

Backend API Documentation

This document provides comprehensive documentation for all backend API endpoints including request/response examples and Postman-ready formats.

Base URL: `http://localhost:8000`

1. Authentication APIs

1.1 Login

Authenticate user and get JWT access and refresh tokens.

Method	Endpoint	Auth Required
POST	/api/login/	No

Request Body:

```
{  
    "email": "mani@gmail.com",  
    "password": "Mani@2004"  
}
```

Response (Success - 200):

```
{  
    "success": true,  
    "message": "Login successful",  
    "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  
    "refresh_token": "ICWAVy3_u16fJB3PrYv3hB_qIFb0uTha6Vu-TfKjyVjhczEbtmp0FJp-Dt",  
    "token_type": "Bearer",  
    "expires_in": 900,  
    "user": {  
        "email": "mani@gmail.com",  
        "user_type": "user",  
        "is_admin": false,  
        "permissions": ["read", "write"],  
        "first_name": "John",  
        "last_name": "Doe",  
        "full_name": "John Doe"  
    }  
}
```

Postman Setup:

- Method: POST
- URL: <http://localhost:8000/api/login/>
- Headers: Content-Type: application/json
- Body: Select "raw" and "JSON", then paste the request body above

1.2 Refresh Token

Get a new access token using refresh token.

Method	Endpoint	Auth Required
--------	----------	---------------

POST /api/refresh-token/ No

Request Body:

```
{  
  "refresh_token": "ICWAVy3_u16fJB3PrYv3hB_qIFb0uTha6Vu-TfKjyVjhczEbtmp0FJp-Dt"  
}
```

Response (Success - 200):

```
{  
  "success": true,  
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...".,  
  "token_type": "Bearer",  
  "expires_in": 900,  
  "refresh_token": "new_refresh_token_if_rotated"  
}
```

Postman Setup:

- Method: POST
- URL: <http://localhost:8000/api/refresh-token/>
- Headers: Content-Type: application/json

1.3 Check Authentication

Verify if access token is valid and get user information.

Method	Endpoint	Auth Required
GET	/api/check-auth/	Yes

Headers:

Authorization: Bearer <your_access_token>

Response (Success - 200):

```
{  
  "authenticated": true,  
  "user_email": "mani@gmail.com",  
  "is_admin": false,  
  "permissions": ["read", "write"],  
  "user_name": "John Doe",  
  "user_type": "user",  
  "user_details": {  
    "first_name": "John",  
    "last_name": "Doe",  
    "is_active": true,  
    "created_at": "2024-01-01T00:00:00Z",  
    "updated_at": "2024-01-01T00:00:00Z"  
  }  
}
```

```
    }  
}
```

Postman Setup:

- Method: GET
- URL: <http://localhost:8000/api/check-auth/>
- Headers: Authorization: Bearer <your_access_token>

1.4 Logout

Logout and revoke refresh token.

Method	Endpoint	Auth Required
POST	/api/logout/	No

Request Body:

```
{  
    "refresh_token": "ICWAVy3_u16fJB3PrYv3hB_qIFb0uTha6Vu-TfKjyVjhczEbtmp0FJp-Dt"  
}
```

Response (Success - 200):

```
{  
    "success": true,  
    "message": "Logout successful"  
}
```

Postman Setup:

- Method: POST
- URL: <http://localhost:8000/api/logout/>
- Headers: Content-Type: application/json

2. Customer Dashboard APIs

2.1 List All Customers

Get list of all customers with optional search.

Method	Endpoint	Auth Required
GET	/api/clients/	Yes

Headers:

```
Authorization: Bearer <your_access_token>
```

Query Parameters (Optional):

- **search:** Search term to filter customers (searches in customer_name, company_name, email, phone_number, address)

Example Request:

```
GET http://localhost:8000/api/clients/?search=customer_name
```

Response (Success - 200):

```
{
  "clients": [
    {
      "id": 1,
      "customer_name": "Test Customer",
      "company_name": "Test Company",
      "phone_number": "+1234567890",
      "email": "test@example.com",
      "address": "Test Address",
      "is_active": true,
      "created_at": "2024-01-01T00:00:00Z",
      "updated_at": "2024-01-01T00:00:00Z"
    }
  ],
  "count": 1
}
```

Postman Setup:

- Method: GET
- URL: <http://localhost:8000/api/clients/>
- Headers: Authorization: Bearer <your_access_token>
- For search: Add query parameter "search" with value

2.2 Create Customer

Create a new customer.

