

Maritime Reservation System API Documentation

Overview

The Maritime Reservation System API provides comprehensive endpoints for managing ferry bookings, user authentication, payment processing, and administrative operations. This RESTful API is built with FastAPI and follows OpenAPI 3.0 specifications.

Base URL: `https://api.maritime-reservations.com/api/v1`

API Version: 1.0.0

Authentication

The API uses JWT (JSON Web Token) based authentication. Include the token in the Authorization header:

```
Authorization: Bearer <your_jwt_token>
```

Obtaining a Token

POST `/auth/login`

```
{
  "username": "user@example.com",
  "password": "your_password"
}
```

Response:

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "bearer",
  "expires_in": 3600
}
```

User Management

Register New User

POST /auth/register

Creates a new user account.

Request Body:

```
{
  "email": "user@example.com",
  "password": "secure_password",
  "first_name": "John",
  "last_name": "Doe",
  "phone": "+1234567890",
  "preferred_language": "en"
}
```

Response (201 Created):

```
{
  "id": "user_123",
  "email": "user@example.com",
  "first_name": "John",
  "last_name": "Doe",
  "phone": "+1234567890",
  "preferred_language": "en",
  "created_at": "2024-06-04T10:30:00Z",
  "is_active": true
}
```

Get Current User

GET /users/me

Retrieves the current authenticated user's information.

Headers: Authorization: Bearer <token>

Response (200 OK):

```
{
  "id": "user_123",
  "email": "user@example.com",
}
```

```
"first_name": "John",
"last_name": "Doe",
"phone": "+1234567890",
"preferred_language": "en",
"created_at": "2024-06-04T10:30:00Z",
"booking_count": 5,
"total_spent": "€450.00"
}
```

Update User Profile

PUT /users/me

Updates the current user's profile information.

Headers: Authorization: Bearer <token>

Request Body:

```
{
  "first_name": "John",
  "last_name": "Smith",
  "phone": "+1234567891",
  "preferred_language": "fr"
}
```

Ferry Search and Booking

Search Ferries

GET /ferries/search

Searches for available ferry routes based on criteria.

Query Parameters: - `departure_port` (required): Departure port name - `arrival_port` (required): Arrival port name - `departure_date` (required): Date in YYYY-MM-DD format - `return_date` (optional): Return date for round-trip - `passengers` (required): Number of passengers (1-9) - `vehicle` (optional): Include vehicle (true/false)

Example Request:

GET /ferries/search?

departure_port=Tunis&arrival_port=Marseille&departure_date=2024-07-15&pass

Response (200 OK):

```
[
  {
    "id": "ferry_123",
    "operator": "CTN",
    "vessel_name": "Carthage",
    "departure_port": "Tunis",
    "arrival_port": "Marseille",
    "departure_time": "08:00",
    "arrival_time": "14:00",
    "duration": "6h 00m",
    "price": {
      "adult": "€89.00",
      "child": "€45.00",
      "vehicle": "€120.00"
    },
    "available_seats": 150,
    "amenities": ["restaurant", "wifi", "deck", "cabin"],
    "booking_class": "economy"
  },
  {
    "id": "ferry_456",
    "operator": "GNV",
    "vessel_name": "GNV Excellent",
    "departure_port": "Tunis",
    "arrival_port": "Marseille",
    "departure_time": "20:00",
    "arrival_time": "08:00+1",
    "duration": "12h 00m",
    "price": {
      "adult": "€95.00",
      "child": "€48.00",
      "vehicle": "€135.00"
    },
    "available_seats": 200,
    "amenities": ["restaurant", "wifi", "cabin", "pool"],
    "booking_class": "comfort"
  }
]
```

Get Ferry Details

GET /ferries/{ferry_id}

Retrieves detailed information about a specific ferry.

Response (200 OK):

