

时序图



step1 发送邮箱验证请求:

皮爷科技
Py Technology

请输入您要搜索的内容

❤️ | 📄 | 🛒 | 帮助 | 登录

邮箱名称:

发送验证

发送邮箱验证

URL: `http://127.0.0.1:8000/v1/users/<username>/password/sms`

请求方法: POST

请求参数: JSON

字段	含义	类型	备注
email	邮箱	char	通过邮箱找回密码

request_json

```
{
  'email':xxxxx # 邮箱
}
```

响应数据

字段	含义	类型	备注
code	状态码	int	必填
data	返回数据	dict	与error二选一
error	返回错误原因	char	与data二选一

返回值**: JSON

响应格式:


正常响应

```
{
  'code': 200,
  'data': {
    'message': "邮件发送成功",
    'email': 'xxxx' # 返回邮箱
  }
}
```

异常响应

```
{
  'code': xxx,
  'error': 'error reason'
}
```

step2: 发送邮箱验证码

 皮爷科技
Py Technology

请输入您要搜索的内容

❤️ | 📁 | 🛒 | 帮助 | 登录

邮箱名称: 842549758@qq.com

验证码:

重置密码

发送邮箱验证码:

URL: `http://127.0.0.1:8000/v1/users/<username>/password/verification/`

请求方法: POST

请求参数: JSON

字段	含义	类型	备注
email	邮箱	char	必填
code	验证码	char	必填

```
{
  'email': 'xxxx',
  'code': 'xxxx'
}
```

响应数据：

字段	含义	类型	备注
code	响应状态码	char	必填
data	返回正常数据()	dict	与error二选一
error	返回错误原因()	char	与data二选一


返回值：JSON

响应格式：


```
# 正常响应
{
  'code': 200,
  'data': {
    'message': '验证成功',
    'email': 'xxxx' # 返回邮箱
  }
}




# 异常响应
{
  'code': xxx,
  'error': {
    'message': 'error reason'
  }
}
```

step3 修改密码



请输入您要搜索的内容



 |  |  | [帮助](#) | [登录](#)

原密码:

新密码:

确定新密码:

确定更新

修改密码：

URL： `http://127.0.0.1:8000/v1/users/password/renew`

请求方法： POST

请求参数： JSON

字段	含义	类型	备注
email	邮箱	char	必填
old_password	旧密码	char	必填
password1	新密码	char	必填
Password2	确认密码	char	必填

request_json

```
{
  'old_password': 'xxx' # 旧密码
  'password1': 'xxx' # 新密码
  'password2': 'xxx' # 确认密码
}
```

响应数据：

字段	含义	类型	备注
code	响应状态码	char	必填
data	返回正常数据	dict	与error二选一
error	返回异常数据	dict	与data二选一

返回值：JSON

响应格式：

```
# 正常数据
{
  'code': 200,
  'data': {
    'message': '修改成功'
  }
}

# 异常数据
{
  'code': xxx,
  'error': {
    'message': 'error reason'
  }
}
```

邮箱验证激活功能

在用户注册时候，后台会给用户注册填写的邮箱发送一条激活的链接。通过点击链接跳转到达达商城的用户激活页面。通过激活来实现商城所有功能的体验。

激活的页面



前端的激活链接如上图所示，通过点击是否激活按钮。点击按钮之后，前端页面的js代码会获取激活链接参数，将其拼接到后端路由。

```
var url_params = location.search # 获取前端地址输入栏的参数
var isBtn = document.getElementsByClassName('active_true')[0];
isBtn.onclick=function(){
  $.ajax({
    url: 'http://127.0.0.1:8000/v1/users/activation'+url_params,
    type: 'get',
    dataType: "json",
```



```
        success: function (d) {
            console.log(d)
            if (d.code == 200) {
                alert('激活成功! ')
                window.location.href="index.html" # 如果激活成功，跳出弹
框，并冲定向主页

            }else{
                alert(d.error.message) # 激活失败，弹出失败原因
            }
        }
    })
}
```

URL: http://127.0.0.1:8000/v1/users/activation?username=xxx&code=xxxx

请求方法: GET

请求参数: JSON

字段	含义	类型	备注
code	用户名和密码base64编码字符串	char	必填
username	用户名	char	必填

返回值: JSON

响应格式:

```
# 正常响应
{
  'code': 200,
  'data': {
    'message': '激活成功'
  }
}

# 异常响应
{
  'code': xxx,
  'error': {
    'message': 'error_reason'
  }
}
```

