

OTP Authentication API Documentation

This document describes the **Send OTP** and **Verify OTP** APIs used for mobile-based authentication.

Base URL

```
http://127.0.0.1:8000/api
```

1. Send OTP API

Endpoint

```
POST /send-otp
```

Description

This API sends a One-Time Password (OTP) to the user's registered mobile number. A `user_session_id` is used to track the login session for guest or logged-in users.

Request Parameters

| Parameter | Type | Required | Description |
|-----------------|--------|----------|---|
| mobile | string | Yes | 10-digit mobile number (must start with 6-9) |
| user_session_id | string | No | Unique session identifier. If not provided, it will be auto-generated |

Request Example (form-data)

```
mobile=7905512947
user_session_id=postman_guest_124
```

Success Response (200 OK)

```
{
  "status": true,
  "message": "OTP sent",
}
```

```
"user_session_id": "postman_guest_124",
"otp_for_testing": 856150
}
```

⚠ Note: `otp_for_testing` should be returned **only in development/testing environments**. It must be removed in production.

Error Response (Validation Failed)

```
{
  "status": false,
  "message": "Invalid mobile number"
}
```

2. Verify OTP API

Endpoint

```
POST /verify-otp
```

Description

This API verifies the OTP entered by the user. On successful verification, it logs the user in and returns an authentication token along with user details.

Request Parameters

| Parameter | Type | Required | Description |
|-----------------|--------|----------|--|
| mobile | string | Yes | Mobile number used to receive OTP |
| otp | string | Yes | 6-digit OTP received on mobile |
| user_session_id | string | Yes | Same session ID used while sending OTP |

Request Example (form-data)

```
mobile=7905512947
otp=856150
user_session_id=postman_guest_124
```

Success Response (200 OK)

```
{
  "status": true,
  "message": "Login successful",
  "token": "23|A9TZ63EGR8ERNySUXkZt8XKeYEgc0MciKCUFqsecb66d5045",
  "user_session_id": "postman_guest_124",
  "user": {
    "id": 3,
    "name": "Gazal",
    "email": "user@gmail.com",
    "mobile": "7905512947",
    "role": "user",
    "user_image": null,
    "user_status": true,
    "email_verified_at": null,
    "created_at": "2025-12-20T07:27:15.000000Z",
    "updated_at": "2026-01-28T06:11:27.000000Z"
  }
}
```

Error Response (Invalid OTP)

```
{
  "status": false,
  "message": "Invalid or expired OTP"
}
```

Authentication

- After successful OTP verification, use the returned token for authenticated requests.
- Send the token in headers as:

```
Authorization: Bearer {token}
```

OTP Rules & Security Notes

- OTP is valid for a limited time (recommended: 5 minutes)
- Resend OTP should be restricted (recommended: 60 seconds cooldown)
- OTP must be invalidated immediately after successful verification
- Do not expose OTP in production APIs

Status Codes Summary

| Status Code | Meaning |
|-------------|------------------|
| 200 | Success |
| 400 | Validation error |
| 401 | Invalid OTP |
| 500 | Server error |

3. Get All Categories API


Endpoint

```
GET /allCategories
```

Description

This API returns a list of all available product categories along with their icons and banner images. It is typically used to display categories on the homepage or category listing screens.

Authentication

 No authentication required (Public API)

Request Parameters

This endpoint does **not** require any request parameters.

Request Example

```
GET /api/allCategories
```

Success Response (200 OK)

```
{  
  "success": true,
```

```

"data": [
  {
    "id": 1,
    "name": "Mutton Cut",
    "icon": "http://127.0.0.1:8000/storage/uploads/category/1765800286_icon.png",
    "banner_image": "http://127.0.0.1:8000/storage/uploads/category/1765800286_banner.jpg"
  },
  {
    "id": 3,
    "name": "Fresh Chicken",
    "icon": "http://127.0.0.1:8000/storage/uploads/category/1765962893_icon.jpg",
    "banner_image": "http://127.0.0.1:8000/storage/uploads/category/1765962893_banner.png"
  }
]
}

```

Response Fields Description

| Field | Type | Description |
|-------------------|--------------|------------------------------|
| success | boolean | Indicates API request status |
| data | array | List of category objects |
| data.id | integer | Unique category ID |
| data.name | string | Category name |
| data.icon | string (URL) | Category icon image URL |
| data.banner_image | string (URL) | Category banner image URL |

Error Response (500 Server Error)

```

{
  "success": false,
  "message": "Something went wrong"
}

```

4. Delete Category API

Endpoint

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

Description

This API deletes a specific category using its unique ID. Once deleted, the category and its associated images (icon/banner) will no longer be available.

Authentication

Authentication required

```
Authorization: Bearer {token}
```

Only **Admin** users should be allowed to access this API.

URL Parameters

| Parameter | Type | Required | Description |
|-----------|---------|----------|-------------------------------------|
| id | integer | Yes | Unique ID of the category to delete |

Request Example

```
DELETE /api/categories/2
```

Success Response (200 OK)

```
{
  "success": true,
  "message": "Category deleted successfully"
}
```

Error Responses

Category Not Found (404)

```
{
  "success": false,
  "message": "Category not found"
}
```

Unauthorized (401)

```
{
  "message": "Unauthenticated."
}
```

Forbidden (403)

```
{
  "success": false,
  "message": "You are not authorized to delete this category"
}
```

Important Notes

- Deleting a category is **irreversible**
- Make sure the category is not linked to active products before deleting
- Associated images should be removed from storage

5. Add to Cart API

Endpoint

```
POST /add-to-cart
```

Description

This API adds a product variant to the user's cart. It supports both guest users and logged-in users using a `user_session_id`.

Authentication

 Not required (Guest & Logged-in supported)

If the user is logged in, the cart can later be mapped to the user account.

Request Parameters (form-data)

| Parameter | Type | Required | Description |
|-----------------|---------|----------|----------------------------------|
| variant_id | integer | Yes | Unique ID of the product variant |
| quantity | integer | Yes | Quantity of the product |
| user_session_id | string | Yes | Unique session ID for guest/user |

Request Example

```
variant_id=8
quantity=3
user_session_id=postman_guest_124
```

Success Response (200 OK)

```
{
  "status": true,
  "message": "Product added to cart successfully",
  "cart_id": 6,
  "item_qty": "3",
  "user_session_id": "postman_guest_124"
}
```

Error Responses

Validation Error (400)

```
{
  "status": false,
  "message": "Invalid variant or quantity"
}
```

Variant Not Found (404)

```
{
  "status": false,
  "message": "Product variant not found"
}
```

Important Notes

- If the same variant already exists in the cart, quantity may be increased
- Quantity must be greater than 0
- Cart is tracked using `user_session_id`

Version

API Version: v1

If you want, I can also: - Convert this into **Swagger / OpenAPI** format - Create a **Postman collection** - Add **flow diagrams** for OTP login