

《达达商城》购物车API文档说明

一、购物车模块概述

二、事件定义

- 1. 添加购物车
- 2. 查询购物车
- 3. 删除购物车
- 4. 购物车商品
- 5. 合并购物车

三、API接口说明

- 1. 添加购物车
 - 1.1 事件触发
 - 1.2 触发结果
 - 1.3 API说明
- 2. 查询购物车
 - 2.1 事件触发
 - 2.2 触发结果
 - 2.3 API说明
- 3. 删除购物车数据
 - 3.1 事件触发
 - 3.2 触发结果
 - 3.3 API说明
- 4. 购物车页面商品数量增加
 - 4.1 事件触发
 - 4.2 触发结果
 - 4.3 API说明
- 5. 购物车页面商品数量减少
 - 5.1 事件触发
 - 5.2 触发结果
 - 5.3 API说明
- 6. 购物车商品单选
 - 6.1 事件触发
 - 6.2 触发结果
 - 6.3 API说明
- 7. 购物车商品取消单选
 - 7.1 事件触发
 - 7.2 触发结果
 - 7.3 API说明
- 8. 购物车商品全选
 - 8.1 事件触发
 - 8.2 触发结果
 - 8.3 API说明
- 9. 购物车商品取消全选
 - 9.1 事件触发
 - 9.2 触发结果
 - 9.3 API说明

《达达商城》购物车API文档说明

一、购物车模块概述

购物车模块主要提供用户本产品的基础。购物车模块提供一下功能：

添加购物车

- 查询购物车
- 删除购物车
- 购物车商品数量添加/减少
- 购物车商品单选/取消单选
- 购物车商品全选/取消全选

二、事件定义

1. 添加购物车

用户未登陆状态下进入商品详情页，选择规格和数量，将商品加入到购物车，数据将保存到前端的浏览器的本地存储里。

用户登陆状态下进入商品详情页，选择规格和数量，将商品加入到购物车，前端会通过ajax进行与后端交互。

2. 查询购物车

用户未登录状态下进入购物车页面，前端通过浏览器的本地存储保存的数据展示给用户。

用户登录状态下进入购物车页面，后端通过redis存好的商品数据，传给前端展示给用户。

3. 删除购物车

用户进入购物车页面，点击购物车某个商品删除按钮，会有一个弹框显示确定删除和不删除，点击确定删除按钮，商品会被删除。

4. 购物车商品

商品数量增加和减少：用户进入购物车页面，点击商品“+”按钮，将商品数量增加1个。点击商品“-”按钮，将商品数量减少1个。

商品单选和取消单选：用户进入购物车页面，默认进入购物车页面单选是“√”，取消单件商品点击单选按钮，按钮显示空，选择单件商品点击单选按钮，按钮显示“√”。

商品全选和取消全选：用户进入购物车页面，默认进入购物车页面全选是“√”，取消全部商品点击全选按钮，按钮显示空，选择全部商品点击全选按钮，按钮显示“√”。

5. 合并购物车

用户在登陆时，创建token，会将用户未登陆状态下浏览器本地存储的购物车商品数据传给后端，后端通过redis存储的商品数量，做数量比较将商品数量最多的存入redis，返还给前端显示。