Method	Endpoint	Auth Required
POST	/api/clients/	Yes

Headers:

```
Authorization: Bearer <your_access_token>
Content-Type: application/json
```

Request Body:

```
{
  "customer_name": "Test Customer",
  "company_name": "Test Company",
  "phone_number": "+1234567890",
  "email": "test@example.com",
  "address": "Test Address"
}
```

Response (Success - 201):

```
{
  "success": true,
  "client": {
    "id": 1,
    "customer_name": "Test Customer",
    "company_name": "Test Company",
    "phone_number": "+1234567890",
    "email": "test@example.com",
    "address": "Test Address",
    "created_at": "2024-01-01T00:00:00Z",
    "updated_at": "2024-01-01T00:00:00Z"
  }
}
```

Postman Setup:

- Method: POST
- URL: <http://localhost:8000/api/clients/>
- Headers: Authorization: Bearer <your_access_token>, Content-Type: application/json
- Body: Select "raw" and "JSON", then paste the request body above

2.3 Update Customer

Update an existing customer.

Method	Endpoint	Auth Required
PUT	/api/clients/{id}/	Yes

Headers:

```
Authorization: Bearer <your_access_token>
Content-Type: application/json
```

URL Parameters:

- id: Customer ID (integer)

Request Body:

```
{
  "customer_name": "Updated Customer",
  "company_name": "Updated Company",
  "phone_number": "+1234567890",
  "email": "updated@example.com",
  "address": "Updated Address",
  "is_active": true
}
```

Response (Success - 200):

```
{
  "success": true,
  "client": {
    "id": 1,
    "customer_name": "Updated Customer",
    "company_name": "Updated Company",
    "phone_number": "+1234567890",
    "email": "updated@example.com",
    "address": "Updated Address",
    "created_at": "2024-01-01T00:00:00Z",
    "updated_at": "2024-01-01T00:00:00Z"
  }
}
```

Postman Setup:

- Method: PUT
- URL: <http://localhost:8000/api/clients/1/>
- Headers: Authorization: Bearer <your_access_token>, Content-Type: application/json
- Body: Select "raw" and "JSON", then paste the request body above

2.4 Delete Customer

Delete a customer.

Method	Endpoint	Auth Required
DELETE	/api/clients/{id}/	Yes

Headers:

```
Authorization: Bearer <your_access_token>
```

URL Parameters:

- id: Customer ID (integer)

Response (Success - 200):

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

Postman Setup:

- Method: DELETE
- URL: <http://localhost:8000/api/clients/1/>
- Headers: Authorization: Bearer <your_access_token>

3. User Dashboard APIs

3.1 List All Users

Get list of all users with optional search.

Method	Endpoint	Auth Required
GET	/api/users/	Yes

Headers:

```
Authorization: Bearer <your_access_token>
```

Query Parameters (Optional):

- search: Search term to filter users (searches in email, first_name, last_name)

Example Request:

```
GET http://localhost:8000/api/users/?search=user_name
```

Response (Success - 200):

```
{
  "users": [
    {
      "id": 1,
      "email": "user@example.com",
      "first_name": "John",
      "last_name": "Doe",
      "full_name": "John Doe",
      "is_active": true,
      "is_admin": false,
      "permissions": ["read", "write"],
      "created_at": "2024-01-01T00:00:00Z",
      "updated_at": "2024-01-01T00:00:00Z"
    }
  ],
  "count": 1
}
```

Postman Setup:

- Method: GET
- URL: <http://localhost:8000/api/users/>
- Headers: Authorization: Bearer <your_access_token>
- For search: Add query parameter "search" with value

3.2 Create User

Create a new user.

Method	Endpoint	Auth Required
POST	/api/users/	Yes

Headers:

```
Authorization: Bearer <your_access_token>
Content-Type: application/json
```

Request Body:

```
{
  "email": "user@example.com",
  "password": "password123",
  "first_name": "John",
  "last_name": "Doe",
  "is_active": true,
  "is_admin": false,
  "permissions": ["read", "write"]
}
```

Response (Success - 201):

```
{
  "success": true,
  "user": {
    "id": 1,
    "email": "user@example.com",
    "first_name": "John",
    "last_name": "Doe",
    "full_name": "John Doe",
    "is_active": true,
    "is_admin": false,
    "permissions": ["read", "write"],
    "created_at": "2024-01-01T00:00:00Z",
    "updated_at": "2024-01-01T00:00:00Z"
  }
}
```

Postman Setup:

- Method: POST
- URL: <http://localhost:8000/api/users/>
- Headers: Authorization: Bearer <your_access_token>, Content-Type: application/json
- Body: Select "raw" and "JSON", then paste the request body above

3.3 Update User

Update an existing user.

