

 Table of Contents

1. Overview & Getting Started
2. Authentication & Security
3. API Endpoints Reference
4. Rate Limiting & Quotas
5. Response Formats
6. HTTP Status Codes
7. Usage Examples & Best Practices
8. Common Errors & Solutions

1 Overview & Getting Started

What is the Mock Train Reservation API?

The Mock Train Reservation API is a comprehensive REST API service that provides realistic train reservation simulation capabilities. It enables developers to build, test, and deploy train booking applications with production-like data and workflows.

Key Features

- **Complete Booking Workflow:** From train search to ticket confirmation and cancellation
- **Real-time Data:** Live train status, station information, and seat availability
- **Secure & Reliable:** Enterprise-grade security with rate limiting and authentication
- **Developer Friendly:** RESTful endpoints with comprehensive documentation
- **Scalable Infrastructure:** Built on PostgreSQL and Redis for high performance

Base URL

http://localhost:8888/mockapis/serverpeuser/api/mocktrain/reserved

Authentication & Security

API Key Setup

All API endpoints require two security headers for authentication:

Header	Type	Description	Example
x-api-key	string	Your unique API key provided upon registration	sk_live_4eC39HqLyjWDarhtT657j8
x-secret-key	string	Your secret key (keep this confidential)	sec_1234567890abcdef
Content-Type	string	Request content type (for POST/PUT requests)	application/json

Security Best Practices

Important Security Notes:

- Never share your API keys in public repositories or client-side code
- Rotate your API keys regularly
- Use HTTPS in production (TLS 1.2 or higher required)
- Monitor your API usage for suspicious activity
- Implement proper error handling without exposing sensitive data

Authentication Example

```
curl -X GET http://localhost:8888/mockapis/serverpeuser/api/mocktrain/reserved/stations \ -H "x-api-key: your-api-key-here" \ -H "x-secret-key: your-secret-key-here" \ -H "Content-Type: application/json"
```

API Endpoints Reference

Reference Data Endpoints (Unchargeable)

These endpoints provide reference data without consuming API credits.

GET /stations Free

Description: Retrieve the complete list of all train stations in the network with their codes and locations.

Request

Headers Required:

x-api-key: your-api-key x-secret-key: your-secret-key

Response (200 - Success)

```
{ "success": true, "remaining_calls": 995, "data": [ { "station_code": "NDLS", "station_name": "New Delhi Junction", "state": "Delhi", "city": "Delhi", "zone": "Northern" }, { "station_code": "CST", "station_name": "Chhatrapati Shivaji Terminus", "state": "Maharashtra", "city": "Mumbai", "zone": "Western" }, { "station_code": "MAS", "station_name": "Chennai Central", "state": "Tamil Nadu", "city": "Chennai", "zone": "Southern" } ] }
```

Error Response (401 - Unauthorized)

```
{ "error": "API key missing" }
```

GET /reservation-type Free

Description: Get the list of available reservation types for train bookings.

Response (200 - Success)

```
{ "success": true, "remaining_calls": 994, "data": [ { "type_id": 1, "type_name": "General", "description": "General Reserved (GN)" }, { "type_id": 2, "type_name": "Tatkal", "description": "Tatkal Reservation (TK)" }, { "type_id": 3, "type_name": "Premium", "description": "Premium Tatkal (PT)" }, { "type_id": 4, "type_name": "First AC", "description": "First AC Premium (AC)" } ] }
```

GET /coach-type Free

Description: Get the list of available coach types with their specifications.

Response (200 - Success)

```
{ "success": true, "remaining_calls": 993, "data": [ { "coach_id": 1, "coach_type": "1A", "seats": 18, "ac": true, "cabin": true, "description": "First AC - Premium Single Cabin" }, { "coach_id": 2, "coach_type": "2A", "seats": 48, "ac": true, "cabin": true, "description": "Second AC - Two Berth Cabins" }, { "coach_id": 3, "coach_type": "3A", "seats": 72, "ac": true, "cabin": false, "description": "Third AC" }, { "coach_id": 4, "coach_type": "SL", "seats": 72, "ac": false, "cabin": false, "description": "Sleeper - Open Compartments" }, { "coach_id": 5, "coach_type": "CC", "seats": 100, "ac": false, "cabin": false, "description": "Chair Car" }, { "coach_id": 6, "coach_type": "2S", "seats": 100, "ac": false, "cabin": false, "description": "Second Sitting" } ] }
```

Train Search Endpoints (Chargeable)

POST /train-schedule Chargeable

Description: Retrieve the complete train schedule database. Returns all trains with their routes and timing information.

