

"/search/"

Search endpoint is used to search the products or vendors in the system.

Method: GET

Parameter	Description	Required/ Optional
search_string	string: String to be used in searching the products or vendors	Required
search_type	string: "vendor" for searching vendors, "product" for searching products in the system (Note: In the scope of this milestone you can only use product searching)	Required
frating_gte	int: Filter parameter for filtering products with ratings greater than or equal to the given value	Optional
fprice_lower	float: Filter parameter for filtering products with price greater than or equal to the given value, lower limit to the price	Optional
fprice_upper	float: Filter parameter for filtering products with price less than or equal to the given value, upper limit to the price	Optional
fvendor_name	string: Filter parameter for filtering products by the vendors	Optional
fbrand	string: Filter parameter for filtering products by the brands	Optional
fcategory	string: Filter parameter for filtering products by the categories	Optional
sort_by	=bestseller: sorting by best seller =newest: sorting by newest arrival =priceAsc: sorting by price in ascending order =priceDesc: sorting by price in descending order =numComments: sorting by number of comments	Optional

Example query:

http://3.232.20.250/search?search_string=apple&search_type=product

Response:

JSON object: List of products with product_info if search_type is product
List of vendors with vendor_info if search_type is vendor

Notes:

- If you don't use the filtering type, you should not use the parameter in the request

"/product/"

Product page endpoint is used to see the product with a given ID.

Method: **GET**

Parameter	Description	Required/ Optional
id	int: ID of the product	Required

Example query:

<http://3.232.20.250/product?id=1>

Response:

JSON object: detailed_product_info

Notes:

- If you don't use the filtering type, you should not use the parameter in the request

"/product/category/"

Category endpoint is used to see the products in the category.

Method: **GET**

Parameter	Description	Required/ Optional
name	string: Name of the category	Required
frating_gte	int: Filter parameter for filtering products with ratings greater than or equal to the given value	Optional
fprice_lower	float: Filter parameter for filtering products with price greater than or equal to the given value, lower limit to the price	Optional
fprice_upper	float: Filter parameter for filtering products with price less than or equal to the given value, upper limit to the price	Optional
fvendor_name	string: Filter parameter for filtering products by the vendors	Optional
fbrand	string: Filter parameter for filtering products by the brands	Optional
fcategory	string: Filter parameter for filtering products by the categories	Optional
sort_by	=bestseller: sorting by best seller =newest: sorting by newest arrival =priceAsc: sorting by price in ascending order =priceDesc: sorting by price in descending order =numComments: sorting by number of comments	Optional

Example query:

<http://3.232.20.250/product/category?name=Home>

Response:

JSON object: List of products with product_info

Notes:

- If you don't use the filtering type, you should not use the parameter in the request

"/product/add/"

Product adding endpoint is used by vendors to add a new product.

Method: POST

Parameter	Description	Required/Optional
name	string: Name of the product to be added	Required
category	string: Name of the category of the product to be added	Required
description	string: Description of the product to be added	Required
brand	string: Brand of the product to be added	Required
stock	int: Stock of the product to be added	Required
price	float: Price of the product to be added	Required
photo	file: Primary photo of the product to	Optional

Example query:

<http://3.232.20.250/product/add/>

Response:

HTTP Response: Success or fail messages

Notes:

- You should give the parameters in the form data.
- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/product/edit/"

Product editing endpoint is used by vendors to edit a product of their own.

Method: **POST**

Parameter	Description	Required/ Optional
id	int: ID of the product to be edited	Required
name	string: New name of the product to be edited	Optional
category	string: New name of the category of the product to be edited	Optional
description	string: New description of the product to be edited	Optional
brand	string: New brand of the product to be edited	Optional
stock	int: New stock of the product to be edited	Optional
price	float: New price of the product to be edited	Optional
photo	file: Optional photo of the product to be edited	Optional

Example query:

<http://3.232.20.250/product/edit/>

Response:

HTTP Response: Success or fail messages

Notes:

- You should give the parameters in the form data.
- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/product/delete/"

Product deleting endpoint is used by vendors to delete a product of their own by its id.

Method: **POST / DELETE**

Parameter	Description	Required/ Optional
id	int: ID of the product to be deleted	Required

Example query:

<http://3.232.20.250/product/delete/>

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/product/addphoto/"

Product photo adding endpoint is used by vendors to add a photo to a product of their own by its id.