地址功能-增删改查设为默认

达达商城提供地址增删改察设为默认的功能:

查询功能：在用户请求的地址管理页面的时候，等页面主体加载完之后，会发送查询当前用户的绑定地址。返回当前用户绑定的所有地址。

增加地址功能：用户可以新增绑定地址。

修改地址功能：可以修改用户地址中的任何字段。

删除地址功能：可以删除用户所绑定的地址，不是真正的删除，而是在后台将地址改为非活跃的。

设为默认地址：用户可以将任意一个非默认地址改为默认地址。默认地址在用户完成付款之前会直接绑定到该订单上。

新增地址

页面加载完之后查询地址

点击修改地址

删除地址

设为默认地址

地址名称	姓名	地址详情	联系电话	操作	是否默认
公司	校长	广东省深圳市龙华区民治大道100号	18667018590	修改 删除	设置默认
公司	校长	广东省深圳市龙华区民治大道100号	18667018590	修改 删除	设置默认
公司	校长	广东省深圳市龙华区民治大道100号	18667018590	修改 删除	设置默认
公司	校长	广东省深圳市龙华区民治大道100号	18667018590	修改 删除	设置默认

新增地址和查询地址API：

URL: `http://127.0.0.1:8000/v1/users/<username>address/<id>`

查询地址：

请求方法：GET

请求参数：JSON

返回数据：在访问页面的时候，返回当前用户的所有地址

返回值：JSON

响应格式：

```
# 正常数据返回
{
  "code": 200,
```

```
"data":{
  "addressList":[
    {
      'id':123456, # 地址id
      'address':'广东省深圳市龙华区嘉熙业广场1155室', # 地址
      'receiver':'达内科技', # 收货人
      'receiver_mobile':'12345678901', # 联系电话
      'tag':'家', # 地址标签
      'postcode':'722494', #
      'is_default':"True",
    },
    {
      'id':123457,
      'address':'广东省深圳市龙华区嘉熙业广场1142室',
      'receiver':'达内科技',
      'receiver_mobile':'12345678910',
      'tag':'公司',
      'postcode':'722406',
      'is_default':"False",
    }
  ]
}
# 异常数据返回类型:
{
  'code': status_code,
  'error':{
    'message':'error_reason'
  }
}
```

新增地址

请求方法: POST

请求参数: JSON

参数	参数类型	含义	备注
receiver	char(10)	收件人	必填
receiver_phone	char(11)	收件人电话	必填
address	char(100)	地址	必填
postcode	char(7)	邮编	必填
tag	char(10)	标签	必填

request_json

```
{
  'receiver': '小王',
  'receiver_phone': '18667018590',
  'address': '北京市东城区珠市口大街珍贝大厦2楼',
  'postcode': '722405',
  'tag': '公司'
}
```

返回值: JSON

响应格式:

正常响应

```
{
  "code": 200,
  "data": {
    "addressList": [
      {
        'id': 123456, # 地址id
        'address': '广东省深圳市龙华区嘉熙业广场1155室', # 地址
        'receiver': '达内科技', # 收货人
        'receiver_mobile': '12345678901', # 联系电话
        'tag': '家', # 地址标签
        'postcode': '722494', #
        'is_default': "True",
      },
      {
        'id': 123457,
        'address': '广东省深圳市龙华区嘉熙业广场1142室',
        'receiver': '达内科技',
        'receiver_mobile': '12345678910',
        'tag': '公司',
        'postcode': '722406',
        'is_default': "False",
      }
    ]
  }
}
```

异常响应

```
{
  'code': status_code,
  'error': {
    'message': 'error_reason'
  }
}
```

删除地址

请求方法: DELETE

请求参数: JSON

参数	参数类型	含义	备注
id	char(10)	地址ID	必填
username	char(11)	用户名	必填

```
{
  'id': 'xxxxxx'
}
```