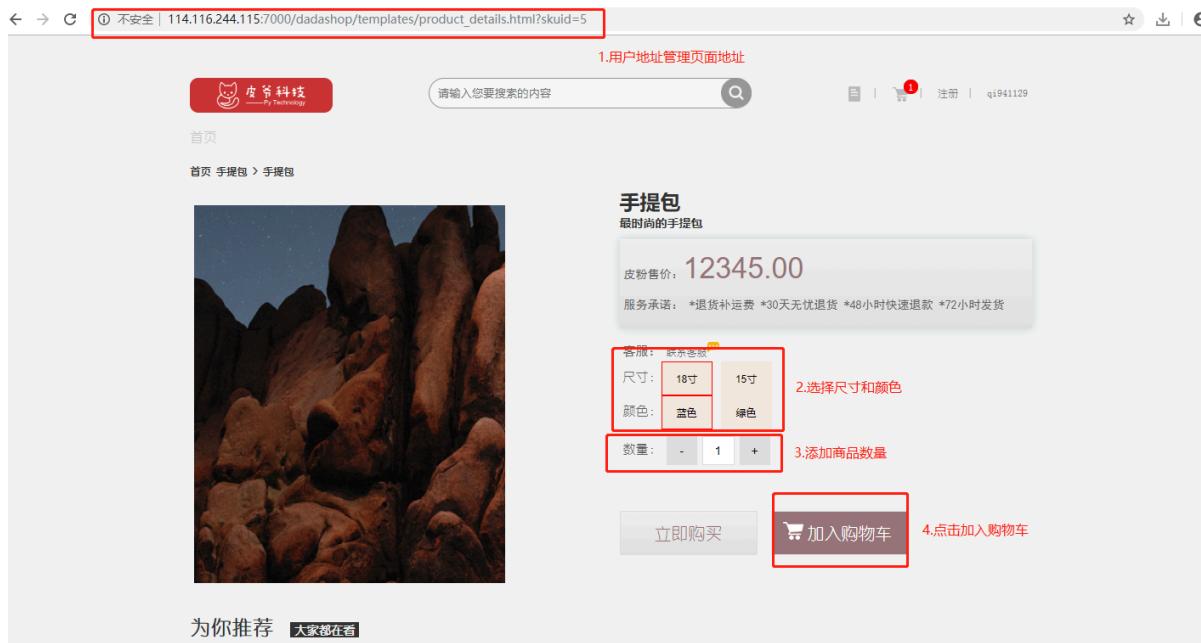
三、API接口说明

1. 添加购物车

在登录的状态下添加购物车

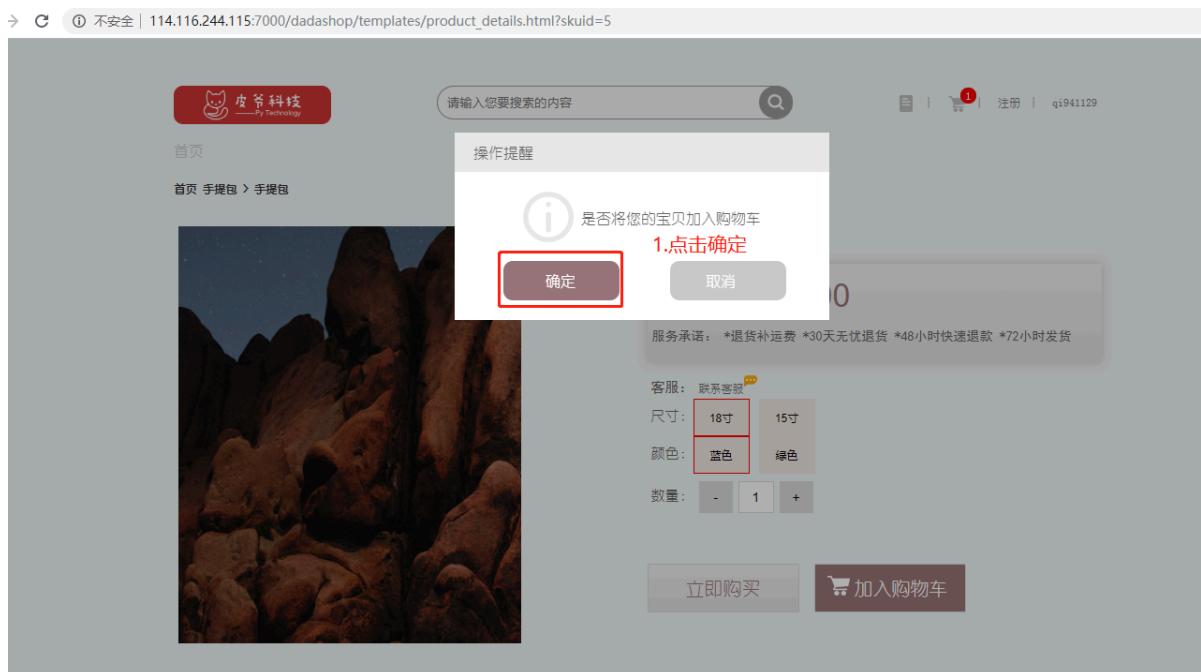
1.1 事件触发

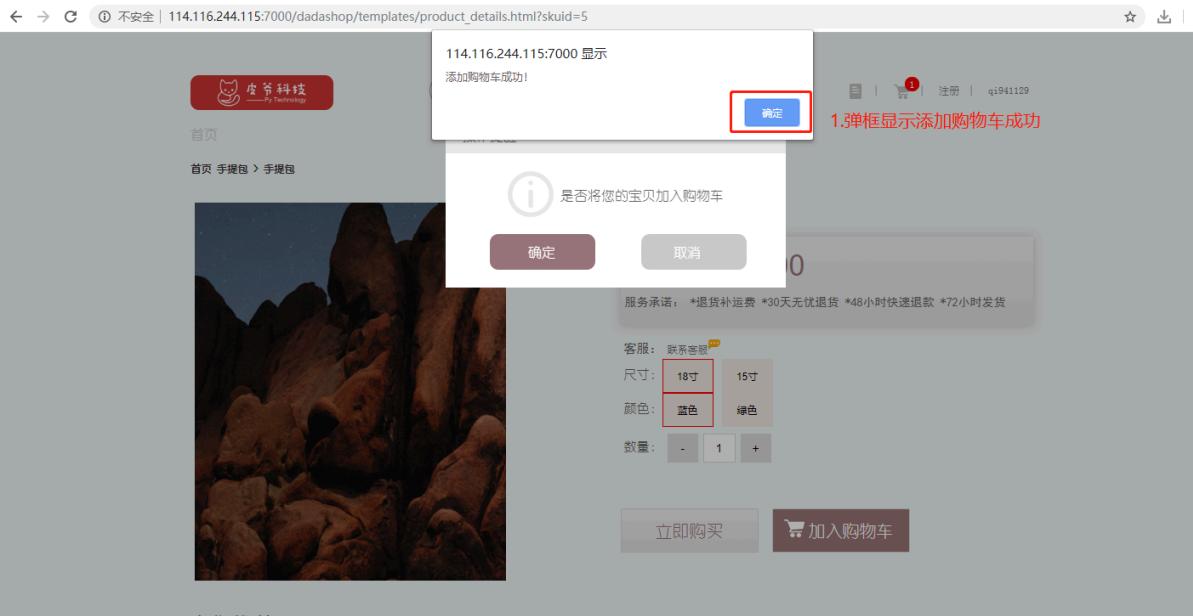
页面名：product_details.html



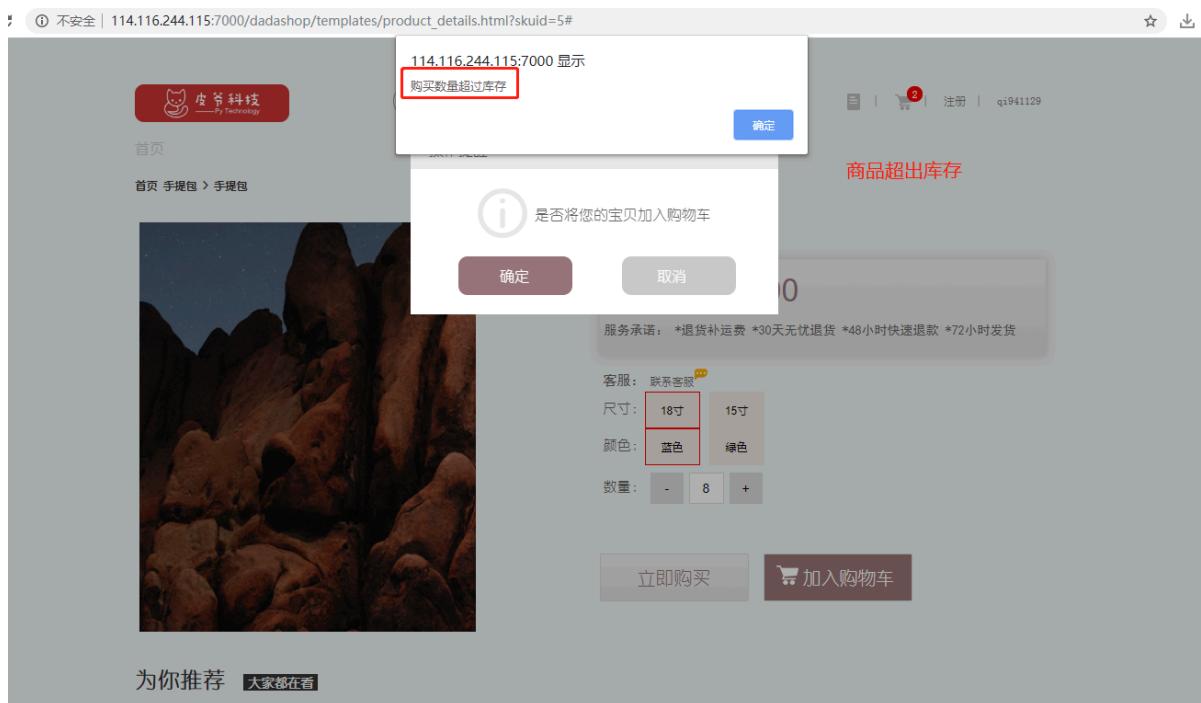
1.2 触发结果

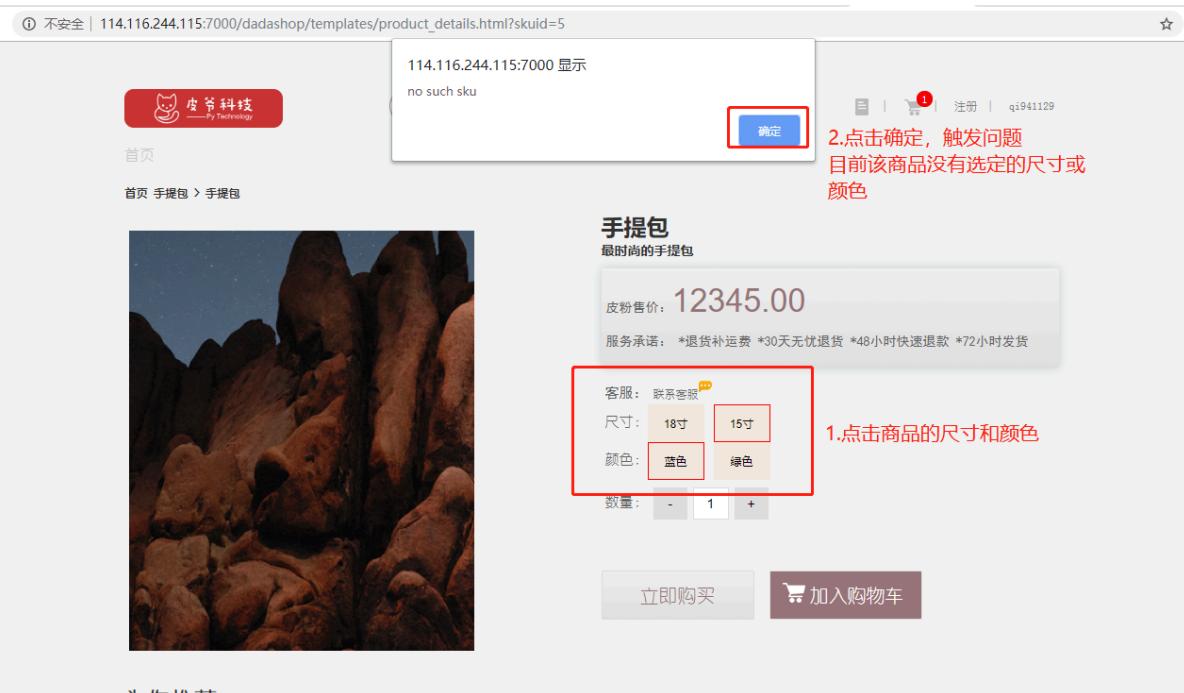
触发成功





触发失败:





1.3 API说明

- URL

`http://127.0.0.1:8000/v1/carts/<username>`

- 请求方式

POST

- 请求参数

JSON

参数	类型	是否必须	说明
sku_id	int	是	商品sku_id
count	int	是	单个商品数量

- 返回值

JSON

- 响应格式

```
# 正确响应:
{
    'code': 200,
    'data':
    {
        'carts_count': count
    },
    'base_url': PIC_URL
}
```

```
# 错误响应:
{
    "code":xxx,
    "error":xxx
}
```

```
}
```

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
base_url	图片存储路径	char	成功返回图片存储路径

base_url 示例

<http://127.0.0.1:8000/media/>

data 参数信息

参数	类型	是否必须	说明
carts_count	int	是	购物车里数量长度

状态码参数：

状态码	响应信息	原因短句
200	正常	OK
30101	未找到商品	SKU query error
