

Software Mind Internship

API Documentation

Mateusz Boroch

Juny 2025

1 Authorization

This API uses **JSON Web Tokens (JWT)** for authorization. JWTs are issued upon successful *user registration* or *login* and must be included in the **Authorization** header of all protected requests.

Getting a Token

Clients can obtain a JWT by calling one of the following endpoints:

- **POST /api/user/register** – Registers a new user and returns a JWT.
- **POST /api/user/login** – Authenticates an existing user and returns a JWT.

Each successful response will include a JWT token in the response body:

```
{
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

Using the Token

To access protected endpoints, include the JWT in the **Authorization** header using the **Bearer** schema:

Authorization: Bearer <your-jwt-token>

Example:

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...

Token Expiration

JWT tokens have an expiration time encoded in the payload. After the token expires, the user must re-authenticate via the **login** endpoint to receive a new token.

Error Responses

If a request is made to a protected endpoint without a valid token, the API will respond with an appropriate error:

- **401 Unauthorized** – Missing, invalid, or expired token.
- **403 Forbidden** – Token is valid but the user does not have permission for the requested action.

2 User API

Post `api/user/login`

Description: Authenticates an existing user using their credentials and returns a JSON Web Token (JWT) for authorized access to protected resources.

- **Method:** POST
- **URL:** `/api/user/login`
- **Authentication Required:** No

Request Headers

Content-Type: `application/json`

Request Body

```
{  
  "email": "email@example.com",  
  "password": "Password"  
}
```

Response Example

```
{  
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."  
}
```

Response Codes

- **200 OK** – Successful request
- **400 Bad Request** – Wrong email or password

POST api/user/Register

Description: Registers a new user and returns a JSON Web Token (JWT) for authorization in future requests.

- **Method:** POST
- **URL:** /api/user/register
- **Authentication Required:** No

Request Headers

Content-Type: application/json

Request Body

```
{
  "nickname": "Nickname",
  "password": "Password",
  "email": "email@example.com",
  "phoneNumber": "123 123 123"
}
```

Response Example

```
{
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

Response Codes

- 200 OK – Successful request
- 400 Bad Request – Nickname or email exist in database

3 Product API

GET `api/product/get/{id}`

Description: Retrieves detailed information about a single product by its unique identifier.

- **Method:** GET
- **URL:** `/api/product/get/{id}`
- **Authentication Required:** No

Request Headers

Content-Type: application/json

Request Body

```
{  
}
```

Response Example

```
{  
  "id": 5,  
  "name": "Item name",  
  "description": "Item description"  
}
```

Response Codes

- 200 OK – Successful request
- 404 Not Found – Product with given id was not found

POST api/product/add

Description: Creates a new product in the system. Returns ID of created product.

- **Method:** POST
- **URL:** /api/product/add
- **Authentication Required:** Yes

Request Headers

Authorization: Bearer <jwt-token>
Content-Type: application/json

Request Body

```
{
  "name": "Item name",
  "description": "Item description"
}
```

Response Example

```
{
  5
}
```

Response Codes

- 200 OK – Successful request
- 400 Bad Request – Invalid input(specified in response text)
- 401 Unauthorized – Missing or invalid JWT

PUT api/product/update

Description: Updates the details of an existing product. The product to be updated is identified by its ID, which must be included in the request body.

- **Method:** PUT
- **URL:** /api/product/Update
- **Authentication Required:** Yes

Request Headers

Authorization: Bearer <jwt-token>
Content-Type: application/json

Request Body

```
{
  "id": 1,
  "name": "New name",
  "description": "New description"
}
```

Response Example

```
{
  6
}
```

Response Codes

- 200 OK – Successful request
- 400 Bad Request – Invalid input or new data already exist in database
- 401 Unauthorized – Missing or invalid JWT

POST api/product/getList

Description: Retrieves a list of all available products. Supports optional filtering or pagination depending on query parameters.

- **Method:** POST
- **URL:** /api/product/getList
- **Authentication Required:** No

Request Headers

Content-Type: application/json

Request Body

```
{
  "searchingName": "product",
  "page": 1,
  "pageSize": 20
}
```

Response Example

```
{
  "productList": [
    {
      "id": 1,
      "name": "Product one"
    },
    {
      "id": 3,
      "name": "Product two"
    }
  ],
  "count": 9,
  "searchingName": "product",
  "actualPage": 1,
  "pages": 5,
  "pageSize": 2
}
```

Response Codes

- 200 OK – Successful request
- 400 Bad Request – Invalid page number or size
- 404 Not Found – Resources not found

4 Negotiation API

POST api/negotiation/add

Description: Initiates a new negotiation for a specific product. Returns negotiation token for future use.