返回值: JSON

响应格式:

```
{
  "code": 200,
  "data": {
    "addressList": [
      {
        'id': 123456, # 地址id
        'address': '广东省深圳市龙华区嘉熙业广场1155室', # 地址
        'receiver': '达内科技', # 收货人
        'receiver_mobile': '12345678901', # 联系电话
        'tag': '家', # 地址标签
        'postcode': '722494', #
        'is_default': "True",
      },
      {
        'id': 123457,
        'address': '广东省深圳市龙华区嘉熙业广场1142室',
        'receiver': '达内科技',
        'receiver_mobile': '12345678910',
        'tag': '公司',
        'postcode': '722406',
        'is_default': "False",
      }
    ]
  }
}
```

修改地址

收获地址管理

* 收货人：

* 省 份：

北

* 详细地址：

建

* 手机号码：

邮政编码：

地址名称：

tag:

公司

receiver:

校长

address:

广东省深圳市龙华区民治大道100号

receiver_mobile:

18667018590

确定

取消

保存收货人信息

请求方式：PUT

请求参数：JSON

参数	参数类型	含义	备注
tag	char(10)	地址标签	必填
receiver_phone	char(11)	收件人电话	必填
receiver	char(7)	收件人姓名	必填
Address	char(100)	收件人地址	必填

```
{
  'receiver': '校长',
  'receiver_phone': '18667018590',
  'address': '广东省深圳市龙华区民治大道100号',
  'tag': '公司',
}
```

返回值：JSON

响应格式：

```
# 正常响应
{
  "code": 200,
  "data": {
    "addressList": [
      {
```

```
'id':123456, # 地址id
'address':'广东省深圳市龙华区嘉熙业广场1155室', # 地址
'receiver':'达内科技', # 收货人
'receiver_mobile':'12345678901', # 联系电话
'tag':'家', # 地址标签
'postcode':'722494', #
'is_default':"True",
},
{
  'id':123457,
  'address':'广东省深圳市龙华区嘉熙业广场1142室',
  'receiver':'达内科技',
  'receiver_mobile':'12345678910',
  'tag':'公司',
  'postcode':'722406',
  'is_default':"False",
}
]
}
}

# 异常响应
{
  "code": 200,
  "data": {
    "addresslist": []
  }
}
```

设为默认：

设为默认地址：

URL： `http://127.0.0.1:8000/v1/users/address/<username>/default`

请求类型：**POST**

参数	参数类型	含义	备注
id	char(10)	地址ID	必填

```
{
  'id':'xxxxxxx'
}
```

返回值：JSON

响应格式：

```
{
  "code":200,
  "data":{
    "addressList":[
      {
        'id':123456, # 地址id
        'address':'广东省深圳市龙华区嘉熙业广场1155室', # 地址
        'receiver':'达内科技', # 收货人
        'receiver_mobile':'12345678901', # 联系电话
        'tag':'家', # 地址标签
        'postcode':'722494', #
        'is_default':"True",
      },
      {
        'id':123457,
        'address':'广东省深圳市龙华区嘉熙业广场1142室',
        'receiver':'达内科技',
        'receiver_mobile':'12345678910',
        'tag':'公司',
        'postcode':'722406',
        'is_default':"False",
      }
    ]
  }
}
```

附录：用户模块异常状态码参考：

状态码	响应信息	原因短语
200	正常	OK
10100	无效参数	Invalid parameters
10101	无法获取用户名，需要登陆。	User not logged in
10102	设置新密码两次输入不一致	Inconsistent passwords
10103	旧密码错误	Old password error
10104	用户查询出错	User query error
10105	存储验证码失败	Storage authentication code failed
10106	验证码失效，请重新发送	The validation code is invalid. Please send it again.
10107	无法获取用户旧密码	Unable to retrieve old password

10108	无效的密码	Please enter your password
10109	确认密码有误	Confirm that the password is incorrect
10110	两次密码不一致	Password Inconsistencies
10111	旧密码错误	Old password error!
10112	验证链接失效	Verify that the link is invalid and resend it
10113	激活链接参数有误	Error activating link parameters
10114	地址查询出错	Error in Address Query
10115	无效的收件人	invalid recipients
10116	无效的地址	Invalid address
10117	无效的电话号码	Invalid phone number
10118	无效的邮编	Invalid postcode
10119	无效的地址标签	Invalid tag
10120	地址存储异常	Address storage exception
10121	地址查询异常	Get address exception
10122	无法获取地址ID	Get address id error
10123	修改地址参数有误	Error in address modification parameters
10124	无法获取微博登陆页	Cant get weibo login page
10125	无法获取微博token	Unable to get weibo token
10126	无效的邮箱	Invalid email
10127	无效的用户名	Invalid username
10128	创建用户失败	Create user failed
10129	无效的微博token	Invalid access token
10130	关联微博失败	Related to weibo failed
10131	无效的手机号	Invalid phone number
201	微博账号未绑定本地用户	access_token used
10131	重复请求	Repeated requests
10132	获取验证码异常	Verify Code Error
10133	新旧密码重复	Use Different Password