30102	传参不正确	Incorrect pass of reference
30103	购买数量超过库存	exceeds the inventory
30104	未找到用户	User query error

2. 查询购物车

2.1 事件触发

页面名：**index.html**

页面名：**cart.html**

① 不安全 | 114.116.244.115:7000/dadashop/templates/index.html

1.进入首页地址

请输入您要搜索的内容

首页

2.点击购物车

[手提包更多>>](#)

2.2 触发结果

① 不安全 | 114.116.244.115:7000/dadashop/templates/cart.html

1.进入购物车地址

已选 2 件商品 合计(不含运费): 271590元

2.已经加入购物车的商品

去结算

2.3 API说明

URL: 127.0.0.1:8000/v1/carts/<username>

请求方式: GET

请求参数: 无

返回值: JSON

响应格式:

```

# 正常响应:
{
    'code': 200,
    'data': skus_list,
    'base_url': PIC_URL
}

# 异常响应
{
    "code":xxx,
    "error":xxxx
}

```

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
base_url	图片存储路径	char	成功返回图片存储路径

base_url 示例
<http://127.0.0.1:8000/media/>

skus_list参数信息

参数	类型	是否必须	说明
id	int	是	商品sku_id
name	str	是	商品名称
count	int	是	单个商品数量
default_image_url	str	是	商品默认图片路径
price	decimal	是	商品单价
sku_sale_attr_name	list	是	商品属性