Request Body

This endpoint does not require any request body parameters.

Response (200 - Success)

```
{ "success": true, "remaining_calls": 992, "data": [ { "train_number": "12001", "train_name": "New Delhi - Chennai Rajdhani Express", "source": "NDLS", "destination": "MAS", "departure_time": "22:00", "arrival_time": "08:15", "duration_hours": 30.25, "frequency": "Daily", "total_halts": 6, "distance_km": 2144 }, { "train_number": "12002", "train_name": "Chennai - New Delhi Rajdhani Express", "source": "MAS", "destination": "NDLS", "departure_time": "18:00", "arrival_time": "06:15", "duration_hours": 30.25, "frequency": "Daily", "total_halts": 6, "distance_km": 2144 } ] }
```

Credits Cost

1 API call = 1 credit

POST /search-trains Chargeable

Description: Search for trains between two stations on a specific date. Returns available trains with real-time seat availability.

Request Body

```
{ "source_code": "NDLS", "destination_code": "MAS", "doj": "2025-12-25" }
```

Parameter Type Required Description

source_code	string	Yes	Station code of origin (e.g., "NDLS", "CST")
destination_code	string	Yes	Station code of destination (e.g., "MAS", "HWH")
doj	string	Yes	Date of journey (format: YYYY-MM-DD)

Response (200 - Success)

```
{ "success": true, "remaining_calls": 991, "data": [ { "train_number": "12001", "train_name": "Rajdhani Express", "source": "NDLS", "destination": "MAS", "departure": "22:00", "arrival": "08:15", "total_duration": "30 hrs 15 mins", "availability": { "1A": 5, "2A": 12, "3A": 28, "SL": 42, "CC": 0 }, "fare_base": 2500, "classes_available": ["1A", "2A", "3A", "SL"] }, { "train_number": "12003", "train_name": "Express", "source": "NDLS", "destination": "MAS", "departure": "06:00", "arrival": "18:30", "total_duration": "36 hrs 30 mins", "availability": { "SL": 25, "2S": 15 }, "fare_base": 1200, "classes_available": ["SL", "2S"] } ] }
```

Error Response (404 - Invalid Input)

```
{ "success": false, "data": { "statuscode": 404, "successstatus": false, "message": "destination_code not found!" } }
```

Credits Cost

1 API call = 1 credit

Booking Endpoints (Chargeable)

POST /proceed-booking Chargeable

Description: Initiate a booking request. This endpoint reserves seats and generates a booking ID. Payment processing happens after this step.

Request Body

```
{ "train_number": "12001", "doj": "2025-12-25", "coach_type": "2A", "source_code": "NDLS", "destination_code": "MAS", "mobile_number": "9876543210", "passenger_name": "John Doe", "age": 30, "gender": "M", "berth_preference": "U" }
```

Parameter	Type	Required	Description
train_number	string	Yes	Train number (e.g., "12001")
doj	string	Yes	Date of journey (YYYY-MM-DD)
coach_type	string	Yes	Coach class (1A, 2A, 3A, SL, CC, 2S)
source_code	string	Yes	Source station code
destination_code	string	Yes	Destination station code
mobile_number	string	Yes	Passenger mobile number (10 digits)
passenger_name	string	Yes	Full name of passenger
age	integer	Yes	Age of passenger
gender	string	Yes	M (Male) or F (Female)
berth_preference	string	No	U (Upper), M (Middle), L (Lower), SU (Side Upper), SL (Side Lower)

Response (200 - Success)

```
{ "success": true, "remaining_calls": 990, "data": { "booking_id": "BK202512250001", "status": "pending_payment", "allocated_coach": "A1", "allocated_seat": "21U", "seat_type": "Upper Berth", "journey_details": { "train_number": "12001", "train_name": "Rajdhani Express", "departure": "2025-12-25 22:00", "arrival": "2025-12-26 08:15", "source": "New Delhi", "destination": "Chennai" }, "fare_details": { "base_fare": 2500, "reservation_charge": 50, "gst": 405, "total_amount": 2955 }, "validity": { "booking_valid_until": "2025-12-25 18:00", "confirmation_valid_until": "2025-12-25 20:00" } } }
```

✗ Error Response (404 - Train Not Found)

```
{ "success": false, "data": { "statuscode": 404, "message": "Invalid train number or date" } }
```

Credits Cost

1 API call = 1 credit

POST /confirm-ticket Chargeable

Description: Confirm a booking after payment. Generates PNR and ticket number. This is the final step in the booking process.

Caution: Please call end-point /proceed-booking before calling this end-point. Otherwise it throws the error message ‘Booking details not found!...’.