Celery-消息任务队列

Celery 是一个 基于python开发的分布式异步消息任务队列，通过它可以轻松的实现任务的异步处理，在我们这个项目中会用到使用celery进行发送用户的 **邮箱激活邮件、短信验证码、邮箱验证码**：

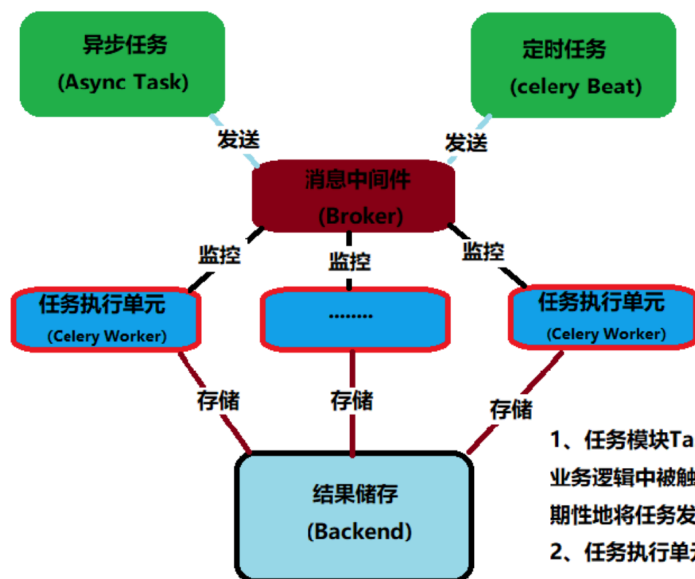
- 你想对100台机器执行一条批量命令，可能会花很长时间，但你不让你的程序等着结果返回，而是给你返回一个任务ID,你过一段时间只需要拿着这个任务id就可以拿到任务执行结果，在任务执行ing进行时，你可以继续做其它的事情。
- 你想做一个定时任务，比如每天检测一下你们所有客户的资料，如果发现今天是客户的生日，就给他发个短信祝福

Celery 在执行任务时需要通过一个消息中间件来接收和发送任务消息，以及存储任务结果，一般使用 rabbitMQ 或者Redis,本项目中主要使用redis。

Celery有以下优点：

- 简单：一旦熟悉了celery的工作流程后，配置和使用还是比较简单的
- 高可用：当任务执行失败或执行过程中发生连接中断，celery 会自动尝试重新执行任务
- 快速：一个单进程的celery每分钟可处理上百万个任务
- 灵活：几乎celery的各个组件都可以被扩展及自定义

工作原理



- 1、任务模块Task包含异步任务和定时任务。其中，异步任务通常在业务逻辑中被触发并发送往消息队列，而定时任务由Celery Beat进程周期性地将任务发送往消息队列；
- 2、任务执行单元Worker实时监视消息队列获取队列中的任务执行；
- 3、Worker执行完任务后将结果保存在Backend中；

基于redis实现的Celery

- 安装Celery: `pip3 install celery`
- 后台启动redis服务 `redis-server`

创建celery任务文件夹

```
celery_tasks
|
|----main.py    # 主模块
|---- user_tasks.py  # 任务执行模块
```

settings.py

```
CACHES = {
    "default": {
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "redis://127.0.0.1:6379",
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
        }
    },
    # 所有的验证有关的参数都存在redis 2号数据库。
    "verify_email": {
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "redis://127.0.0.1:6379/2",
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
        }
    },
}

# redis相关配置
# 官网定义的变量名:
# https://docs.celeryproject.org/en/latest/userguide/configuration.html#task-
# result-backend-settings
BROKER_URL = 'redis://127.0.0.1/14'
CELERY_RESULT_BACKEND = 'redis://127.0.0.1/15'
```