sku_sale_attr_val	list	是	商品属性值
selected	int	是	商品的选中状态 (0未选中, 1选中)

```

# skus_list中数据示例
[
    {
        "id":"",
        "name":"",
        "count":"",
        "selected":"",
        "default_image_url":"",
        "price": ""
    }
]

```

```

        "sku_sale_attr_name":[] ,
        "sku_sale_attr_val":[]
    },
{
"":...
]
}

# 注:
sku_sale_attr_name = ["尺寸", "颜色"]
sku_sale_attr_val = ["18寸", "蓝色"]

```

状态码参数

状态码	响应信息	原因短语
200	正常	OK
30104	未找到用户	User query error

3. 删除购物车数据

3.1 事件触发

页面名: cart.html

1.点击删除键

3.2 触发结果

1.点击删除

3.3 API说明

URL: <http://127.0.0.1:8000/v1/carts/<username>>

请求方式: DELETE

请求参数

```
# 请求参数示例:  
{  
    "sku_id":1001  
}
```

参数	类型	是否必须	说明
sku_id	int	是	商品sku_id

返回值: json

响应格式:

```
# 正常响应:  
{  
    'code': 200,  
    'data':  
    {  
        'carts_count': count  
    },  
    'base_url': PIC_URL  
}  
  
# 异常响应  
{  
    "code":xxx,  
    "error":xxxx  
}
```