Method: POST

Parameter	Description	Required/Optional
id	int: ID of the product to be edited	Required
photo	file: Photo to be added to the product	Required

Example query:

<http://3.232.20.250/product/addphoto/>

Response:

HTTP Response: Success or fail messages

Notes:

- You should give the parameters in the form data.
- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/product/deleteallphotos/"

Product delete all photos endpoint is used by vendors to delete all photos of a product of their own by its id.

Method: POST / DELETE

Parameter	Description	Required/Optional
id	int: ID of the product to be edited	Required

Example query:

<http://3.232.20.250/product/deleteallphotos/>

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/product/deletephoto/"

Product delete photos endpoint is used by vendors to delete photos of a product of their own by its id and the url of the photo to be deleted.

Method: POST / DELETE

Parameter	Description	Required/ Optional
id	int: ID of the product to be edited	Required
photo_url	string: Url of the photo to be deleted.	Required

Example query:

<http://3.232.20.250/product/deletephoto/>

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/shoppingcart/add/"

Adds given product to the shopping cart with given quantity for the authenticated customer. If quantity is not given, then it is set to 1.

Method: POST

Parameter	Description	Required/ Optional
product_id	int: id of product to add	Required
quantity	int: quantity to be added	Optional

Example query:

<http://3.232.20.250/shoppingcart/add/>

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

["/shoppingcart/increase/"](#)

Adds given product to the shopping cart if it doesn't exist, else increases its quantity.

Method: POST

Parameter	Description	Required/Optional
product_id	int: id of product to increase	Required

Example query:

<http://3.232.20.250/shoppingcart/increase>

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

["/shoppingcart/decrease/"](#)

Deletes the given product to the shopping cart if it's quantity is 1, decreases the quantity otherwise.

Method: POST

Parameter	Description	Required/Optional
product_id	int: id of product to decrease	Required

Example query:

<http://3.232.20.250/shoppingcart/decrease>

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/shoppingcart/delete/"

Deletes the given product (all of them with the given product_id) from the shopping cart for the authenticated customer.

Method: POST / DELETE

Parameter	Description	Required/ Optional
product_id	int: id of product to delete	Required

Example query:

<http://3.232.20.250/shoppingcart/delete>

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/shoppingcart/all/"

Returns the shopping cart for the authenticated customer.

Method: GET

Parameter	Description	Required/ Optional
-----------	-------------	-----------------------

Example query:

<http://3.232.20.250/shoppingcart/all>

Response:

JSON Response: List of products in the cart with their quantities.

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace
-

"/shoppinglist/createlist/"

Shopping list create list endpoint is used by customers to create a new shopping list with the given name

Method: **POST**

Parameter	Description	Required/Optional
list_name	string: Name of the new shopping list	Required

Example query:

<http://3.232.20.250/shoppinglist/createlist/>

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/shoppinglist/deletelist/"

Shopping list delete list endpoint is used by customers to delete a shopping list of their own with the given name

Method: **POST / DELETE**

Parameter	Description	Required/Optional
list_name	string: Name of the shopping list	Required

Example query:

<http://3.232.20.250/shoppinglist/deletelist/>

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/shoppinglist/addtolist/"

Shopping list add to list endpoint is used by customers to add a product to a shopping list with the given product id and list name

Method: POST

Parameter	Description	Required/Optional
list_name	string: Name of the shopping list	Required
product_id	int: ID of the product to be added to the list	Required

Example query:

<http://3.232.20.250/shoppinglist/addtolist/>

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token seperated with a whitespace

["/shoppinglist/deletefromlist/"](#)

Shopping list delete from list endpoint is used by customers to delete a product from a shopping list with the given product id and list name

Method: POST / DELETE

Parameter	Description	Required/Optional
list_name	string: Name of the shopping list	Required
product_id	int: ID of the product to be deleted from the list	Required

Example query:

<http://3.232.20.250/shoppinglist/deletefromlist/>

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token seperated with a whitespace

["/shoppinglist/getlists/"](#)

Shopping list get lists endpoint is used by customers to retrieve all the lists they created before

Method: GET

Parameter	Description	Required/ Optional
-----------	-------------	-----------------------

Example query:

<http://3.232.20.250/shoppinglist/getlists/>

Response:

JSON Response: List of strings with the list names i.e. ["mylist1", "mylist2"]

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

["/shoppinglist/products/"](#)

Shopping list get lists endpoint is used by customers to get all the products from a shopping list they created.

