Travel Go API Documentation

Base URL

```
http://localhost:3000/api
```

Authentication

Login

```
POST /auth/login
```

Request Body:

```
{
  "email": "string",
  "password": "string"
}
```

Response (200):

```
{
  "token": "string",
  "user": {
    "id": "number",
    "email": "string",
    "firstName": "string",
    "lastName": "string",
    "role": "CLIENT | SALES_AGENT | ADMIN",
    "country": "string",
    "phone": "string | null"
  }
}
```

Register

```
POST /auth/register
```

Request Body:

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "country": "string",
```

```
  "phone": "string" // optional
}
```

Response (201):

```
{
  "id": "number",
  "email": "string",
  "firstName": "string",
  "lastName": "string",
  "role": "CLIENT",
  "country": "string",
  "phone": "string | null"
}
```

Users

Get User Profile

```
GET /users/profile
```

Headers:

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

Response (200):

```
{
  "id": "number",
  "email": "string",
  "firstName": "string",
  "lastName": "string",
  "role": "CLIENT | SALES_AGENT | ADMIN",
  "country": "string",
  "phone": "string | null"
}
```

Update User Profile

```
PUT /users/profile
```

Headers:

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

Request Body:

```
{
  "firstName": "string", // optional
  "lastName": "string", // optional
  "country": "string", // optional
  "phone": "string" // optional
}
```

Response (200):

```
{
  "id": "number",
  "email": "string",
  "firstName": "string",
  "lastName": "string",
  "role": "CLIENT | SALES_AGENT | ADMIN",
  "country": "string",
  "phone": "string | null"
}
```

Products

List Products

```
GET /products
```

Query Parameters:

- `page` (optional): number (default: 1)
- `limit` (optional): number (default: 10)
- `search` (optional): string
- `type` (optional): "FLIGHT" | "HOTEL" | "TRANSFER" | "ACTIVITY" | "INSURANCE"

Response (200):

```
{
  "items": [
    {
      "id": "number",
      "name": "string",
      "description": "string",
      "type": "FLIGHT | HOTEL | TRANSFER | ACTIVITY | INSURANCE",
      "price": "number",
      "currency": "string",
      "provider": "string",
      "availability": "boolean"
    }
  ],
  "total": "number",
  "page": "number",
```

```
  "limit": "number"
}
```

Get Product Details

```
GET /products/{id}
```

Response (200):

```
{
  "id": "number",
  "name": "string",
  "description": "string",
  "type": "FLIGHT | HOTEL | TRANSFER | ACTIVITY | INSURANCE",
  "price": "number",
  "currency": "string",
  "provider": "string",
  "availability": "boolean",
  "details": {
    // Specific details based on product type
  }
}
```

Create Product (Admin/Sales Agent)

```
POST /products
```

Headers:

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

Request Body:

```
{
  "name": "string",
  "description": "string",
  "type": "FLIGHT | HOTEL | TRANSFER | ACTIVITY | INSURANCE",
  "price": "number",
  "currency": "string",
  "provider": "string",
  "details": {
    // Specific details based on product type
  }
}
```

Response (201):

```
{
  "id": "number",
  "name": "string",
  "description": "string",
  "type": "FLIGHT | HOTEL | TRANSFER | ACTIVITY | INSURANCE",
  "price": "number",
  "currency": "string",
  "provider": "string",
  "availability": "boolean",
  "details": {
    // Specific details based on product type
  }
}
```

Packages

List Packages

```
GET /packages
```

Query Parameters:

- `page` (optional): number (default: 1)
- `limit` (optional): number (default: 10)
- `search` (optional): string

Response (200):

```
{
  "items": [
    {
      "id": "number",
      "name": "string",
      "description": "string",
      "totalPrice": "number",
      "currency": "string",
      "products": [
        {
          "id": "number",
          "type": "FLIGHT | HOTEL | TRANSFER | ACTIVITY | INSURANCE",
          "name": "string"
        }
      ]
    }
  ],
  "total": "number",
  "page": "number",
  "limit": "number"
}
```

Get Package Details

```
GET /packages/{id}
```

Response (200):

```
{
  "id": "number",
  "name": "string",
  "description": "string",
  "totalPrice": "number",
  "currency": "string",
  "products": [
    {
      "id": "number",
      "type": "FLIGHT | HOTEL | TRANSFER | ACTIVITY | INSURANCE",
      "name": "string",
      "price": "number",
      "details": {
        // Specific details based on product type
      }
    }
  ]
}
```

Create Package (Admin/Sales Agent)