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
base_url	图片存储路径	char	成功返回图片存储路径

base_url 例子

<http://127.0.0.1:8000/media/>

data参数信息

参数	类型	是否必须	说明
carts_count	int	是	购物车里数量长度

状态码参考

状态码	响应信息	原因短句
200	正常	OK
30101	未找到商品	SKU query error
30104	未找到用户	User query error

4. 购物车页面商品数量增加

4.1 事件触发

页面名：cart.html

① 不安全 | 114.116.244.115:7000/dadashop/templates/cart.html

1.点击购物车地址

皮爷科技

请输入您要搜索的内容

首页

购物车

已选 2 件商品 合计(不含运费): 12345元

全选	商品	单价(元)	数量	金额	操作
<input checked="" type="checkbox"/>	我们的时代 1v 颜色: 18寸 1v 尺寸: 红色	原价: ￥12345.00	<input type="button" value="-"/> <input checked="" type="button" value="1"/> <input type="button" value="+"/>	￥ 12345	删除

2.点击添加该商品的数量

已选2件商品 合计(不含运费): 12345元

去结算

4.2 触发结果

The screenshot shows a shopping cart interface. At the top, there's a header with a logo, a search bar, and user information. Below the header, the page title is "购物车" (Shopping Cart). It displays a list of items with columns for "商品" (Product), "单价(元)" (Unit Price), "数量" (Quantity), and "金额" (Amount). A red box highlights the quantity input field for the first item, which is set to "2". Below the table, a message says "该商品的数量增加" (The quantity of this product has increased). At the bottom right, there's a button labeled "去结算" (Go to Settlement).

4.3 API说明

URL: <http://127.0.0.1:8000/v1/carts/>

请求方式: PUT

请求参数: JSON

参数	类型	是否必须	说明
sku_id	int	是	商品sku_id
count	int	是	单个商品数量
state	str	是	判断事件状态

请求示例 `{'sku_id':1001,count:1,state:'add'}`

返回值: JSON

响应格式:

正常响应: `{'code': 200, 'data': skus_list, 'base_url': PIC_URL}`

异常响应`{"code":xxx,"error":xxxx}`

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
base_url	图片存储路径	char	成功返回图片存储路径

base_url 例子http://127.0.0.1:8000/media/

skus_list参数信息

参数	类型	是否必须	说明
id	int	是	商品sku_id
name	str	是	商品名称
count	int	是	单个商品数量
default_image_url	str	是	商品默认图片路径
price	decimal	是	商品单价
sku_sale_attr_name	list	是	商品属性
sku_sale_attr_val	list	是	商品属性值
selected	int	是	商品的选中状态 (0未选中, 1选中)

#skus_list中数据示例

```
[  
  {  
    "id": "",  
    "name": "",  
    "count": "",  
    "selected": "",  
    "default_image_url": "",  
    "price": "",  
    "sku_sale_attr_name": [],  
    "sku_sale_attr_val": []  
  },  
  {"..."}  
]  
  
#注: sku_sale_attr_name = ["尺寸", "颜色"] sku_sale_attr_val = ["18寸", "蓝色"]
```

状态码参考

状态码	响应信息	原因短语
200	正常	OK
30101	未找到商品	SKU query error
30104	未找到用户	User query error

5. 购物车页面商品数量减少

5.1 事件触发

页面名: cart.html

The screenshot shows a web browser displaying a shopping cart page at the URL 114.116.244.115:7000/dadashop/templates/cart.html. The page title is '1.进入购物车地址'. At the top right, there are user icons for login and a user ID (q1941129). A red box highlights the URL bar. Below the header is a search bar with placeholder text '请输入您要搜索的内容' and a magnifying glass icon. The main content area is titled '购物车' (Shopping Cart) and shows a table of items. The first item is '我们的时代' (Our Era), with a thumbnail image, a checked checkbox, the text 'lv 颜色: 18寸', and '1v 尺寸: 红色'. To its right are columns for '单价(元)' (Unit Price), '数量' (Quantity), and '金额' (Amount). The quantity input field contains the value '5', which is highlighted with a red box. The total amount for this item is '¥ 61725'. A '删除' (Delete) button is also visible. At the bottom of the cart table, it says '已选 2 件商品 合计(不含运费): 61725 元'. A large red box labeled '2.点击该商品的“-”' (Click the minus sign of this product) points to the minus button in the quantity input field.

5.2 触发结果

The screenshot shows the same shopping cart page after the quantity has been decreased. The URL remains 114.116.244.115:7000/dadashop/templates/cart.html. The page title is now '2.商品数量减少' (Product quantity reduced). The main content area shows the same table as before, but the quantity input field for the first item now contains '4', which is highlighted with a red box. The total amount for this item is '¥ 49380'. A red box labeled '该商品数量减少' (The product quantity has been reduced) points to the quantity input field. The footer of the page still shows '已选 2 件商品 合计(不含运费): 49380 元'.