```
{
  "id": "ferry_123",
  "operator": "CTN",
  "vessel_name": "Carthage",
  "vessel_info": {
    "length": "186m",
    "capacity": 1800,
    "year_built": 2010,
    "flag": "Tunisia"
  },
  "route_info": {
    "departure_port": "Tunis",
    "arrival_port": "Marseille",
    "distance": "750km",
    "frequency": "daily"
  },
  "amenities": [
    {
      "name": "restaurant",
      "description":
"Full-service restaurant with Mediterranean cuisine"
    },
    {
      "name": "wifi",
      "description": "Free WiFi throughout the vessel"
    }
  ],
  "deck_plans": [
    {
      "deck": "7",
      "facilities": ["restaurant", "bar", "shop"]
    },
    {
      "deck": "8",
      "facilities": ["cabins", "lounge"]
    }
  ]
}
```

Booking Management

Create Booking

POST /bookings

Creates a new ferry booking.

Headers: Authorization: Bearer <token>

Request Body:

```
{
  "ferry_id": "ferry_123",
  "departure_date": "2024-07-15",
  "return_date": null,
  "passengers": [
    {
      "type": "adult",
      "first_name": "John",
      "last_name": "Doe",
      "date_of_birth": "1990-01-01",
      "passport_number": "AB123456",
      "nationality": "US"
    },
    {
      "type": "adult",
      "first_name": "Jane",
      "last_name": "Doe",
      "date_of_birth": "1992-05-15",
      "passport_number": "CD789012",
      "nationality": "US"
    }
  ],
  "vehicle": {
    "type": "car",
    "license_plate": "ABC123",
    "length": "4.5m",
    "height": "1.8m"
  },
  "special_requests": "Wheelchair accessible cabin",
  "contact_email": "john.doe@example.com",
  "contact_phone": "+1234567890"
}
```

Response (201 Created):

```
{
  "booking_id": "BK001234",
  "status": "pending",
  "total_amount": "€298.00",
  "currency": "EUR",
  "expires_at": "2024-06-04T11:30:00Z",
  "payment_required": true,
  "booking_reference": "CTN-BK001234",
}
```

```

    "passengers": [
      {
        "id": "pax_001",
        "first_name": "John",
        "last_name": "Doe",
        "ticket_number": "TKT001234001"
      },
      {
        "id": "pax_002",
        "first_name": "Jane",
        "last_name": "Doe",
        "ticket_number": "TKT001234002"
      }
    ]
  }
}

```

Get User Bookings

GET /bookings

Retrieves all bookings for the authenticated user.

Headers: Authorization: Bearer <token>

Query Parameters: - `status` (optional): Filter by status (pending, confirmed, cancelled) - `limit` (optional): Number of results (default: 20) - `offset` (optional): Pagination offset (default: 0)

Response (200 OK):

```

{
  "bookings": [
    {
      "booking_id": "BK001234",
      "status": "confirmed",
      "ferry": {
        "operator": "CTN",
        "vessel_name": "Carthage",
        "departure_port": "Tunis",
        "arrival_port": "Marseille"
      },
      "departure_date": "2024-07-15",
      "departure_time": "08:00",
      "passengers_count": 2,
      "total_amount": "€298.00",
      "booking_date": "2024-06-04T10:30:00Z"
    }
  ],
  "total": 1,
}

```

```
"limit": 20,  
"offset": 0  
}
```

Get Booking Details

GET /bookings/{booking_id}

Retrieves detailed information about a specific booking.

Headers: Authorization: Bearer <token>

Response (200 OK):

```
{  
  "booking_id": "BK001234",  
  "status": "confirmed",  
  "booking_reference": "CTN-BK001234",  
  "ferry": {  
    "id": "ferry_123",  
    "operator": "CTN",  
    "vessel_name": "Carthage",  
    "departure_port": "Tunis",  
    "arrival_port": "Marseille",  
    "departure_time": "08:00",  
    "arrival_time": "14:00"  
  },  
  "travel_details": {  
    "departure_date": "2024-07-15",  
    "return_date": null,  
    "is_round_trip": false  
  },  
  "passengers": [  
    {  
      "id": "pax_001",  
      "type": "adult",  
      "first_name": "John",  
      "last_name": "Doe",  
      "ticket_number": "TKT001234001",  
      "seat_number": "A12"  
    }  
  ],  
  "vehicle": {  
    "type": "car",  
    "license_plate": "ABC123",  
    "deck_assignment": "Deck 2"  
  },  
  "payment": {  
    "total_amount": "€298.00",  

```



