

API Integration Documentation:

Backend ↔ Frontend

1. Overview

This document outlines the backend API routes needed and their integration with the frontend. It includes provided inputs from the frontend, required headers, sample request formats, and sample response outputs expected from the backend.

2. Frontend Setup

All requests from the frontend has been sent to the corresponding backend API routes as described below. For authenticated routes under /api/user, the Authorization header is must and has been included in the format:

Authorization: Bearer <token>

3. Authentication related API's base Url: (domain/api/auth)

Sample of how User Data should look:

```
[
  {
    "id": 1744478539040,
    "name": "Rhiannon Grimes",
    "username": "mivewijuz",
    "email": "tysoh@mailinator.com",
    "password": "$2b$10$zDYJa.IBqDRFzyJciLsJkelAr4VDpPaF4NT9b8l2pDtb0YTzAh3N.",
    "userType": "buyer",
    "reviews": [],
    "products": [],
    "cart": [
      {
        "productid": 11,
        "quantity": 1,
        "offeredPrice": 499
      }
    ],
    "wishlist": [
      {
        "productId": 11
      }
    ]
  }
]
```

domain/api/auth /signup [POST]

Sample Request:

```
{  
  "name": "Test User",  
  "username": "admin1",  
  "email": "test1@gmail.com",  
  "password": "Test@123"  
}
```

Sample Response:

```
{  
  "success": true,  
  "message": "Signup successful!",  
  "tokens": {  
    "access": "eyJhbGciOiJI..."  
  },  
  "user": {  
    "id": 1744600225690,  
    "name": "Test User",  
    "username": "admin2",  
    "email": "test5@gmail.com",  
    "userType": "buyer"  
  }  
}
```

domain/api/auth /login [POST]

Sample Request:

```
{  
  "username": "admin1",  
  "password": "Test@123"  
}
```

Sample Response:

```
{  
  "message": "Login successful",  
  "user": {  
    "id": 1744600225690,  
    "name": "Test User",  
    "username": "admin2",
```

```
    "email": "test5@gmail.com",
    "userType": "buyer",
  },
  "tokens": {
    "access": "eyJhbGciOiJI..."
  }
}
```

4. Products related API's base Url:(domain/api/products)

Sample of how Products List should look like:

```
[
  {
    "id": 1,
    "name": "Aerial Phantom Drone",
    "category": "Drones",
    "publisher": "skyflyr99",
    "description": "High-performance drone with 4K camera and long flight time.",
    "stock": 50,
    "sold": 22,
    "productAvatar": "/images/Drones/drone1.png",
    "images": [
      "/images/Drones/drone1.png",
      "/images/Drones/drone2.png",
      "/images/Drones/drone3.png",
      "/images/Drones/drone4.png",
      "/images/Drones/drone5.png"
    ],
    "originalPrice": 1499,
    "offeredPrice": 1199,
    "rating": 4.8,
    "createdAt": "2025-04-01",
    "reviews": [
      {
        "user": "droneGuru",
        "comment": "Amazing flight control and stable footage!",
        "rating": 5,
        "profileImg": "/images/Drones/drone2.png"
      }
    ],
    "features": ["4K Camera", "GPS", "30 min Flight", "Stabilization", "Auto Return"]
  }
]
```

domain/api/products /new-arrivals [GET]

Returns the 4 latest products by createdAt.

Sample Response:

```
[
  {
    "id": 1020,
    "title": "Fresh Sneakers",
    "createdAt": "2024-04-10T08:00:00Z"
  },
  ...
]
```

domain/api/products /top-sold [GET]

Returns the 5 top sold products.

Sample Response:

```
[
  {
    "id": 1002,
    "title": "Running Shoes",
    "sold": 233
  },
  ...
]
```

domain/api/products /category/:category_name [GET]

Returns products by category with sorting and pagination.

Query Params: sort, page, limit

Sample Request: /category/shoes?sort=price_desc&page=1&limit=6

Sample Response:

```
{
  "products": [ { "id": 1001, "title": "Shoe X" }, ... ],
  "total": 20,
  "currentPage": 1,
  "totalPages": 4
}
```

domain/api/products /:product_id [GET]

Fetches product details by product ID.

Sample Request: /api/products/1002

Sample Response:

```
{
  "id": 1002,
  "title": "Running Shoes",
  "price": 1800,
  "sold": 233,
  "category": "shoes" .....
  .....
}
```

5. User APIs (domain/api/user)

All endpoints require an Authorization header with a valid JWT token.

domain/api/user /get-cart [GET]

Requires Authorization header.

Authorization: Bearer <token>

Sample Response:

```
{
  "cart": [
    {
      "productid": 1002,
      "quantity": 2,
      "offeredPrice": 1799
    },
    {
      "productid": 1010,
      "quantity": 1,
      "offeredPrice": 1299
    }
  ]
}
```

domain/api/user /update-cart [POST]

Requires Authorization header.

Authorization: Bearer <token>

Sample Request:

```
{
  "cart": [
    {
      "productid": 1002,
      "quantity": 3,
      "offeredPrice": 1799
    },
    {
      "productid": 1010,
      "quantity": 0
    }
  ]
}
```

Sample Response:

```
{
  "message": "Cart updated successfully",
  "cart": [
    {
      "productid": 1002,
      "quantity": 3,
      "offeredPrice": 1799
    }
  ]
}
```

domain/api/user /get-wishlist [GET]

Requires Authorization header.

Authorization: Bearer <token>

Sample Response:

```
{
  "wishlist": [
    {
      "productId": 1002
    },
    {
      "productId": 1011
    }
  ]
}
```

domain/api/user /update-wishlist [POST]

Requires Authorization header.

Authorization: Bearer <token>

Sample Request:

```
{
  "wishlist": [
    {
      "productId": 1011
    },
    {
      "productId": 1012
    }
  ]
}
```

Sample Response:

```
{
  "message": "Wishlist updated successfully",
  "wishlist": [
    {
      "productId": 1011
    },
    {
      "productId": 1012
    }
  ]
}
```