Note: Please use a proper mobile number while calling /proceed-booking endpoint as mock-system sends the mock ticket-confirmation SMS similar to existing railway system.

Request Body

```
{ "booking_id": "BK202512250001", "can_send_mock_ticket_sms": true }
```

Response (200 - Success)

```
{ "success": true, "remaining_calls": 989, "data": { "status": "confirmed", "pn": "1234567890", "ticket_number": "6201234567", "booking_reference": "BK202512250001", "passenger_details": { "name": "John Doe", "age": 30, "gender": "Male", "mobile": "9876543210" }, "journey_details": { "train_number": "12001", "train_name": "Rajdhani Express", "date_of_journey": "2025-12-25", "departure_time": "22:00", "arrival_time": "08:15", "source": "New Delhi", "destination": "Chennai" } }
```

```
{ "destination": "Chennai", "coach": "A1", "seat": "21U" }, "fare_breakdown": { "base_fare": 2500, "taxes": 405, "total": 2905 }, "confirmation_time": "2025-12-25 17:45:30", "sms_status": "sent" } }
```

Credits Cost

1 API call = 1 credit

POST /pnr-status Chargeable

Description: Check the current status of a PNR. Returns passenger details, seat allocation, and booking status.

Request Body

```
{ "pnr": "1234567890" }
```

Response (200 - Success)

```
{ "success": true, "remaining_calls": 988, "data": { "pnr": "1234567890", "booking_status": "confirmed", "journey_date": "2025-12-25", "train_details": { "train_number": "12001", "train_name": "Rajdhani Express", "source": "New Delhi", "destination": "Chennai" }, "passenger_list": [ { "passenger_id": 1, "name": "John Doe", "age": 30, "gender": "M", "mobile": "9876543210", "coach": "A1", "seat": "21", "berth": "U", "status": "CNF" } ], "fare_details": { "ticket_fare": 2500, "gst": 405, "total_paid": 2905 }, "last_updated": "2025-12-25 17:45:30" } }
```

Error Response (404 - PNR Not Found)

```
{ "success": false, "data": { "statuscode": 404, "message": "PNR not found in system" } }
```

Credits Cost

1 API call = 1 credit

POST /cancel-ticket Chargeable

Description: Cancel a confirmed ticket. Returns refund amount after deducting cancellation charges based on timing.

Request Body

```
{ "pnr": "1234567890", "reason": "Personal reasons" }
```

Response (200 - Success)

```
{ "success": true, "remaining_calls": 987, "data": { "status": "cancelled", "pnr": "1234567890", "cancellation_reference": "CXL202512250001", "cancellation_time": "2025-12-25 16:30:00", "original_fare": 2905, "refund_calculation": { "ticket_fare": 2500, "cancellation_charge": 250, "gst_refund": 0, "total_refund": 2250 }, "refund_status": "processed", "refund_amount": 2250, "journey_details": { "train_number": "12001", "date_of_journey": "2025-12-25" } } }
```

Credits Cost

1 API call = 1 credit

Status & Information Endpoints (Chargeable)

POST /train-live-running-status Chargeable

Description: Get real-time running status of a train. Returns current location, delay status, and next station information.

Request Body

```
{ "train_number": "12001" }
```

📍 Response (200 - Success)

```
{ "success": true, "remaining_calls": 986, "data": { "train_number": "12001", "train_name": "Rajdhani Express", "current_status": "Running on Time", "current_location": "Bhopal Junction", "current_station_code": "BPL", "current_station_name": "Bhopal Junction", "last_halt_time": "2025-12-26 02:15", "next_station": "Itarsi Junction", "next_station_code": "ITJ", "eta_next_station": "2025-12-26 03:45", "current_delay": 0, "delay_unit": "minutes", "current_speed": 75, "speed_unit": "km/h", "latitude": 23.1815, "longitude": 77.4104, "status_as_of": "2025-12-26 02:30:45" } }
```

Credits Cost

1 API call = 1 credit

POST /booking-history Chargeable

Description: Retrieve complete booking history for a mobile number. Shows all past and current bookings.

