FaShop API Documentation

Overview

FaShop is an e-commerce platform with a comprehensive REST API that provides functionality for both administrators and customers. This documentation covers all available API endpoints, their usage, and examples.

Base URL

```
http://localhost:8000/api
```

Authentication

The API uses token-based authentication. There are two types of authentication:

1. Admin Authentication
2. Customer Authentication

Admin Authentication

```
POST /admin/login
```

Request body:

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

Response:

```
{
    "status": true,
    "message": "Login successful",
    "data": {
        "token": "Bearer ..."
    }
}
```

Use the returned token in subsequent requests in the Authorization header:

```
Authorization: Bearer <token>
```

API Endpoints

Admin API

Profile Management

Update Admin Profile

```
POST /profiles/update
```

Required fields:

- name
- email
- phone

Update Admin Password

```
POST /profiles/update/password
```

Required fields:

- current_password
- password
- password_confirmation

Brand Management

List All Brands

```
GET /brands
```

Get Single Brand

```
GET /brands/{id}
```

Create Brand

```
POST /brands
```

Required fields:

- name (object with locale keys)

- logo (image file)
- status (integer)

Update Brand

```
POST /brands/{id}/update
```

Same fields as create

Delete Brand

```
DELETE /brands/{id}/delete
```

Category Management

List All Categories

```
GET /categories
```

Response includes:

- Translations
- Children categories
- Parent category
- Paginated with 20 items per page

Get Single Category

```
GET /categories/{id}
```

Returns category details including:

- Translations
- Children categories
- Parent category information

Create Category

```
POST /categories
```

Required fields:

- name (object with locale translations)
- parent_id (integer, nullable)
- status (integer)
- icon (image file, optional)

Response:

```
{
    "status": true,
    "message": "Created Successfully!",
    "data": {
        "category": {
            "id": 1,
            "name": {"en": "Category Name", "ar": "اسم الفئة"},
            "parent_id": null,
            "status": 1,
            "icon": "path/to/icon.jpg",
            "translations": [...],
            "children": [],
            "_parent": null
        }
    }
}
```

Update Category

```
POST /categories/{id}/update
```

Same fields as create endpoint. Updates both category details and translations.

Delete Category

```
DELETE /categories/{id}/delete
```

Restrictions:

- Cannot delete categories with associated products
- Cannot delete categories with child categories

Error Response (if deletion restricted):

```
{
    "status": false,
    "message": "You Can't Delete This Category Because It Has Associated Products or Subcategories!",
    "code": 409
}
```

Product Management

List All Products

```
GET /products
```

Create Product

```
POST /products
```

Required fields:

- name (object with locale keys)
- description (object with locale keys)
- thumb_image (image file)
- brand_id (integer)
- product_type_id (integer)
- category_id (array)
- qty (integer)
- price (float)
- status (integer)

Optional fields:

- offer_price (float)
- offer_start_date (date)
- offer_end_date (date)
- video_link (string)

Update Product

```
POST /products/{id}/update
```

Same fields as create

Delete Product

```
DELETE /products/{id}
```

Product Variants

List Product Variants

```
GET /products/{id}/product-variants
```

Create Product Variant

```
POST /products/{id}/product-variants
```

Required fields:

- extra_price (float)
- quantity (integer)
- sku (string)
- attribute_values (array)

Update Product Variant

```
POST /products/{id}/product-variants/{variantId}/update
```

Same fields as create

Delete Product Variant

```
DELETE /products/{id}/product-variants/{variantId}
```

Product Gallery

List Product Images

```
GET /product-image-galleries
```

Add Product Images

```
POST /product-image-galleries/{productId}
```

Required fields:

- image (array of image files)

Delete Product Image

```
DELETE /product-image-galleries/{id}/delete
```

Error Handling

The API uses standard HTTP response codes:

- 200: Success
- 201: Created
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden

- 404: Not Found
- 422: Validation Error
- 500: Server Error

Error Response Format:

```
{
    "status": false,
    "message": "Error message here",
    "errors": {
        "field": ["Error message for field"]
    }
}
```

Pagination

List endpoints support pagination with the following query parameters:

- page: Page number (default: 1)
- per_page: Items per page (default: 20)

Response format:

```
{
    "status": true,
    "message": "Success message",
    "data": {
        "current_page": 1,
        "data": [...],
        "first_page_url": "...",
        "from": 1,
        "last_page": 5,
        "last_page_url": "...",
        "next_page_url": "...",
        "path": "...",
        "per_page": 20,
        "prev_page_url": null,
        "to": 20,
        "total": 100
    }
}
```

Localization

The API supports multiple languages. Set the desired locale using the Accept-Language header:

```
Accept-Language: en
```

Supported locales:

- en (English)
- ar (Arabic)

Rate Limiting

The API implements rate limiting to prevent abuse. Current limits:

- Authentication endpoints: 6 requests per minute
- Other endpoints: 60 requests per minute

File Upload

For endpoints that accept file uploads:

- Maximum file size: 80MB
- Supported image formats: jpg, jpeg, png
- Use multipart/form-data content type

Versioning

The current API version is v1. The version is included in the base URL:

```
http://localhost:8000/api/v1
```

Code Examples

PHP (using Guzzle)

```
use GuzzleHttp\Client;

$client = new Client([
    'base_uri' => 'http://localhost:8000/api/',
    'headers' => [
        'Authorization' => 'Bearer ' . $token,
        'Accept' => 'application/json',
    ]
]);

// List products
$response = $client->get('products');
$products = json_decode($response->getBody(), true);
```

JavaScript (using Fetch)

```
const token = 'your-token';

// List products
fetch('http://localhost:8000/api/products', {
    headers: {
        'Authorization': `Bearer ${token}`,
        'Accept': 'application/json'
    }
})
.then(response => response.json())
.then(data => console.log(data));
```

Python (using Requests)

```python
import requests

token = 'your-token'
headers = {
    'Authorization': f'Bearer {token}',
    'Accept': 'application/json'
}

# List products
response = requests.get(
    'http://localhost:8000/api/products',
    headers=headers
)
products = response.json()
```

WebSocket Events

The API also supports real-time updates through WebSocket connections for certain features. WebSocket documentation will be provided separately.

Support

For additional support or to report issues:

- GitHub Issues: [Project Issues (https://github.com/your-repo/issues)](https://github.com/your-repo/issues)
- Email: support@fashop.com

Changelog

Version 1.0.0

- Initial API release
- Basic CRUD operations for products, categories, and brands
- Authentication system
- File upload functionality
- Multi-language support