- **Method:** POST
- **URL:** /api/negotiation/add
- **Authentication Required:** No

Request Headers

Content-Type: application/json

Request Body

```
{
  "price": 1.50,
  "productID": 1
}
```

Response Example

```
{
  q9Ato0NOZxY3
}
```

Response Codes

- 200 OK – Successful request
- 400 Bad Request – Wrong product ID or price below/equals 0.00

GET api/negotiation/get/{id}

Description: Retrieves detailed information about a specific negotiation by its unique identifier.

- **Method:** GET
- **URL:** /api/negotiation/get/{id}
- **Authentication Required:** Yes

Request Headers

Authorization: Bearer <jwt-token>
Content-Type: application/json

Request Body

```
{  
}
```

Response Example

```
{  
  "id": 5,  
  "price": 1.5,  
  "attempCounter": 0,  
  "lastAttemp": "2025-06-09T11:02:12.0029157",  
  "status": 0,  
  "negotiationToken": "q9Ato0N0ZxY3",  
  "productName": "Item name",  
  "productID": 1  
}
```

Response Codes

- 200 OK – Successful request
- 401 Unauthorized – Missing or invalid JWT
- 404 Not Found – Resource not found

GET /api/negotiation/getByToken/{token}

Description: Retrieves a specific negotiation using a unique access token. This endpoint allows limited access to negotiation details without requiring user authentication. Access is controlled via the provided token.

- **Method:** <GET|POST|PUT|DELETE>
- **URL:** /api/negotiation/getByToken/{token}
- **Authentication Required:** No

Request Headers

Content-Type: application/json

Request Body

```
{  
}
```

Response Example

```
{  
  "id": 5,  
  "price": 1.5,  
  "attempCounter": 0,  
  "lastAttemp": "2025-06-09T11:02:12.0029157",  
  "status": 0,  
  "negotiationToken": "q9Ato0NOZxY3",  
  "productName": "Item name",  
  "productID": 1  
}
```

Response Codes

- 200 OK – Successful request
- 404 Not Found – Resource not found

POST api/negotiation/getList

Description: Retrieves a list of all negotiations in the system. Supports optional filtering and pagination.

- **Method:** POST
- **URL:** /api/negotiation/getList
- **Authentication Required:** Yes

Request Headers

Authorization: Bearer <jwt-token>
Content-Type: application/json

Request Body

```
{
  "searchingStatus": 0,
  "searchingProduct": "Item",
  "page": 1,
  "pageSize": 1
}
```

Response Example

```
{
  "negotiationList": [
    {
      "id": 3,
      "lastAttemp": "2025-06-05T12:42:05.3520521",
      "status": 0,
      "productName": "Item one",
      "productID": 5
    }
  ],
  "count": 3,
  "searchingStatus": 0,
  "searchingProduct": "Item",
  "actualPage": 1,
  "pages": 3,
  "pageSize": 1
}
```

Response Codes

- 200 OK – Successful request

- 400 Bad Request – Invalid page or page size
- 401 Unauthorized – Missing or invalid JWT
- 404 Not Found – Resource not found

PUT `api/negotiation/responseToOffer`

Description: Allows a negotiation participant to respond to an offer within an existing negotiation. The response can accept or reject the offer. Returns negotiation ID.

- **Method:** PUT
- **URL:** `/api/negotiation/responseToOffer`
- **Authentication Required:** Yes

Request Headers

Authorization: Bearer <jwt-token>
Content-Type: application/json

Request Body

```
{  
  "id": 3,  
  "status": 1  
}
```

Response Example

```
{  
  3  
}
```

Response Codes

- 200 OK – Successful request
- 400 Bad Request – Invalid input. Details in response.
- 401 Unauthorized – Missing or invalid JWT
- 404 Not Found – Resource not found

PUT `api/negotiation/newOffer`

Description: Submits a new offer as part of an ongoing negotiation. This allows a participant to propose updated price to continue the negotiation process. Returns negotiation ID.

- **Method:** PUT
- **URL:** `/api/negotiation/newOffer`
- **Authentication Required:** No

Request Headers

Content-Type: `application/json`

Request Body

```
{
  "token": "q9Ato0NOZxY3",
  "price": 14
}
```

Response Example

```
{
  5
}
```

Response Codes

- 200 OK – Successful request
- 400 Bad Request – Invalid input. Details in response
- 404 Not Found – Resource not found