📍 Request Body

```
{ "mobile_number": "9876543210" }
```

📍 Response (200 - Success)

```
{ "success": true, "remaining_calls": 985, "data": [ { "booking_id": "BK202512250001", "pnr": "1234567890", "status": "confirmed", "train_number": "12001", "train_name": "Rajdhani Express", "date_of_journey": "2025-12-25", "source": "New Delhi", "destination": "Chennai", "passenger_count": 1, "total_fare": 2905, "booking_date": "2025-12-20 10:30:00" }, { "booking_id": "BK202512150002", "pnr": "1234567891", "status": "cancelled", "train_number": "12003", "train_name": "Express", "date_of_journey": "2025-12-15", "source": "Delhi", "destination": "Mumbai", "passenger_count": 2, "total_fare": 4500, "refund_amount": 4050, "cancellation_date": "2025-12-15 09:00:00", "booking_date": "2025-12-10 14:20:00" } ] }
```

Credits Cost

1 API call = 1 credit

POST /live-station Chargeable

Description: Get list of trains arriving and departing from a station in the next N hours.

📍 Request Body

```
{ "station_code": "NDLS", "next_hours": 24 }
```

Parameter Type Required Description

station_code	string	Yes	Station code (e.g., "NDLS", "CST")
next_hours	integer	Yes	Number of hours to look ahead (1-72)

📍 Response (200 - Success)

```
{ "success": true, "remaining_calls": 984, "data": { "station_code": "NDLS", "station_name": "New Delhi Junction", "query_time": "2025-12-25 20:00:00", "query_duration_hours": 24, "arriving_trains": [ { "train_number": "12002", "train_name": "Chennai Rajdhani", "source": "Chennai Central", "scheduled_arrival": "2025-12-26 06:15", "platform": 1, "status": "On Time" }, { "train_number": "12004", "train_name": "Mumbai Express", "source": "Mumbai Central", "scheduled_arrival": "2025-12-26 18:30", "platform": 3, "status": "Delayed by 45 mins" } ], "departing_trains": [ { "train_number": "12001", "train_name": "Chennai Rajdhani", "destination": "Chennai Central", "scheduled_departure": "2025-12-25 22:00", "platform": 5, "status": "On Time" }, { "train_number": "12005", "train_name": "Mumbai" }
```

Rajdhani", "destination": "Mumbai Central", "scheduled_departure": "2025-12-26 14:00", "platform": 2, "status": "On Time" }] } }

Credits Cost

1 API call = 1 credit

⚡ Rate Limiting & Quotas

Rate Limiting Policy

Metric	Limit	Response	Notes
Requests/Second	3 req/sec	429 Too Many Requests	Per API key
Scraper Detection	50 req/10 sec	429 Too Many Requests	Blocks for 1 hour
Daily Limit	Varies by plan	429 Quota Exceeded	Resets daily at 00:00 UTC

⚠ Rate Limit Headers:

Monitor these response headers to track your usage:

- X-RateLimit-Limit - Total requests allowed
- X-RateLimit-Remaining - Requests remaining in window
- X-RateLimit-Reset - Unix timestamp when limit resets

🎯 HTTP Status Codes Reference

2xx Success Responses

Code Status	Description	Example Scenario
200 OK	Request successful and data returned	Train search completed successfully
201 Created	Resource created successfully	Booking created
204 No Content	Request successful but no content	Successful deletion

4xx Client Errors

Code Status	Description	Resolution
400 Bad Request	Invalid request parameters or malformed JSON	Check request body and parameters
401 Unauthorized	Missing or invalid API credentials	Verify API keys and secret keys
403 Forbidden	API key valid but not authorized for this endpoint	Check API key permissions
404 Not Found	Requested resource not found	Verify train number, PNR, or station code
422 Unprocessable Entity	Request validation failed	Check all required fields are provided
429 Too Many Requests	Rate limit exceeded	Implement backoff strategy and retry

5xx Server Errors

Code Status	Description	Action
500 Internal Server Error	Server encountered an unexpected error	Retry after a few seconds; contact support if persists
502 Bad Gateway	Server returned invalid response	Retry after 30 seconds
503 Service Unavailable	Server temporarily unavailable or under maintenance	Retry later or check status page

💡 Usage Examples & Best Practices

Example 1: Complete Booking Workflow

Here's how to implement a complete train booking flow:

```
// Step 1: Get list of stations GET /mockapis/serverpeuser/api/mocktrain/reserved/stations Headers: x-api-key: your-api-key x-secret-key: your-secret-key // Step 2: Search for trains POST /mockapis/serverpeuser/api/mocktrain/reserved/search-trains Body: { "source_code": "NDLS", "destination_code": "MAS", "doj": "2025-12-25" } // Step 3: Proceed with booking POST /mockapis/serverpeuser/api/mocktrain/reserved/proceed-booking Body: { "train_number": "12001", "doj": "2025-12-25", "coach_type": "2A", "source_code": "NDLS", "destination_code": "MAS", "mobile_number": "9876543210", "passenger_name": "John Doe", "age": 30, "gender": "M" } // Step 4: Confirm booking (after payment) POST /mockapis/serverpeuser/api/mocktrain/reserved/confirm-ticket Body: { "booking_id": "BK202512250001", "can_send_mock_ticket_sms": true }
```