Method: GET

Parameter	Description	Required/ Optional
list_name	string: Name of the shopping list	Required

Example query:

<http://3.232.20.250/shoppinglist/products/>

Response:

JSON Response: List of products with product_info

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

["/order/create_orders/"](#)

Creates the order from the customer's shopping list.

Method: POST

Parameter	Description	Required/ Optional
-----------	-------------	-----------------------

Example query:

http://3.232.20.250/order/create_orders

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

["/order/set_delivered/"](#)

Set the given order as delivered. Authentication must belong to corresponding customer.

Method: POST

Parameter	Description	Required/ Optional
order_id	int: id of the corresponding order	Required

Example query:

http://3.232.20.250/order/set_delivered

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

["/order/set_delivery/"](#)

Sets the given orders as in delivery with estimated arrival time and cargo id, authentication must belong to the corresponding vendor.

Method: POST

Parameter	Description	Required/ Optional
order_id	int: id of the corresponding order	Required

cargo_id	string: id of the cargo	Required
days	int: number of days till arrival	Required

Example query:

http://3.232.20.250/order/set_delivery

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/order/cancel_order/"

Sets the given orders as cancelled.

Method: POST

Parameter	Description	Required/Optional
order_id	int: id of the corresponding order	Required

Example query:

http://3.232.20.250/order/cancel_order

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/order/get_orders/"

Returns the orders of given user, inner lists are the orders in the same shopping cart.

Method: GET

Parameter	Description	Required/Optional
-----------	-------------	-------------------

Example query:

http://3.232.20.250/order/get_orders/

Response:

JSON Response: List of orders nested into the corresponding carts at the time.

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/comment/"

Comment adding endpoint is used by customers to add a comment to a delivered product.

Method: POST

Parameter	Description	Required/Optional
product_id	int: ID of the product to which the comment will be added	Required
text	string: Comment text	Required
product_rating	int: Rating for the product	Required
vendor_rating	int: Rating for the vendor that sells the product	Required

Example query:

<http://3.232.20.250/comment>

Response:

HTTP Response: Success or fail messages

Notes:

- You should give the parameters in the form data.
- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/recommendation/recommended"

Recommended endpoint is used by customers to customized recommendations

Method: GET

Parameter	Description	Required/ Optional
-----------	-------------	-----------------------

Example query:

<http://3.232.20.250/recommendation/recommended>

Response:

JSON Response: List of products with product_info

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token seperated with a whitespace

"/recommendation/bestseller"

Bestseller endpoint returns 10 bestsellers of Tursu

Method: GET

Parameter	Description	Required/ Optional
-----------	-------------	-----------------------

Example query:

<http://3.232.20.250/recommendation/bestseller>

Response:

JSON Response: List of products with product_info

Notes:

"/recommendation/toprated"

Top rated endpoint returns 10 top rated products of Tursu

Method: GET

Parameter	Description	Required/ Optional
-----------	-------------	-----------------------

Example query:

<http://3.232.20.250/recommendation/toprated>

Response:

HTTP Response: List of products with product_info

Notes:

["/recommendation/newest"](#)

Newesr endpoint returns 10 newest added products of Tursu

Method: GET

Parameter	Description	Required/ Optional
-----------	-------------	-----------------------

Example query:

<http://3.232.20.250/recommendation/newest>

Response:

JSON Response: List of products with product_info

Notes:

["/recommendation/recommendation_pack"](#)

Recommendation pack endpoint returns all recommendation types at the same request

Method: GET

Parameter	Description	Required/ Optional
-----------	-------------	-----------------------

Example query:

http://3.232.20.250/recommendation/recommendation_pack

Response:

JSON Response: recommendation_pack:

- **recommended** → List of products with product_info
- **bestseller** → List of products with product_info
- **top_rated** → List of products with product_info
- **newest_arrival** → List of products with product_info

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token seperated with a whitespace

"/admin/verificationpending/products/"

Verification pending products endpoint is used by admin to get verification pending products

Method: **GET**

Parameter	Description	Required/ Optional
-----------	-------------	-----------------------

Example query:

<http://3.232.20.250/admin/verificationpending/products/>

Response:

JSON Response: List of products with product_info

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/admin/removeproduct/"

Product removing endpoint is used by admin to delete a product by its id.