Method	Endpoint	Auth Required
PUT	/api/users/{id}/	Yes

Headers:

```
Authorization: Bearer <your_access_token>
Content-Type: application/json
```

URL Parameters:

- id: User ID (integer)

Request Body:

```
{
  "email": "updated@example.com",
  "first_name": "Jane",
  "last_name": "Smith",
  "is_active": true,
  "is_admin": false,
  "permissions": ["read"]
}
```

Response (Success - 200):

```
{
  "success": true,
  "user": {
    "id": 1,
    "email": "updated@example.com",
    "first_name": "Jane",
    "last_name": "Smith",
    "full_name": "Jane Smith",
    "is_active": true,
    "is_admin": false,
    "permissions": ["read"],
    "created_at": "2024-01-01T00:00:00Z",
    "updated_at": "2024-01-01T00:00:00Z"
  }
}
```

Postman Setup:

- Method: PUT
- URL: <http://localhost:8000/api/users/1/>
- Headers: Authorization: Bearer <your_access_token>, Content-Type: application/json
- Body: Select "raw" and "JSON", then paste the request body above

3.4 Reset User Password

Reset password for a user.

Method	Endpoint	Auth Required
POST	/api/users/{id}/reset-password/	Yes

Headers:

```
Authorization: Bearer <your_access_token>
Content-Type: application/json
```

URL Parameters:

- id: User ID (integer)

Request Body:

```
{
  "new_password": "newpassword123",
  "confirm_password": "newpassword123"
}
```

Response (Success - 200):

```
{
  "success": true,
  "message": "Password reset successfully"
}
```

Postman Setup:

- Method: POST
- URL: http://localhost:8000/api/users/1/reset-password/
- Headers: Authorization: Bearer <your_access_token>, Content-Type: application/json
- Body: Select "raw" and "JSON", then paste the request body above

3.5 Delete User

Delete a user.

Method	Endpoint	Auth Required
DELETE	/api/users/{id}/	Yes

Headers:

```
Authorization: Bearer <your_access_token>
```

URL Parameters:

- id: User ID (integer)

Response (Success - 200):

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

Postman Setup:

- Method: DELETE
- URL: <http://localhost:8000/api/users/1/>
- Headers: Authorization: Bearer <your_access_token>

4. Additional APIs

4.1 Dashboard Statistics

Get dashboard statistics including KPIs and chart data.

Method	Endpoint	Auth Required
GET	/api/dashboard-stats/	Yes

Headers:

```
Authorization: Bearer <your_access_token>
```

Query Parameters (Optional):

- year: Year for filtering statistics (default: current year)

Response (Success - 200):

```
{
  "success": true,
  "data": {
    "kpis": {
      "total_quotations": 100,
      "total_customers": 50,
      "active_customers": 45,
      "inactive_customers": 5,
      "total_users": 10
    },
    "monthly_sends": [
      {
        "month": "JAN",
        "email": 10,
        "whatsapp": 5,
        "total": 15
      }
    ],
    "send_breakdown": {
      "email": {
        "count": 80,
        "percentage": 80.0,
        "grand_total": 50000.00
      },
      "whatsapp": {
        "count": 20,
        "percentage": 20.0,
        "grand_total": 10000.00
      },
      "total": 100,
      "total_grand_total": 60000.00
    },
    "customers": [
      {
        "id": 1,
        "customer_name": "Customer Name",
        "company_name": "Company Name",
        "email": "customer@example.com",
        "status": "Active"
      }
    ]
  }
}
```

```

    "phone_number": "+1234567890",
    "total_quotation": 5,
    "status": "Active",
    "user_breakdown": [
        {
            "user_name": "Admin",
            "count": 3
        },
        {
            "user_name": "John Doe",
            "count": 2
        }
    ]
},
"year": 2024
}
}

```

4.2 Get Company Details

Get all company details for settings page.

Method	Endpoint	Auth Required
GET	/api/company-details/	Yes

Headers:

Authorization: Bearer <your_access_token>

Response (Success - 200):

```
{
  "success": true,
  "company": {
    "company_name": "MAKLOGISTICS",
    "email": "maklogistics@gmail.com",
    "tagline": "DIGITAL SOLUTION ARCHITECTS",
    "phone_number": "9042510714",
    "address": "TBI@TCE, Thiruparankundaram, Madurai - 625 015",
    "sendemail": "send@example.com",
    "sendpassword": "password",
    "sendnumber": "+1234567890",
    "openrouter_api_key": "sk-or-vl-...",
    "openrouter_model": "google/gemini-flash-1.5:free",
    "login_logo_url": "/media/company_logos/logo.png",
    "login_image_url": "/media/company_login/image.png",
    "quotation_logo_url": "/media/company_quotation/logo.png",
    "created_at": "2024-01-01T00:00:00Z",
    "updated_at": "2024-01-01T00:00:00Z"
  }
}
```

4.3 Update Company Details

Update company details.