```
    "currency": "EUR",
    "status": "paid",
    "payment_method": "stripe_card",
    "payment_date": "2024-06-04T10:35:00Z"
  },
  "contact_info": {
    "email": "john.doe@example.com",
    "phone": "+1234567890"
  },
  "created_at": "2024-06-04T10:30:00Z",
  "updated_at": "2024-06-04T10:35:00Z"
}
```

Cancel Booking

DELETE /bookings/{booking_id}

Cancels a booking and processes refund if applicable.

Headers: Authorization: Bearer <token>

Response (200 OK):

```
{
  "booking_id": "BK001234",
  "status": "cancelled",
  "cancellation_date": "2024-06-04T15:30:00Z",
  "refund": {
    "amount": "€268.20",
    "currency": "EUR",
    "processing_fee": "€29.80",
    "refund_method": "original_payment_method",
    "estimated_processing_time": "3-5 business days"
  }
}
```

Payment Processing

Create Payment

POST /payments

Initiates a payment for a booking.

Headers: Authorization: Bearer <token>

Request Body:

```
{
  "booking_id": "BK001234",
  "payment_method": "stripe_card",
  "return_url": "https://example.com/payment/success",
  "cancel_url": "https://example.com/payment/cancel"
}
```

Response (201 Created):

```
{
  "payment_id": "pay_123456",
  "status": "pending",
  "amount": "€298.00",
  "currency": "EUR",
  "payment_method": "stripe_card",
  "redirect_url": "https://checkout.stripe.com/pay/cs_123456",
  "expires_at": "2024-06-04T11:30:00Z"
}
```

Get Payment Status

GET /payments/{payment_id}

Retrieves the current status of a payment.

Headers: Authorization: Bearer <token>

Response (200 OK):

```
{
  "payment_id": "pay_123456",
  "booking_id": "BK001234",
  "status": "completed",
  "amount": "€298.00",
  "currency": "EUR",
  "payment_method": "stripe_card",
  "transaction_id": "pi_1234567890",
  "payment_date": "2024-06-04T10:35:00Z",
  "receipt_url": "https://pay.stripe.com/receipts/123456"
}
```

Process Refund

POST /payments/{payment_id}/refund

Processes a refund for a completed payment.

Headers: Authorization: Bearer <token>

Request Body:

```
{
  "amount": "€298.00",
  "reason": "customer_request"
}
```

Response (200 OK):

```
{
  "refund_id": "ref_123456",
  "payment_id": "pay_123456",
  "amount": "€298.00",
  "currency": "EUR",
  "status": "processing",
  "estimated_completion": "2024-06-07T10:35:00Z"
}
```

Administrative Endpoints

Get Booking Statistics

GET /admin/stats/bookings

Retrieves booking statistics for administrative dashboard.

Headers: Authorization: Bearer <admin_token>

Query Parameters: - period (optional): Time period (7d, 30d, 90d, 1y) - group_by (optional): Grouping (day, week, month)

Response (200 OK):

```
{
  "total_bookings": 2847,
  "confirmed_bookings": 2456,
  "cancelled_bookings": 391,
}
```

```
{
  "total_revenue": "€284750.00",
  "average_booking_value": "€115.90",
  "trends": [
    {
      "date": "2024-06-01",
      "bookings": 45,
      "revenue": "€5220.00"
    },
    {
      "date": "2024-06-02",
      "bookings": 52,
      "revenue": "€6030.00"
    }
  ]
}
```

Manage Bookings

GET /admin/bookings

Retrieves all bookings for administrative management.

Headers: Authorization: Bearer <admin_token>

Query Parameters: - status (optional): Filter by status - operator (optional): Filter by ferry operator - date_from (optional): Start date filter - date_to (optional): End date filter - search (optional): Search term - limit (optional): Results per page - offset (optional): Pagination offset

Response (200 OK):

