PIZZA DELIVERY API DOCS

This is a simple API for ordering pizza. In the current version there is no GUI available, so the requests are to be created with products like postman for example. All of the data sent to this API should be in JSON format. The responses provided by the API will be JSON formatted as well.

The following header, *Content-Type: application/json* should be provided for every request

REGISTER A NEW USER:

METHOD: POST

URL: /users

In order to create a new profile the user should provide the following information:

- First name
- Last name
- Phone number – the number should be 10 digits
- Password – at least 8 characters long
- Address
- E-mail

After the required information has been successfully submitted, the new user profile will be created. After the user profile is created the user should request a token which is going to be used alongside the phone and password for authentication

REQUEST NEW AUTHENTICATION TOKEN:

METHOD: POST

URL: /tokens

For a token to be issued the phone number and password of the user are required. A token is valid for one hour. The token contains the user's phone, token ID and an expiration time field. This information could be requested, the token can be updated and deleted

GET AUTHENTICATION TOKEN INFORMATION

METHOD: GET

URL: /tokens?tokenId=YOUR_TOKEN_ID

In order to get the token related data the user should provide the token ID as a parameter in the query string as per the example provided above.

UPDATE AUTHENTICATION TOKEN INFORMATION

METHOD: PUT

URL: /tokens

To update an existing token a token ID and an extend parameter which should de set to “**TRUE**” should be provided. In case of successful request the token’s validity is extended with one hour, counting from the time of the request for the extension. There is no limit how many times a valid token could be extended. If the token has expired, new token should be requested from the user.

DELETE AUTHENTICATION TOKEN INFORMATION

METHOD: DELETE

URL: /tokens

For a valid token to be deleted, the token ID should be provided in the request body and token ID as a header *e.g. token: YOUR_TOKEN_ID*. After submission of a successful DELETE request the token is deleted.

GET THE INFORMATION OF AN EXISTING USER

METHOD: GET

URL: /users?phone=USER_PHONE

To get the user profile information, the user should provide a valid phone number in the query string as per the example provided above and a token ID in the headers *e.g. token: YOUR_TOKEN_ID*. Upon a successful request all the user’s information but the password is retrieved.

UPDATE USER PROFILE

METHOD: PUT

URL: /users

To update a user profile the user should authenticate with a valid token as a header, *e.g. token: YOUR_TOKEN_ID* and a phone number which should be provided in the request body. After successful authentication the user can change it’s password, first name, last name, address or email.

DELETE USER PROFILE

METHOD: DELETE

URL: /users

To delete it’s profile the user should provide authentication token as a header *e.g. token: YOUR_TOKEN_ID*, phone number and a password. After successful submission of the above mentioned credentials the user profile will be deleted as well as all the orders related to the user (these orders are still considered valid, because they are paid for and a receipt is available).

GET THE MENU

METHOD: GET

URL: /products?phone=USER_PHONE

Only authenticated users are able to make orders. For a new order to be created the user should authenticate with a token, provided as a header e.g. token: YOUR_TOKEN_ID and a phone number as a query string parameter as per the example provided above

MAKE AN ORDER

METHOD: POST

URL: /orders

Only authenticated users are able to make orders. For a new order to be created the user should authenticate with a token, provided as a header *e.g. token: YOUR_TOKEN_ID*. In the request body the user should provide phone number and the actual order as a object *e.g. “orderItems”: {ITEM: QUANTITY, …}*. After the order is submitted the order value is calculated a charge is created and the user receives the receipt via email.

GET ORDER INFORMATION

METHOD: POST

URL: /orders?phone=USER_PHONE&orderId=ORDER_ID

Only authenticated users are able to make orders. For a new order to be created the user should authenticate with a token, provided as a header *e.g. token: YOUR_TOKEN_ID,* phone number and the order ID as query string parameters as per the example provided above.

MARK AN ORDER AS COMPLETED

METHOD: PUT

URL: /orders/markCompleted

There is the possibility for any authenticated user to mark an order as completed so the order could be deleted in a week interval (see **CHECKERS** section for reference). The user should authenticate with a token, provided as a header *e.g. token: YOUR_TOKEN_ID,* phone number and the order ID should be provided in the request body. After successful submission the order with the provided ID is marked as completed.

CHECKERS

There are two types of checkers running in the background of this API:

- ***ORDER CHECKER:***

  All orders are being checked once in a week, for their completion status. If an order is marked as a *completed*, this means that the user has no intention to access this order information anymore and the order will be deleted

- ***TOKEN CHECKER:***

  All tokens are being checked once every hour and if a token is expired, it is deleted so the user could not use the token in question for authentication and should request a new token to be issued

3rd PARTY APIs

This API is integrated with **STRIPE** and **MAILGUN**.