main.py

```
# -*- coding:utf-8 -*-
import os
from celery import Celery

# 使用Django的环境去启动
if not os.getenv('DJANGO_SETTINGS_MODULE'):
    os.environ['DJANGO_SETTINGS_MODULE'] = 'dadashop.settings'

celery_app = Celery('dadashop')
# 导入celery配置
celery_app.config_from_object('celery_tasks.settings')
# 自动注册celery任务
```

```
celery_app.autodiscover_tasks(['celery_tasks.user_tasks'])
```

users_tasks.py

将所有的任务写到同一个函数中，根据sendtype的不同来选择不同的发送类型任务

```
# -*- coding:utf-8 -*-
from celery_tasks.main import celery_app
from django.core.mail import send_mail
from django.conf import settings
from utils.SendTemplateSMS import send_template_SMS

@celery_app.task(name='send_verify')
def send_verify(email=None, code=None, email_code=None, phone=None,
verify_url=None, sendtype=None):
    """
    发送邮件
    :param receiver : 收件人, code 邮箱验证码
    :return:None
    发送类型为0:    则发送邮箱验证码。
    发送类型为1:    则发送邮箱激活链接
    发送类型为2:    则发送短信验证码
    """
    if sendtype == 0:
        # 发送
        subject = '达达商城密码找回邮件'
        html_message = '<p>尊敬的用户您好! </p>' \
            '<p>欢迎注册使用达达商城。</p>' \
            '<p>您的邮箱为: %s, 您的邮箱验证码为: </p>' \
            '<b>%s</b><p>10分钟之内有效。</p>' % (email, email_code)
        send_mail(subject, '', settings.EMAIL_FROM, [email],
html_message=html_message)
    elif sendtype == 1:
        # 发送邮箱激活链接
        subject = '达达商城激活邮件'
        html_message = '<p>尊敬的用户您好! </p>' \
            '<p>欢迎注册使用达达商城。</p>' \
            '<p>您的邮箱为: %s, 请点击此链接激活您的邮箱(30分钟有效): </p>' \
            '<p><a href="%s" target="_blank"><b>请点击激活</b></a>'
        send_mail(subject, '', settings.EMAIL_FROM, [email],
html_message=html_message)
    elif sendtype == 2:
        # 发送短信验证码
        try:
            send_template_SMS(phone, {3, code}, 1)
        except Exception as e:
```

```
print(e)
```

解释说明：

在不同的好事任务中我们只需要调用相应的函数的时候加上 `函数名.delay(*args,**kwarg)`,就可以通过 celery来执行相应的任务。

例如：

```
# first
class SendSmsCodeView(View):
    """
    用户找回密码视图处理函数：
    分为三步：
    1.验证邮箱，并且发送邮件验证码
    2.验证邮件验证码，
    3.验证码验证成功，修改密码
    """

    def post(self, request, username):
        data = request.POST
        email = data.get('email', None)
        .....
        .....
        # 在此处调用发送邮箱验证码
        send_verify.delay(email=email, email_code=email_code, sendtype=0)
        return JsonResponse({'code': 200, 'data': {'message': 'OK'}})

# second
class Users(View):

    def post(self, request):
        ...
        ...

        # 发送用户激活链接
        code_str = username + '.' + email
        # 生成激活链接：
        active_code = base64.b64encode(code_str.encode(encoding='utf-8')).decode('utf-8')
        redis_conn = get_redis_connection('verify_email')
        redis_conn.set("email_active_%s" % email, active_code)
        verify_url = 'http://127.0.0.1:8080/templates/web/active.html?code=%s&username=%s' % (active_code, username)
        token = make_token(username)
        result = {'code': 200, 'username': username, 'data': {'token': token.decode()}}
        # 发送用户激活链接
        send_verify.delay(email=email, verify_url=verify_url, sendtype=1)
```

```

        return JsonResponse(result)

# third
class SmScodeView(View):
    """
    实现短信验证码功能
    """
    def post(self, request):
        """
        短信测试:
        :param request:
        :return:
        """
        ....
        ....
        # 发送短信验证码
        send_verify.delay(phone=phone, code=code, sendtype=2)
        return JsonResponse({'code': 200, 'data': {'message': 'OK'}})

```

Donations

This project relies on your generous donations.

If you enjoy this, please consider becoming our backer or our sponsor to ensure Dadashop's future. alipay account: 18667018590.