5.3 API说明

URL: 127.0.0.1:8000/v1/carts/

请求方式: PUT

请求参数: JSON

参数	类型	是否必须	说明
sku_id	int	是	商品的sku_id
count	int	是	单个商品数量
state	str	是	判断事件状态

```
# 请求示例:
{
    'sku_id':1001,
    count:1,
    state:'del'
}
```

返回值: JSON

响应格式:

```
# 正常响应:
{
    'code': 200,
    'data': skus_list,
    'base_url': PIC_URL
}

# 异常响应
{
    "code":xxx,
    "error":xxxx
}
```

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
base_url	图片存储路径	char	成功返回图片存储路径

base_url 例子<http://127.0.0.1:8000/media/>

skus_list参数信息

参数	类型	是否必须	说明
id	int	是	商品sku_id
name	str	是	商品名称
count	int	是	单个商品数量
default_image_url	str	是	商品默认图片路径
price	decimal	是	商品单价
sku_sale_attr_name	list	是	商品属性
sku_sale_attr_val	list	是	商品属性值
selected	int	是	商品的选中类型 (0未选中, 1选中)

```

# skus_list中数据示例
[
  {
    "id":"",
    "name":"",
    "count":"",
    "selected":"",
    "default_image_url":"",
    "price":"",
    "sku_sale_attr_name":[] ,
    "sku_sale_attr_val":[]
  },
  {"":""...}
]

# 注:
sku_sale_attr_name = ["尺寸", "颜色"]
sku_sale_attr_val = ["18寸", "蓝色"]

```

状态码参数

状态码	响应信息	原因短语
200	正常	OK
30101	未找到商品	SKU query error
30104	未找到用户	User query error

6. 购物车商品单选

6.1 事件触发

页面名：cart.html

1.进入购物车地址

2.点击某个商品点击选择

商品	单价(元)	数量	金额	操作
手提包 尺寸：18寸 颜色：蓝色	¥12345.00	- +	¥ 88415	删除
我们的时代 1v 尺寸：红色	¥12345.00	- +	¥ 49380	删除

已选1件商品 合计(不含运费): 49380元 去结算

6.2 触发结果

The screenshot shows a shopping cart interface with the following details:

商品	单价(元)	数量	金额	操作
手提包 尺寸: 18寸 颜色: 蓝色	¥12345.00	- 7 +	¥ 86415	删除
我们的时代 1v 颜色: 18寸 1v 尺寸: 红色	¥12345.00	- 4 +	¥ 49380	删除

Total items: 2, Total amount: 135795元.

6.3 API说明

URL: 127.0.0.1:8000/v1/carts/

请求方式: PUT

请求参数: JSON

参数	类型	是否必须	说明
sku_id	int	是	购物车显示商品的sku_id
state	str	是	判断事件状态

```
# 请求示例
{
    'sku_id': 1001,
    state: 'select'
}
```

返回值: JSON

响应格式

```
# 正常响应:
{
    'code': 200,
    'data': skus_list,
    'base_url': PIC_URL
}

# 异常响应
{
    "code": xxx,
    "error": xxxx
}
```

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

字段	含义	类型	备注
data	具体数据	dict	与error二选一
error	错误信息	char	与data二选一
base_url	图片存储路径	char	成功返回图片存储路径

base_url 例子http://127.0.0.1:8000/media/

skus_list参数信息

参数	类型	是否必须	说明
id	int	是	商品sku_id
name	str	是	商品名称
count	int	是	单个商品数量
default_image_url	str	是	商品默认图片路径
price	decimal	是	商品单价
sku_sale_attr_name	list	是	商品属性
sku_sale_attr_val	list	是	商品属性值
selected	int	是	商品的选中状态 (0未选中, 1选中)

```
# skus_list中数据示例
[
    {
        "id":"",
        "name":"",
        "count":"",
        "selected":"",
        "default_image_url":"",
        "price":"",
        "sku_sale_attr_name":[] ,
        "sku_sale_attr_val":[]
    },
    {"":""...}]
# 注:
sku_sale_attr_name = ["尺寸", "颜色"]
sku_sale_attr_val = ["18寸", "蓝色"]
```