Method: **POST / DELETE**

Parameter	Description	Required/ Optional
id	int: ID of the product to be deleted	Required

Example query:

<http://3.232.20.250/admin/removeproduct/>

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/admin/verifyproduct/"

Product verification endpoint is used by admin to verify a product by its id.

Method: **POST**

Parameter	Description	Required/ Optional
id	int: ID of the product to be verified	Required

Example query:

<http://3.232.20.250/admin/verifyproduct/>

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

["/admin/deletecomment/"](#)

Comment deletion endpoint is used by admin to delete a comment by its id.

Method: **POST / DELETE**

Parameter	Description	Required/ Optional
id	int: ID of the comment to be deleted	Required

Example query:

<http://3.232.20.250/admin/deletecomment/>

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

["/admin/banuser/"](#)

User banning endpoint is used by admin to ban a user by its username.

Method: **POST**

Parameter	Description	Required/Optional
username	string: username of the user to be banned	Required

Example query:

<http://3.232.20.250/admin/banuser/>

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/user/signup/google"

Sign up with Google endpoint.

Method: POST

Parameter	Description	Required/Optional
tokenId	string: Token ID of the Google token obtained from Google	Required
is_vendor	if customer leave it empty, if vendor any non empty value is evaluated as true.	Required
IBAN	string: IBAN of the vendor	Required for Vendor
latitude	float: Latitude of the location of Vendor	Required for Vendor
longitude	float: Longitude of the location of Vendor	Required for Vendor
city	string: City of the vendor	Required for Vendor

Example query:

<http://3.232.20.250/user/signup/google>

Response:

JSON Response: login_info

Notes:

"/user/login/google"

Login with Google endpoint

Method: **POST**

Parameter	Description	Required/ Optional
tokenId	string: Token ID of the Google token obtained from Google	Required

Example query:

<http://3.232.20.250/user/login/google>

Response:

JSON Response: login_info

Notes:

"/user/signup"

Sign up endpoint. Sends a verification code to given email address

Method: **POST**

Parameter	Description	Required/ Optional
username	string: User's unique username.	Required
password	string: User's password.	Required
email	string: User's email address.	Required
first_name	string: User's first name.	Required
last_name	string: User's last name.	Required
IBAN	string: IBAN of the vendor	Required for Vendor
latitude	float: Latitude of the location of Vendor	Required for Vendor
longitude	float: Longitude of the location of Vendor	Required for Vendor
city	string: City of the vendor	Required for Vendor

Example query:

<http://3.232.20.250/user/signup>

Response:

JSON Response: {}

Notes:

"/user/login"

Login endpoint

Method: POST

Parameter	Description	Required/ Optional
email	string: Email or username of the user	Required
password	string: Password of the user	Required
verification_code	string: Verification code for the user if needed	Optional

Example query:

<http://3.232.20.250/user/login>

Response:

JSON Response: login_info

Notes:

"/user/resend_verification_code"

Sends verification email to the given email, doesn't override the old ones

Method: POST

Parameter	Description	Required/ Optional
email	string: Email or username of the user	Required

Example query:

http://3.232.20.250/user/resend_verification_code

Response:

JSON Response: {}

Notes:

"/user/forgot_password"

Resets the password and sends an email for the new one.

Method: **POST**

Parameter	Description	Required/ Optional
email	string: Email or username of the user	Required

Example query:

http://3.232.20.250/user/forgot_password

Response:

JSON Response: {}

Notes:

"/helper/allbrands"

Returns all existing brands in a list.

Method: **GET**

Parameter	Description	Required/ Optional
-----------	-------------	-----------------------

Example query:

<http://3.232.20.250/helper/allbrands>

Response:

JSON Response: string list

Notes:

"/helper/allvendors"

Returns all verified vendors in a list.

Method: **GET**

Parameter	Description	Required/ Optional
-----------	-------------	-----------------------

Example query:

<http://3.232.20.250/helper/allvendors>

Response:

JSON Response: string list

Notes:

["/helper/allcategories"](#)

Returns all existing categories in a list.