```
{
  "bookings": [
    {
      "booking_id": "BK001234",
      "customer": {
        "name": "John Doe",
        "email": "john.doe@example.com"
      },
      "ferry": {
        "operator": "CTN",
        "route": "Tunis → Marseille"
      },
      "departure_date": "2024-07-15",
      "status": "confirmed",
      "amount": "€298.00",
      "booking_date": "2024-06-04T10:30:00Z"
    }
  ]
}
```

```
],  
  "total": 2847,  
  "limit": 50,  
  "offset": 0  
}
```

Error Handling

The API uses standard HTTP status codes and returns error details in JSON format.

Error Response Format

```
{  
  "error": {  
    "code": "VALIDATION_ERROR",  
    "message": "Invalid input data",  
    "details": [  
      {  
        "field": "departure_date",  
        "message": "Date must be in the future"  
      }  
    ],  
    "request_id": "req_123456789"  
  }  
}
```

Common Error Codes

Status Code	Error Code	Description
400	VALIDATION_ERROR	Invalid request data
401	UNAUTHORIZED	Authentication required
403	FORBIDDEN	Insufficient permissions
404	NOT_FOUND	Resource not found
409	CONFLICT	Resource conflict
422	UNPROCESSABLE_ENTITY	Validation failed
429	RATE_LIMITED	Too many requests
500	INTERNAL_ERROR	Server error

Rate Limiting

The API implements rate limiting to ensure fair usage:

- **General endpoints:** 100 requests per minute per IP
- **Search endpoints:** 50 requests per minute per IP
- **Authentication endpoints:** 10 requests per minute per IP

Rate limit headers are included in responses:

```
X-RateLimit-Limit: 100
X-RateLimit-Remaining: 95
X-RateLimit-Reset: 1623456789
```

Webhooks

The system supports webhooks for real-time notifications of booking and payment events.

Webhook Events

- `booking.created` - New booking created
- `booking.confirmed` - Booking confirmed
- `booking.cancelled` - Booking cancelled
- `payment.completed` - Payment successful
- `payment.failed` - Payment failed

Webhook Payload Example

```
{
  "event": "booking.confirmed",
  "timestamp": "2024-06-04T10:35:00Z",
  "data": {
    "booking_id": "BK001234",
    "status": "confirmed",
    "customer_email": "john.doe@example.com",
    "total_amount": "€298.00"
  }
}
```

SDKs and Libraries

Official SDKs are available for popular programming languages:

- **JavaScript/Node.js:** `npm install maritime-reservations-sdk`
- **Python:** `pip install maritime-reservations-sdk`
- **PHP:** `composer require maritime-reservations/sdk`

JavaScript SDK Example

```
import { MaritimeReservations } from 'maritime-reservations-sdk';

const client = new MaritimeReservations({
  apiKey: 'your_api_key',
  environment: 'production' // or 'sandbox'
});

// Search ferries
const ferries = await client ferries.search({
  departurePort: 'Tunis',
  arrivalPort: 'Marseille',
  departureDate: '2024-07-15',
  passengers: 2
});

// Create booking
const booking = await client bookings.create({
  ferryId: ferries[0].id,
  passengers: [
    {
      firstName: 'John',
      lastName: 'Doe',
      dateOfBirth: '1990-01-01'
    }
  ]
});
```

Testing

Sandbox Environment

Use the sandbox environment for testing:

Base URL: `https://api-sandbox.maritime-reservations.com/api/v1`

Test Data

The sandbox includes test data for various scenarios:

- **Test Ferry Routes:** Tunis ↔ Marseille, Genoa ↔ Tunis
- **Test Payment Cards:** Use Stripe test cards
- **Test Users:** Pre-created test accounts available

Test Cards

Card Number	Description
4242424242424242	Visa - Success
4000000000000002	Visa - Declined
40000000000009995	Visa - Insufficient funds

Support

For API support and questions:

- **Documentation:** <https://docs.maritime-reservations.com>
- **Support Email:** api-support@maritime-reservations.com
- **Status Page:** <https://status.maritime-reservations.com>
- **GitHub Issues:** <https://github.com/maritime-reservations/api/issues>