状态码参考

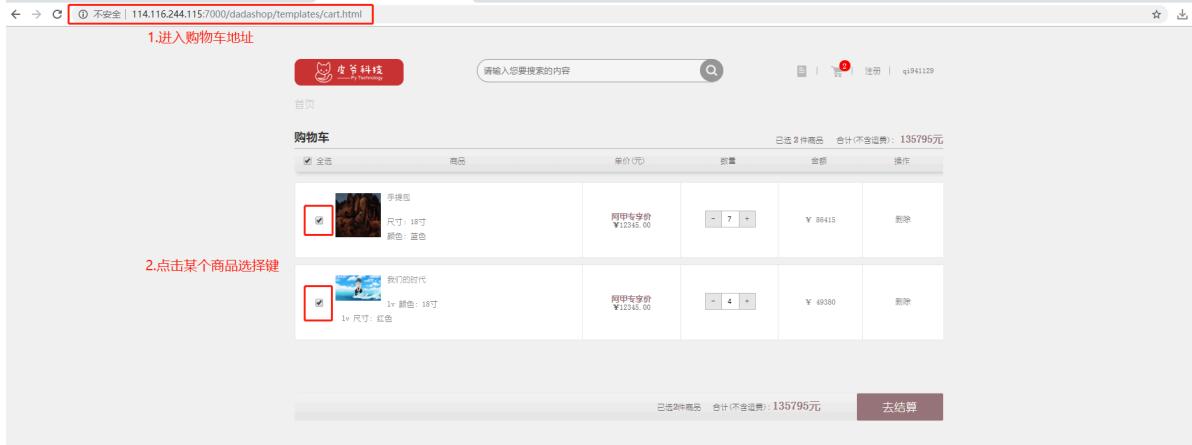
状态码	响应信息	原因短语
200	正常	OK
30101	未找到商品	SKU query error

状态码	响应信息	原因短语
30104	未找到用户	User query error

7. 购物车商品取消单选

7.1 事件触发

页面名：cart.html



The screenshot shows a shopping cart interface. At the top, there's a header with a logo, search bar, and user information. Below it is a table for the shopping cart. The first row of the table has a checkbox with a red border around it, indicating it's selected. The second row also has a selected checkbox. The table includes columns for product image, name, unit price, quantity, total price, and delete button. At the bottom, there's a summary bar showing '已选 2 件商品' and '合计(不含运费): 135795元'.

7.2 触发结果



This screenshot shows the same shopping cart page after the selection has been cleared. The checkboxes for both items are now unselected, with a gray border instead of a red one. The rest of the interface remains the same, including the table structure and the summary at the bottom.

7.3 API说明

URL: 127.0.0.1:8000/v1/carts/

请求方式: PUT

请求参数: JSON

参数	类型	是否必须	说明
state	str	是	判断事件状态
sku_id	int	是	购物车显示商品的sku_id

```
# 请求示例
{
    'sku_id': 1001,
    state: 'unselect'
}
```

返回值:JSON

响应格式:

```
# 正常响应:
{
    'code': 200,
    'data': skus_list,
    'base_url': PIC_URL
}

# 异常响应
{
    "code": xxx,
    "error": xxxx
}
```

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
base_url	图片存储路径	char	成功返回图片存储路径

base_url 例子http://127.0.0.1:8000/media/

skus_list参数信息

参数	类型	是否必须	说明
id	int	是	商品sku_id
name	str	是	商品名称
count	int	是	单个商品数量
default_image_url	str	是	商品默认图片路径
price	decimal	是	商品单价
sku_sale_attr_name	list	是	商品属性
sku_sale_attr_val	list	是	商品属性值
selected	int	是	商品的选中状态 (0未选中, 1选中)

```

# skus_list中数据示例
[
  {
    "id":"",
    "name":"",
    "count":"",
    "selected":"",
    "default_image_url":"",
    "price":"",
    "sku_sale_attr_name":[] ,
    "sku_sale_attr_val":[]
  },
  {"": "..."}
]

# 注:
sku_sale_attr_name = ["尺寸", "颜色"]
sku_sale_attr_val = ["18寸", "蓝色"]

```

状态码参考

状态码	响应信息	原因短语
200	正常	OK
30101	未找到商品	SKU query error
30104	未找到用户	User query error

8. 购物车商品全选

8.1 事件触发

页面名: cart.html

8.2 触发结果



8.3 API说明

URL: 127.0.0.1:8000/v1/carts/

请求方式: PUT

请求参数: JSON

参数	类型	是否必须	说明
state	str	是	判断事件状态

```
# 请求示例:  
{  
    state: 'selectall'  
}
```

返回值: JSON

响应格式

```
# 正常响应:  
{  
    'code': 200,  
    'data': skus_list,  
    'base_url': PIC_URL  
}  
  
# 异常响应  
{  
    "code":xxx,  
    "error":xxxx  
}
```

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

字段	含义	类型	备注
error	错误信息	char	与data二选一
base_url	图片存储路径	char	成功返回图片存储路径

base_url 例子<http://127.0.0.1:8000/media/>

skus_list参数信息

参数	类型	是否必须	说明
id	int	是	商品sku_id
name	str	是	商品名称
cart_count	int	是	单个商品数量
default_image_url	str	是	商品默认图片路径
price	decimal	是	商品单价
sku_sale_attr_name	list	是	商品属性
sku_sale_attr_val	list	是	商品属性值
selected	int	是	商品的选中状态 (0未选中, 1选中)

```
# skus_list中数据示例
[
  {
    "id":"",
    "name":"",
    "count":"",
    "selected":"",
    "default_image_url":"",
    "price":"",
    "sku_sale_attr_name":[] ,
    "sku_sale_attr_val":[]
  },
  {"":...}
]
# 注:
sku_sale_attr_name = ["尺寸", "颜色"]
sku_sale_attr_val = ["18寸", "蓝色"]
```