Method: GET

Parameter	Description	Required/ Optional
-----------	-------------	-----------------------

Example query:

<http://3.232.20.250/helper/allcategories>

Response:

JSON Response: string list

Notes:

["/customerpage"](#)

Returns customer information.

Method: GET

Parameter	Description	Required/ Optional
-----------	-------------	-----------------------

Example query:

<http://3.232.20.250/customerpage>

Response:

JSON Response: customer_info

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token seperated with a whitespace

["/vendorpage"](#)

Returns vendor information.

Method: GET

Parameter	Description	Required/ Optional
-----------	-------------	-----------------------

Example query:

<http://3.232.20.250/vendorpage>

Response:

JSON Response: vendor_info

Notes:

- Auth token should be added to the header of the request as:
“Authorization: ‘Token <auth_token>’” where key is Authorization and value is string with the word “Token” and the actual auth_token separated with a whitespace

"/vendorpage/public"

The public vendor page endpoint returns public information about given vendor

Method: GET

Parameter	Description	Required/ Optional
vendor_name	string: Name of the vendor	Required

Example query:

Response:

JSON Response: public_vendor_info

"/message/flow/admin/"

Message flow for admin endpoint returns the message flow of the admin user

Method: GET

Parameter	Description	Required/ Optional

Example query:

Response:

JSON Response: List of message flows with message_flow_type_1

Notes:

- Auth token should be added to the header of the request as:
“Authorization: ‘Token <auth_token>’” where key is Authorization and value is string with the word “Token” and the actual auth_token separated with a whitespace

"/message/flow/customer/"

Message flow for customer endpoint returns the message flow of the customer user

Method: GET

Parameter	Description	Required/ Optional
-----------	-------------	-----------------------

Example query:

Response:

JSON Response: List of message flows with **message_flow_type_2**

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/message/flow/vendor/"

Message flow for vendor endpoint returns the message flow of the vendor user

Method: GET

Parameter	Description	Required/ Optional
-----------	-------------	-----------------------

Example query:

Response:

JSON Response: List of message flows with **message_flow_type_1**

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/message/startflow/customer/"

Message start flow for customer endpoint is used by customer to start a chat with a vendor with given vendor name

Method: POST

Parameter	Description	Required/ Optional
vendor_name	string: Name of the vendor to start a chat with	Required

Example query:

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/message/startflow/vendor/"

Message start flow for vendor endpoint is used by customer to start a chat with a admin about an order or a product

Method: POST

Parameter	Description	Required/Optional
context	string: "product" or "order"	Required
object_id	int: id of the object that the context is related to i.e. MessageFlow is created to discuss on Product with id 37 → context=product, object_id=37	Required

Example query:

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/message/send/admin/tovendor/"

Admin send message to vendor endpoint is used by admin to send message to a vendor in the given flow

Method: POST

Parameter	Description	Required/Optional
message	string: message with max 200 characters	Required
flow_id	int: id of the related flow	Required

Example query:

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
“Authorization: ‘Token <auth_token>’” where key is Authorization and value is string with the word “Token” and the actual auth_token separated with a whitespace

"/message/send/customer/tovendor/"

Customer send message to vendor endpoint is used by customer to send message to a vendor in the given flow

Method: POST

Parameter	Description	Required/Optional
message	string: message with max 200 characters	Required
flow_id	int: id of the related flow	Required

Example query:

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
“Authorization: ‘Token <auth_token>’” where key is Authorization and value is string with the word “Token” and the actual auth_token separated with a whitespace

"/message/send/vendor/tocustomer/"

Vendor send message to customer endpoint is used by vendor to send message to a customer in the given flow

Method: POST

Parameter	Description	Required/Optional
message	string: message with max 200 characters	Required
flow_id	int: id of the related flow	Required

Example query:

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
“Authorization: ‘Token <auth_token>’” where key is Authorization and value is

string with the word "Token" and the actual auth_token separated with a whitespace

"/message/send/vendor/toadmin/"

Vendor send message to admin endpoint is used by vendor to send message to a admin in the given flow

Method: POST

Parameter	Description	Required/Optional
message	string: message with max 200 characters	Required
flow_id	int: id of the related flow	Required

