

Banner Management API Documentation - Frontend

Overview

This document provides comprehensive documentation for the Banner Management frontend API endpoints, including request/response formats and all possible status codes.

Base URL

`/api/frontend/banners`

Authentication

No authentication required for frontend endpoints.

Headers

Content-Type: application/json

Accept: application/json

lang: en|ar (optional, handled by handleLang middleware)

Endpoints

1. Get All Banners

Endpoint: GET `/api/frontend/banners`

Description: Retrieve all banners from the system.

Query Parameters:

- `page` (optional, string): Filter banners by page (e.g., 'home', 'about', 'products', etc.)
- `position` (optional, string): Filter banners by position ('top', 'middle', 'bottom')

Request:

```
GET /api/frontend/banners?page=home&position=top
```

```
Accept: application/json
```

```
lang: en
```

Response Format:

```
{  
  "code": int,  
  "message": string,  
  "errors": array,  
  "data": array|object  
}
```

Success Responses

Status Code: 200 OK

```
{
  "code": 200,
  "message": "Banners fetched successfully",
  "errors": [],
  "data": [
    {
      "id": 1,
      "image": "/storage/banners/banner-1.jpg",
      "url": "/shop",
      "position": "top",
      "page": "home",
      "is_active": true
    },
    {
      "id": 2,
      "image": "/storage/banners/banner-2.jpg",
      "url": "/sale",
      "position": "middle",
      "page": "home",
      "is_active": true
    }
  ]
}
```

Empty Result (200 OK):

```
{
  "code": 200,
  "message": "Banners fetched successfully",
  "errors": [],
  "data": []
}
```

Error Responses

Status Code: 500 Internal Server Error

```
{
  "code": 500,
  "message": "Failed to fetch banners",
  "errors": [],
  "data": null
}
```

Status Code: 503 Service Unavailable

```
{
  "code": 503,
  "message": "Service unavailable",
  "errors": [
    {
      "key": "error",
      "value": "Service unavailable"
    }
  ],
  "data": null
}
```

2. Get Banner by ID

Endpoint: GET /api/frontend/banners/{id}

Description: Retrieve a specific banner by its ID.

Request:

```
GET /api/frontend/banners/1
Accept: application/json
lang: en
```

Parameters:

- id (required, integer): The banner ID

Success Responses

Status Code: 200 OK

```
{
  "code": 200,
  "message": "Banner fetched successfully",
  "errors": [],
  "data": {
    "id": 1,
    "image": "/storage/banners/banner-1.jpg",
    "url": "/shop",
    "position": "top",
    "page": "home",
    "is_active": true
  }
}
```

Error Responses

Status Code: 404 Not Found (Banner Not Found)

```
{
  "code": 404,
  "message": "Banner not found",
  "errors": [],
  "data": null
}
```

Status Code: 500 Internal Server Error

```
{
  "code": 500,
  "message": "Failed to fetch banner",
  "errors": [],
  "data": null
}
```

Status Code: 400 Bad Request (Invalid ID format)

```

{
  "code": 400,
  "message": "Client error",
  "errors": [
    {
      "key": "error",
      "value": "401 Unauthorized - Client Error"
    }
  ],
  "data": null
}

```

Data Structure Reference

Banner Object Structure

```

{
  "id": integer,           // Unique banner identifier
  "image": string,         // URL/path to banner image
  "url": string,           // Optional URL for banner link
  "position": string,      // Banner position (top, middle, bottom)
  "page": string,          // Page where banner appears (home, about, products, etc.)
  "is_active": boolean     // Whether banner is active/enabled
}

```

Response Structure

```

{
  "code": integer,         // HTTP status code
  "message": string,       // Human-readable message
  "errors": array,         // Array of error objects (empty if no errors)
  "data": array|object     // Response data (banners array or single banner)
}

```

Error Object Structure

```
{
  "key": string,          // Error field/type identifier
  "value": string         // Error message description
}
```

Available Enums

Position Values

- top
- middle
- bottom

Page Values

- home (default)
- offers

Translation Support

The API supports multi-language responses through the `lang` header:

- **English:** `lang: en`
- **Arabic:** `lang: ar`

Example with Arabic:

```
GET /api/frontend/banners/1
Accept: application/json
lang: ar
```

```
{
  "code": 200,
  "message": "تم جلب البانر بنجاح",
  "errors": [],
  "data": {
    "id": 1,
    "image": "/storage/banners/banner-1.jpg",
    "url": "/shop",
    "position": "top",
    "page": "home",
    "is_active": true
  }
}
```