```
POST /packages
```

Headers:

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

Request Body:

```
{
  "name": "string",
  "description": "string",
  "productIds": ["number"]
}
```

Response (201):

```
{
  "id": "number",
  "name": "string",
  "description": "string",
  "totalPrice": "number",
```

```
  "currency": "string",
  "products": [
    {
      "id": "number",
      "type": "FLIGHT | HOTEL | TRANSFER | ACTIVITY | INSURANCE",
      "name": "string",
      "price": "number"
    }
  ]
}
```

Cart

Get Cart

```
GET /cart
```

Headers:

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

Response (200):

```
{
  "items": [
    {
      "id": "number",
      "quantity": "number",
      "product": {
        "id": "number",
        "name": "string",
        "price": "number",
        "currency": "string"
      }
    }
  ],
  "totalAmount": "number",
  "currency": "string"
}
```

Add to Cart

```
POST /cart/items
```

Headers:

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

Request Body:

```
{
  "productId": "number",
  "quantity": "number"
}
```

Response (200):

```
{
  "items": [
    {
      "id": "number",
      "quantity": "number",
      "product": {
        "id": "number",
        "name": "string",
        "price": "number",
        "currency": "string"
      }
    }
  ],
  "totalAmount": "number",
  "currency": "string"
}
```

Update Cart Item

```
PUT /cart/items/{id}
```

Headers:

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

Request Body:

```
{
  "quantity": "number"
}
```

Response (200):

```
{
  "items": [
    {
      "id": "number",
      "quantity": "number",
      "product": {
        "id": "number",
        "name": "string",
```

```
        "price": "number",
        "currency": "string"
      }
    }
  ],
  "totalAmount": "number",
  "currency": "string"
}
```

Remove from Cart

```
DELETE /cart/items/{id}
```

Headers:

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

Response (200):

```
{
  "items": [
    {
      "id": "number",
      "quantity": "number",
      "product": {
        "id": "number",
        "name": "string",
        "price": "number",
        "currency": "string"
      }
    }
  ],
  "totalAmount": "number",
  "currency": "string"
}
```

Orders

Create Order

```
POST /orders
```

Headers:

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

Request Body:

```
{
  "billingInfo": {
    "firstName": "string",
    "lastName": "string",
    "email": "string",
    "phone": "string",
    "address": "string",
    "city": "string",
    "country": "string",
    "postalCode": "string"
  },
  "couponCode": "string" // optional
}
```

Response (201):

```
{
  "id": "number",
  "orderNumber": "string",
  "status": "PENDING",
  "totalAmount": "number",
  "currency": "string",
  "items": [
    {
      "product": {
        "id": "number",
        "name": "string",
        "price": "number"
      },
      "quantity": "number",
      "subtotal": "number"
    }
  ],
  "billingInfo": {
    "firstName": "string",
    "lastName": "string",
    "email": "string",
    "phone": "string",
    "address": "string",
    "city": "string",
    "country": "string",
    "postalCode": "string"
  }
}
```

Get Order Details

```
GET /orders/{id}
```

Headers:

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

Response (200):

```
{
  "id": "number",
  "orderNumber": "string",
  "status": "PENDING | PAID | CONFIRMED | CANCELLED",
  "totalAmount": "number",
  "currency": "string",
  "items": [
    {
      "product": {
        "id": "number",
        "name": "string",
        "price": "number"
      },
      "quantity": "number",
      "subtotal": "number"
    }
  ],
  "billingInfo": {
    "firstName": "string",
    "lastName": "string",
    "email": "string",
    "phone": "string",
    "address": "string",
    "city": "string",
    "country": "string",
    "postalCode": "string"
  },
  "payments": [
    {
      "id": "number",
      "amount": "number",
      "currency": "string",
      "status": "PENDING | COMPLETED | FAILED",
      "provider": "string",
      "createdAt": "string"
    }
  ]
}
```

List User Orders

```
GET /orders
```

Headers:

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

Query Parameters:

- `page` (optional): number (default: 1)
- `limit` (optional): number (default: 10)
- `status` (optional): "PENDING" | "PAID" | "CONFIRMED" | "CANCELLED"

Response (200):

```
{
  "items": [
    {
      "id": "number",
      "orderNumber": "string",
      "status": "PENDING | PAID | CONFIRMED | CANCELLED",
      "totalAmount": "number",
      "currency": "string",
      "createdAt": "string"
    }
  ],
  "total": "number",
  "page": "number",
  "limit": "number"
}
```

Payments

Create Payment