Example query:

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/message/chat/ofcustomer/"

Chat of customer endpoint is used to get messages in the message flow with the given flow id

Method: GET

Parameter	Description	Required/Optional
flow_id	int: ID of the message flow	Required

Example query:

Response:

JSON Response: List of message_info:

message_info:

- **sender** → string: "self" if current user sent the message, "other" if current user received the message
- **customer** → string: username of the customer messaging
- **vendor_name** → string: name of the vendor messaged
- **message** → string: message text

- **date_sent** → string: datetime repr of the time of message

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/message/chat/ofvendor/wadmin/"

Chat of vendor endpoint is used to get messages of vendor with admin in the message flow with the given flow id

Method: GET

Parameter	Description	Required/Optional
flow_id	int: ID of the message flow	Required

Example query:

Response:

JSON Response: List of **message_info**:

message_info:

- **sender** → string: "self" if current user sent the message, "other" if current user received the message
- **admin** → string: username of the admin messaged
- **vendor_name** → string: name of the vendor messaging
- **message** → string: message text
- **date_sent** → string: datetime repr of the time of message

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/message/chat/ofvendor/wcustomer/"

Chat of vendor endpoint is used to get messages of vendor with customer in the message flow with the given flow id

Method: GET

Parameter	Description	Required/Optional
flow_id	int: ID of the message flow	Required

Example query:

Response:

JSON Response: List of **message_info**:

message_info:

- **sender** → string: "self" if current user sent the message, "other" if current user received the message
- **customer** → string: username of the customer messaged
- **vendor_name** → string: name of the vendor messaging
- **message** → string: message text
- **date_sent** → string: datetime repr of the time of message

Notes:

- Auth token should be added to the header of the request as: "Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/message/chat/ofadmin/"

Chat of admin endpoint is used to get messages in the message flow with the given flow id

Method: GET

Parameter	Description	Required/Optional
flow_id	int: ID of the message flow	Required

Example query:

Response:

JSON Response: List of **message_info**:

message_info:

- **sender** → string: "self" if current user sent the message, "other" if current user received the message
- **admin** → string: username of the admin messaging
- **vendor_name** → string: name of the vendor messaged
- **message** → string: message text
- **date_sent** → string: datetime repr of the time of message

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/as/vendor"

Returns all verified vendors in a list.

Method: **GET**

Parameter	Description	Required/ Optional
vendor_name	string: Name of the vendor to get the activity stream of	Required

Example query:

http://3.232.20.250/as/vendor?vendor_name=Apple

Response:

JSON Response: Activity Stream

Notes:

"/as/product"

Returns all verified vendors in a list.

Method: **GET**

Parameter	Description	Required/ Optional
product_id	int: ID of the product to get the activity stream of	Required

Example query:

http://3.232.20.250/as/product?product_id=1

Response:

JSON Response: Activity Stream

Notes:

"/notifications/get_notifications"

Returns a list of notifications for user.

Method: **GET**

Parameter	Description	Required/ Optional
-----------	-------------	-----------------------

Example query:

Response:**JSON Response:** List of **notification**:**notification:**

- **id** → int: id of the notification
- **type** → int: type of the notification (look below notification types for valid types)
- **read** → bool, true if read, false otherwise
- **product_name** → string: name of the product
- **product_id** → int: id of the product
- **order_id** → int: only if order related
- **status** → string: status order, only if order related
- **new_value** → int: changed new value, only if product related

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace
-

"/notifications/set_read"

Set's given notification as read

Method: POST

Parameter	Description	Required/ Optional
id	int: ID of the notification	Required

Example query:**Response:****HTTP Response:** Success or fail messages**Notes:**

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace
-

"/notifications/create_alert"

Creates an alert with given type, product and value

Method: **POST**

Parameter	Description	Required/ Optional
product_id	int: id of the product to set an alarm	Required
type	int: type of the alert	Required
value	int: value for according type	Optional

Example query:

Response:

Json Response: ID of the created alert

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

"/notifications/get_alerts"

Returns the alerts that the user created. If product_id is given, returns related alerts.

