API Documentation – Shipping Transport System

# 1. General Information

API Name: Shipping Transport API  
Version: v1.0  
Base URL (Production): http://13.48.130.179:4000/api  
Description:  
This API enables user management, freight tracking, warehouse operations, invoice processing, and notification handling. All endpoints return JSON responses and support secure communication over HTTPS.  
  
Contact Information:  
- Support Email: support@yourdomain.com  
- Developer Contact: devteam@yourdomain.com

# 2. Authentication & Security

Authentication Type: JWT (JSON Web Token)  
How to Authenticate:  
- Obtain a token by registering (/user/register) or logging in (/auth/login).  
- Include the token in the Authorization header for all authenticated endpoints:  
 Authorization: Bearer <your\_token\_here>  
  
Security Recommendations:  
- Always use HTTPS to secure API requests.  
- Do not expose JWT tokens in URLs or logs.  
- Enforce strong password policies.  
- Implement role-based access control for different endpoints.

# 3. Conventions

Request Format: JSON  
Response Format: JSON  
HTTP Status Codes Used:  
- 200 OK – Request successful  
- 201 Created – Resource successfully created  
- 400 Bad Request – Invalid request or validation error  
- 401 Unauthorized – Invalid or missing authentication token  
- 403 Forbidden – Access denied  
- 404 Not Found – Resource not found  
- 500 Internal Server Error – Unexpected server error  
  
Field Naming: camelCase for API request and response fields.  
Timestamps: UTC in ISO 8601 format (YYYY-MM-DDTHH:mm:ssZ).

# 4. Endpoint Catalog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Category | Endpoint | Method | Description | Auth Required |
| User | /user/register | POST | Register user and return JWT token | No |
| User | /auth/login | POST | Login and return JWT token | No |
| Freight | /freight/add | POST | Add freight details | Yes |
| Freight | /freight/:id | GET | Get freight details by ID | Yes |
| Warehouse | /warehouse/add | POST | Add new warehouse | Yes |
| Batches | /batches/edit/:id | PUT | Edit batch details | Yes |
| Batches | /batches/delete/:id | DELETE | Delete batch record | Yes |
| Invoices | /invoice/:clientId | GET | Get invoices by client ID | Yes |
| Notifications | /notifications/send | POST | Send notification to client or admin | Yes |
| Claims | /getClaim | GET | Get claims based on user ID and role | Yes |

# 5. Endpoint Details

## User Registration & Token

Method: POST  
URL: http://13.48.130.179:4000/api/user/register  
Authentication: No authentication required  
  
Request Body:  
{  
 "email": "user@example.com",  
 "password": "yourpassword"  
}  
  
Success Response (200):  
{  
 "success": true,  
 "message": "User registered successfully",  
 "token": "eyJhbGciOiJIUzI1NiIsInR..."  
}  
  
Error Response (400 – Validation Error):  
{  
 "success": false,  
 "errors": [  
 "\"email\" is not allowed to be empty",  
 "\"password\" is not allowed to be empty"  
 ]  
}  
  
Error Response (400 – Duplicate Email):  
{  
 "success": false,  
 "message": "Email already in use"  
}  
  
Error Response (500 – Server Error):  
{  
 "success": false,  
 "message": "Internal server error"  
}  
  
Notes:  
- After registration, use the returned token for authenticated endpoints:  
 Authorization: Bearer <token>

# 6. Data Models

Include JSON models for User, Auth Token, Freight, Warehouse, Invoices, etc.

# 7. Use Cases / Workflows

Document real-world flows, such as user onboarding and freight creation.

# 8. Environments & Integration

Production URL: http://13.48.130.179:4000/api  
Staging URL: (add if available)  
Testing Tools: Postman, Curl, Swagger UI  
Third-party Integrations: Nodemailer, Puppeteer, MySQL

# 9. Changelog

v1.0 – 2025-09-03  
- Added /user/register endpoint  
- Added validation and duplicate email checks  
- Implemented freight, warehouse, invoice, and notification endpoints