Example 2: Error Handling with Retry Logic

```
async function searchTrainsWithRetry(sourceCode, destCode, doj) { const maxRetries = 3; let retries = 0; while (retries < maxRetries) { try { const response = await fetch('http://localhost:8888/mockapis/serverpeuser/api/mocktrain/reserved/search-trains', { method: 'POST', headers: { 'x-api-key': YOUR_API_KEY, 'x-secret-key': YOUR_SECRET_KEY, 'Content-Type': 'application/json' }, body: JSON.stringify({ source_code: sourceCode, destination_code: destCode, doj: doj }) } ); if (response.status === 429) { // Rate limited, wait and retry const resetTime = response.headers['X-RateLimit-Reset']; const waitTime = Math.max(1000, new Date(resetTime) - Date.now()); await sleep(waitTime); retries++; continue; } if (!response.ok) { throw new Error(`HTTP ${response.status}`); } return await response.json(); } catch (error) { retries++; if (retries >= maxRetries) throw error; await sleep(2000 * retries); // exponential backoff } } } function sleep(ms) { return new Promise(resolve => setTimeout(resolve, ms)); }
```

Best Practices

✓ Recommended Practices:

- **Use Connection Pooling:** Reuse HTTP connections for better performance
- **Implement Caching:** Cache station and coach type data locally (valid for 24 hours)
- **Error Handling:** Always implement proper error handling for all edge cases
- **Rate Limiting:** Respect rate limits and implement exponential backoff for retries
- **Timeouts:** Set appropriate request timeouts (recommend 30 seconds)
- **Logging:** Log all API interactions for debugging and monitoring
- **Monitoring:** Track API usage and performance metrics
- **Data Validation:** Validate all user inputs before sending to API

✗ Common Errors & Solutions

Error: "API key missing"

Response: { "error": "API key missing" } Status: 401

Solution: Ensure the x-api-key header is included in your request.

Error: "Invalid api key or secret key"

Response: { "status": "Failed", "message": "Invalid api key or secret key" } Status: 401

Solution: Verify that both x-api-key and x-secret-key headers are correct. Check for typos or extra spaces.

Error: "destination_code not found!"

Response: { "statuscode": 404, "message": "destination_code not found!" } Status: 404

Solution: The destination station code is invalid. Call the /stations endpoint to get valid station codes.

Error: "Invalid train number"

Response: { "statuscode": 404, "message": "Invalid train number" } Status: 404

Solution: The train number is not valid for the given date. Search trains first to get valid train numbers.

Error: "PNR not found"

Response: { "statuscode": 404, "message": "PNR not found in system" } Status: 404

Solution: The PNR doesn't exist or has expired. Verify the PNR number or check booking history.

Error: Too Many Requests (429)

Response: { "error": "Rate limit exceeded" } Status: 429 Headers: X-RateLimit-Reset: 1703520000

Solution: You've exceeded the rate limit. Implement exponential backoff and retry after the time specified in X-RateLimit-Reset header.

Appendix

Valid Station Codes

Code	Station Name	State	Zone
NDLS	New Delhi Junction	Delhi	Northern
CST	Chhatrapati Shivaji Terminus	Maharashtra	Western
MAS	Chennai Central	Tamil Nadu	Southern
HWH	Howrah Junction	West Bengal	Eastern
BPL	Bhopal Junction	Madhya Pradesh Central	

Valid Coach Types

Code	Type	AC Capacity	Description
1A	First AC	Yes 18	Premium single cabin compartments
2A	Second AC	Yes 48	Two-berth cabin compartments
3A	Third AC	Yes 72	Three-berth open compartments
SL	Sleeper	No 72	Open compartments with sleeper berths
CC	Chair Car	No 100	Non-AC seating with reclining chairs
2S	Second Sitting	No 100	Budget seating option

Berth Preference Codes

Code	Meaning	Available In
U	Upper Berth	All AC and SL
M	Middle Berth	3A, SL
L	Lower Berth	All AC and SL
SU	Side Upper Berth	3A, SL
SL	Side Lower Berth	3A, SL

Mock Train Reservation API v1.0.0

Professional API Documentation & Reference Guide

Generated: December 17, 2025 at 02:24:18 PM

For support and technical assistance, please contact: support@serverpe.com

© 2025 ServerPE. All rights reserved. | This documentation is confidential and intended for authorized users only.