Method: **GET**

Parameter	Description	Required/ Optional
product_id	int: id of the product to filter	Optional

Example query:

Response:

JSON Response: List of alerts:

alert:

- **id** → int: id of the notification
- **type** → int: type of the notification (look below alert types for valid types)
- **value** → value for the according alert type
- **product_name** → string: name of the product
- **product_id** → int: id of the product

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace
-

"/notifications/delete_alert"

Deletes given alert.

Method: POST, DELETE

Parameter	Description	Required/ Optional
id	int: ID of the alert to delete	Required

Example query:

Response:

HTTP Response: Success or fail messages

Notes:

- Auth token should be added to the header of the request as:
"Authorization: 'Token <auth_token>'" where key is Authorization and value is string with the word "Token" and the actual auth_token separated with a whitespace

DATA STRUCTURES**product_info:**

Field	Description
id	int: ID of the product
name	string: Name of the product
photo_url	string: Link to the photo of the product
vendor_name	string: Name of the vendor of the product
category	string: Name of the category of the product
rating	float: Rating of the product
stock	int: Available stock of the product
price	float: Price of the product
brand	string: Name of the brand of the product

detailed_product_info:

Field	Description
id	int: ID of the product
name	string: Name of the product
photo_url	string: Link to the photo of the product
vendor_name	string: Name of the vendor of the product
category	string: Name of the category of the product
rating	float: Rating of the product
stock	int: Available stock of the product
price	float: Price of the product
brand	string: Name of the brand of the product
comments	List of comment
all_photos	List of string: List of links to alternative photos of the product

login_info:

Field	Description
auth_token	string: Auth token of the user
first_name	string: First name of the user
last_name	string: Last name of the user
user_type	string: "admin", "customer" or "vendor"

customer_info:

Field	Description
username	string: User name of the customer
email	string: Email of the customer
first_name	string: First name of the customer
last_name	string: Last name of the customer
money_spent	float: Total amount of money spent of the customer
orders	order_info: Orders of the customer
lists	list of product: Product lists of the customer

order_info:

Field	Description
vendor	string: User name of the vendor
product	int: ID of the product
status	string: Status of the order
cargoID	int: ID of the cargo
orderDate	date: Date of the order
estimatedArrivalDate	date: Estimated arrival date of the order
arrivalDate	date: Arrival date of the order
quantity	int: Order quantity
comment	string: Comment of the order

vendor_info:

Field	Description
username	string: User name of the vendor
email	string: Email of the vendor
first_name	string: First name of the vendor
last_name	string: Last name of the vendor
attitude	string: Attitude of the location of the vendor
longitude	string: Longitude of the location of the vendor
iban	string: Iban of the vendor
rating	float: Rating of the vendor
orders	order_info: Orders of the vendor
products	product_info: Products of the vendor

comment:

Field	Description
id	int: ID of the comment

customer	string: Username of the customer commented
text	string: Text of the comment
rating	float: Rating given with the comment

message_flow_type_1:

Field	Description
id	int: Flow id
notify	bool: True if unseen messages are in flow
context	string: “ product ” or “ order ”
object_id	int: ID of the object that the context is related to i.e. MessageFlow is created to discuss on Product with id 37 → context=product, object_id=37
product	string: Name of the product if the context is “product”, name of the ordered product if the context is “order”
customer	string: Username of the customer who ordered the product if the context is “order”, null if context is product
type	string: “ admin ” for vendor-admin communication “ customer ” for customer-vendor communication
vendor_name	string: name of the vendor messaged

message_flow_type_2:

Field	Description
id	int: Flow id
notify	bool: True if unseen messages are in flow
vendor_name	string: name of the vendor messaged

public_vendor_info:

Field	Description
username	string: Username of the vendor
email	string: Email of the vendor
latitude	float: Latitude of the location of the vendor
longitude	float: Longitude of the location of the vendor

city	string: City of the location of the vendor
first_name	string: Name of the vendor
last_name	string: Empty string
iban	string: IBAN of the vendor
rating	float: Rating of the vendor
products	List of product_info

notification_types:

Field	Value
ORDER STATUS CHANGE	1
PRODUCT VERIFIED	2
PRICE BELOW ALERT	3
PRICE CHANGE ALERT	4
STOCK ABOVE ALERT	5

alert_types:

Field	Value
PRICE BELOW ALERT	1
PRICE CHANGE ALERT	2
STOCK ABOVE ALERT	3