状态码参考

状态码	响应信息	原因短句
200	正常	OK
30104	未找到用户	User query error

9. 购物车商品取消全选

9.1 事件触发

页面名：cart.html



① 不安全 | 114.116.244.115:7000/dashshop/templates/cart.html

1.进入购物车地址

2.点击全选键

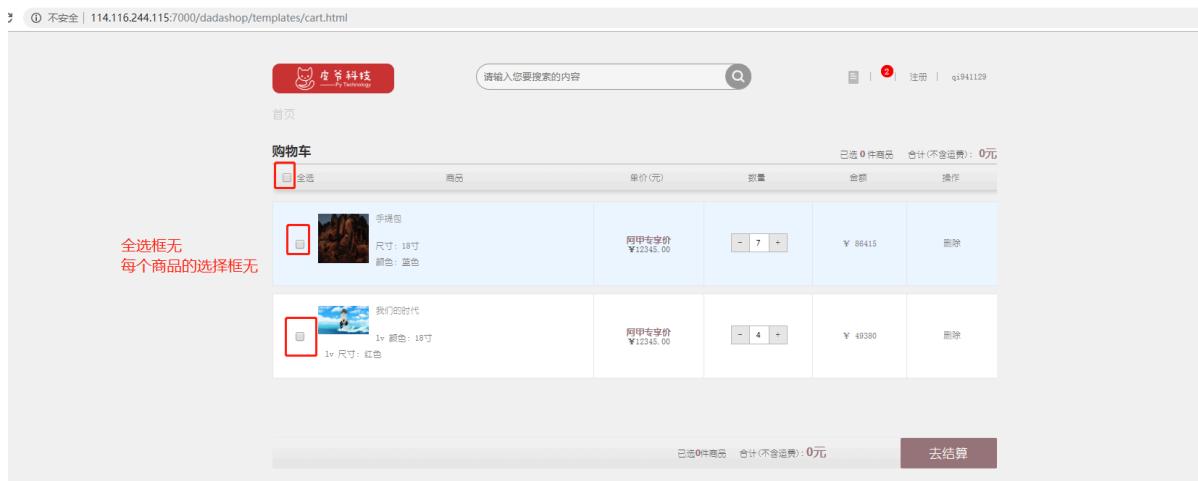
购物车

商品	单价(元)	数量	金额	操作
手提包 尺寸：18寸 颜色：蓝色	网甲专享价 ¥12345.00	- 7 +	¥ 86415	删除
我们的时代 1v 颜色：18寸 1v 尺寸：红色	网甲专享价 ¥12345.00	- 4 +	¥ 49380	删除

已选2件商品 合计(不含运费): 135795元

去结算

9.2 触发结果



① 不安全 | 114.116.244.115:7000/dashshop/templates/cart.html

全选框无 每个商品的选择框无

购物车

商品	单价(元)	数量	金额	操作
手提包 尺寸：18寸 颜色：蓝色	网甲专享价 ¥12345.00	- 7 +	¥ 86415	删除
我们的时代 1v 颜色：18寸 1v 尺寸：红色	网甲专享价 ¥12345.00	- 4 +	¥ 49380	删除

已选0件商品 合计(不含运费): 0元

去结算

9.3 API说明

URL: 127.0.0.1:8000/v1/carts/

请求方式: PUT

请求参数: JSON

参数	类型	是否必须	说明
state	str	是	判断事件状态

```
# 请求示例
{
    state: 'unselectall'
}
```

返回值: JSON

响应格式:

```

# 正常响应:
{
    'code': 200,
    'data': skus_list,
    'base_url': PIC_URL
}

# 异常响应
{
    "code":xxx,
    "error":xxxx
}

```

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
base_url	图片存储路径	char	成功返回图片存储路径

base_url 例子http://127.0.0.1:8000/media/

skus_list参数信息

参数	类型	是否必须	说明
id	int	是	商品sku_id
name	str	是	商品名称
count	int	是	单个商品数量
default_image_url	str	是	商品默认图片路径
price	decimal	是	商品单价
sku_sale_attr_name	list	是	商品属性
sku_sale_attr_val	list	是	商品属性值
selected	int	是	商品的选中状态 (0未选中,1选中)

```

# skus_list中数据示例
[
    {
        "id":"",
        "name":"",
        "count":"",
        "selected":"",
        "default_image_url":"",
        "price":"",
        "sku_sale_attr_name": []
    }
]
```

```
    "sku_sale_attr_val":[]  
},  
{""":""...}  
]  
  
# 注:  
sku_sale_attr_name = ["尺寸", "颜色"]  
sku_sale_attr_val = ["18寸", "蓝色"]
```

状态码参考

状态码	响应信息	原因短语
200	正常	OK
错误状态码	错误原因	error reason