Method	Endpoint	Auth Required
PUT	/api/company-details/update/	Yes

Headers:

```
Authorization: Bearer <your_access_token>
Content-Type: application/json
```

Request Body:

```
{
  "company_name": "Updated Company",
  "email": "updated@example.com",
  "tagline": "New Tagline",
  "phone_number": "+1234567890",
  "address": "New Address",
  "sendemail": "send@example.com",
  "sendpassword": "password",
  "sendnumber": "+1234567890",
  "openrouter_api_key": "sk-or-v1-...",
  "openrouter_model": "google/gemini-flash-1.5:free"
}
```

4.4 Send Quotation Email

Send quotation PDF via email using SMTP.

Method	Endpoint	Auth Required
POST	/api/send-quotation-email/	Yes

Headers:

```
Authorization: Bearer <your_access_token>
Content-Type: application/json
```

Request Body:

```
{
  "recipient_email": "customer@example.com",
  "customer_name": "Customer Name",
  "pdf_base64": "base64_encoded_pdf_string",
  "pdf_filename": "quotation.pdf"
}
```

Response (Success - 200):

```
{
  "success": true,
  "message": "Quotation sent successfully to customer@example.com"
}
```

4.5 Get Company Info

Get company information for header display.

Method	Endpoint	Auth Required
GET	/api/company-info/	No

Response (Success - 200):

```
{  
    "company_name": "MAKLOGISTICS",  
    "tagline": "DIGITAL SOLUTION ARCHITECTS",  
    "website": "",  
    "phone_number": "9042510714",  
    "email": "maklogistics@gmail.com",  
    "address": "TBI@TCE, Thiruparankundaram, Madurai - 625 015",  
    "logo_url": "/media/company_quotation/logo.png"  
}
```

4.6 Get Company Login

Get company login credentials and images for login page.

Method	Endpoint	Auth Required
GET	/api/company-login/	No

Response (Success - 200):

```
{  
    "email": "maklogistics@gmail.com",  
    "login_logo_url": "/media/company_logos/logo.png",  
    "login_image_url": "/media/company_login/image.png"  
}
```

5. Quotation Management APIs

5.1 Chat

Chat with AI for quotation generation.

Method	Endpoint	Auth Required
POST	/api/chat/	Yes

Headers:

```
Authorization: Bearer <your_access_token>
Content-Type: application/json
```

Request Body:

```
{
  "message": "Add a web development service for $5000"
}
```

Response (Success - 200):

```
{
  "response": "I've added the web development service to your quotation.",
  "quotation": {
    "services": [
      {
        "name": "Web Development",
        "price": 5000,
        "quantity": 1,
        "gst": 18,
        "total": 5900
      }
    ],
    "grand_total": 5900
  }
}
```

5.2 Get Quotation

Get current quotation from session.

Method	Endpoint	Auth Required
GET	/api/quotation/	Yes

Headers:

```
Authorization: Bearer <your_access_token>
```

5.3 Reset Quotation

Reset quotation to empty state.

Method	Endpoint	Auth Required
POST	/api/reset/	Yes

5.4 Sync Quotation

Sync quotation state from frontend.

Method	Endpoint	Auth Required
POST	/api-sync-quotation/	Yes

5.5 Get Conversation History

Get conversation history from session.

Method	Endpoint	Auth Required
GET	/api/conversation-history/	Yes

5.6 Sync Conversation History

Sync conversation history from frontend.

Method	Endpoint	Auth Required
POST	/api-sync-conversation-history/	Yes

6. Error Responses

All APIs may return the following error responses:

6.1 400 Bad Request

```
{  
  "error": "Error message describing what went wrong"  
}
```

6.2 401 Unauthorized

```
{  
  "error": "Invalid email or password"  
}
```

6.3 404 Not Found

```
{  
  "error": "Client not found"  
}
```

6.4 500 Internal Server Error

```
{  
  "error": "Error message describing the server error"  
}
```

7. Authentication Notes

- Access tokens expire in 15 minutes (900 seconds)
- Use the refresh token endpoint to get a new access token
- Include the access token in the Authorization header for protected endpoints
- Format: Authorization: Bearer <access_token>
- Both company login and user login are supported
- Company login has admin privileges by default
- User login permissions are based on the user's assigned permissions

8. Postman Collection Tips

1. Create a new Postman Collection named "Kattappa API"

2. Set collection variable: base_url = http://localhost:8000

3. Create a folder structure:

- Authentication
- Customer Dashboard
- User Dashboard
- Quotation Management
- Additional APIs

4. For authentication:

- First call Login API
- Save access_token and refresh_token as collection variables
- Use {{access_token}} in Authorization header for protected endpoints

5. Set up Pre-request Script for token refresh (optional):

```
// Check if token is expired and refresh if needed
// This is a simplified example
if (pm.collectionVariables.get("token_expiry") < Date.now()) {
    // Call refresh token endpoint
}
```