```
POST /payments/orders/{orderId}
```

Headers:

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

Request Body:

```
{
  "provider": "MERCADOPAGO",
  "paymentMethod": "CREDIT_CARD | DEBIT_CARD | BANK_TRANSFER",
  "paymentDetails": {
    // Provider-specific payment details
  }
}
```

Response (201):

```
{
  "id": "number",
  "orderId": "number",
  "amount": "number",
  "currency": "string",
  "status": "PENDING",
  "provider": "string",
  "paymentUrl": "string", // URL to complete payment if needed
  "createdAt": "string"
}
```

Get Payment Status

```
GET /payments/{id}
```

Headers:

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

Response (200):

```
{
  "id": "number",
  "orderId": "number",
  "amount": "number",
  "currency": "string",
  "status": "PENDING | COMPLETED | FAILED",
  "provider": "string",
  "createdAt": "string",
  "completedAt": "string | null"
}
```

Notifications

Get User Notifications

```
GET /notifications
```

Headers:

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

Query Parameters:

- `page` (optional): number (default: 1)
- `limit` (optional): number (default: 50)
- `unreadOnly` (optional): boolean

Response (200):

```
{
  "items": [
    {
      "id": "number",
      "type": "ORDER_CREATED | ORDER_PAID | ORDER_STATUS_UPDATE | TRAVEL_REMINDER",
      "message": "string",
      "read": "boolean",
      "createdAt": "string"
    }
  ],
  "total": "number",
  "page": "number",
  "limit": "number"
}
```

Mark Notification as Read

```
PUT /notifications/{id}/read
```

Headers:

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

Response (200):

```
{
  "id": "number",
  "type": "ORDER_CREATED | ORDER_PAID | ORDER_STATUS_UPDATE | TRAVEL_REMINDER",
  "message": "string",
  "read": true,
  "createdAt": "string"
}
```

Error Responses

400 Bad Request

```
{
  "error": "string",
  "message": "string"
}
```

401 Unauthorized

```
{
  "error": "Unauthorized",
```

```
  "message": "Invalid or expired token"
}
```

403 Forbidden

```
{
  "error": "Forbidden",
  "message": "Insufficient permissions"
}
```

404 Not Found

```
{
  "error": "Not Found",
  "message": "Resource not found"
}
```

500 Internal Server Error

```
{
  "error": "Internal Server Error",
  "message": "An unexpected error occurred"
}
```

Implementation Details

Authentication

- JWT-based authentication
- Tokens expire after 24 hours
- Refresh tokens not implemented (user must login again)
- Password hashing using bcrypt

Authorization

Three roles with different permissions:

1. **CLIENT**
    - Can view products and packages
    - Can manage their own cart
    - Can create and view their own orders
    - Can view their own notifications
    - Can update their own profile
2. **SALES_AGENT**
    - All CLIENT permissions
    - Can create and update products
    - Can create and update packages

    - Can view all orders
    - Can update order status
    - Can view customer information

3. **ADMIN**
    - All SALES_AGENT permissions
    - Can manage users
    - Can view system statistics
    - Can configure system settings
    - Can manage email templates

Database Schema

- SQLite database using Prisma ORM
- Key tables:
    - users
    - products
    - packages
    - cart_items
    - orders
    - order_items
    - payments
    - notifications
    - email_config

Email Notifications

- Using nodemailer for email delivery
- Configurable templates per notification type
- Support for multiple recipients
- HTML and text email formats
- Environment-based SMTP configuration

Payment Integration

- MercadoPago integration (primary payment provider)
- Support for multiple payment methods
- Webhook handling for payment status updates
- Automatic order status updates
- Payment retry mechanism

Security Measures

1. **Input Validation**
    - Request body validation using Zod
    - Query parameter sanitization
    - File upload restrictions

2. **Rate Limiting**
    - API rate limiting per IP
    - More strict limits for authentication endpoints

3. **Security Headers**

    - CORS configuration
    - Helmet middleware
    - XSS protection
    - CSRF protection

4. **Error Handling**
    - Centralized error handling
    - Sanitized error messages
    - Detailed logging for debugging

Caching Strategy

- Redis cache for:
    - Product listings
    - Package listings
    - User sessions
    - Cart data
- Cache invalidation on updates
- TTL-based expiration

Monitoring

- Request logging
- Error tracking
- Performance metrics
- Health check endpoint

Testing

- Unit tests for services
- Integration tests for APIs
- End-to-end tests for critical flows
- Test coverage reporting