

ServerPe App Solutions

(www.serverpe.in).

Overview

This is the API documentation for the **Mock Train Seat Reservation System**. All endpoints require proper authentication headers to access the APIs.

Security Notice: API keys have been redacted for security purposes. To access these APIs, you must obtain an **API key through subscription** from ServerPe. Once subscribed, include the issued API key as a **key-value pair** in the request headers (e.g., x-api-key) when invoking the endpoints below using tools such as Postman or from your application.

Authentication

All API requests require the following headers:

Header	Description
x-api-key	Your API key

Brief about Mock train reservation apis

The **Mock Train Reservation API** is a fully simulated ticketing system designed to replicate real-world railway reservation workflows. It enables UI developers to build, test, and validate complete train booking journeys without integrating with live railway systems.

This API supports the entire reservation lifecycle, including train search, seat availability simulation, passenger details capture, booking confirmation, PNR generation, ticket cancellation, and live train status tracking. Dynamic state transitions such as **Waitlist** → **RAC** → **Confirmed** are handled internally to closely mimic real-world booking behavior.

To further enhance realism, a **backend booking simulator** runs continuously as a scheduled background process (cron job). This simulator automatically books seats across all trains **sequentially every second**, dynamically updating availability, waitlist, RAC, and confirmation states. This mechanism ensures that seat availability and booking responses change over time, closely replicating real-world demand and booking patterns.

Each booking flow follows a structured, step-by-step process where outputs from one endpoint (such as **booking_id** or **PNR**) are required as inputs for subsequent endpoints. This enforces a

realistic end-to-end booking sequence for frontend applications.

Important: This is a mock API strictly intended for **learning, UI development, demos, and testing**. No real railway data, live systems, or actual ticket bookings are involved.

Endpoints

Point-1

GET **Get all available reservation types**

Description:

This endpoint returns a list of all available reservation types, closely simulating real-world railway reservation categories. Use the response JSON object to accurately map and validate your UI implementation.

`https://api.serverpe.in/api/mockapis/serverpeuser/api/mocktrain/reserved/reservation-type`

Headers

Header	Value
x-api-key	YOUR_API_KEY_HERE

Point-2

GET **Get all available stations with details**

Description:

This endpoint returns a list of all available station details, closely simulating real-world railway station data. Use the response JSON object to accurately map and validate your UI implementation.

`https://api.serverpe.in/api/mockapis/serverpeuser/api/mocktrain/reserved/stations`

Headers

Header	Value
x-api-key	YOUR_API_KEY_HERE

Point-3

GET Get all available coach types

Description:

This endpoint returns a list of all available coach types, closely simulating real-world railway coach classifications. Use the response JSON object to accurately map and validate your UI implementation.

`https://api.serverpe.in/api/mockapis/serverpeuser/api/mocktrain/reserved/coach-type`

Headers

Header	Value
x-api-key	YOUR_API_KEY_HERE

Point-4

POST Search trains

Description:

This endpoint searches and returns a list of available trains based on the provided **source station code**, **destination station code**, and **date of journey** (yyyy-mm-dd format) specified in the request body below. Use the response JSON object to appropriately map and validate your UI implementation.

`https://api.serverpe.in/api/mockapis/serverpeuser/api/mocktrain/reserved/search-trains`

Headers

Header	Value
x-api-key	YOUR_API_KEY_HERE

Request Body

```
{ "source_code": "ypr", "destination_code": "hvr", "doj": "2025-12-23",  
  "coach_type": null, "reservation_type": null }
```

Point-5

POST Proceed booking (this takes train details, passenger details)

This endpoint accepts the selected train details and passenger information in the request body to **summarize ticket details** before seat confirmation or invoking **Point-6**. Please use the example request body provided below to receive a valid response. Use the response JSON object to accurately map and validate your UI implementation.

<https://api.serverpe.in/api/mockapis/serverpeuser/api/mocktrain/reserved/proceed-booking>

Headers

Header	Value
x-api-key	YOUR_API_KEY_HERE

Request Body

```
{ "train_number": "17391", "doj": "2025-12-23", "coach_type": "sl",  
  "source_code": "ypr", "destination_code": "hvr", "mobile_number":  
  "9*****", "reservation_type": "gen", "passenger_details": [ {  
    "passenger_name": "Amruta", "passenger_gender": "F", "passenger_age": 36,  
    "passenger_ischild": false, "passenger_issenior": false } ] }
```

Point-6

POST Mock Confirm seat booking

This endpoint depends on **Point-5**. You must obtain the **booking_id** from the response of that endpoint and include it in the **request body** shown below. The additional parameter `can_send_mock_ticket_sms: true` is used to simulate sending a mock confirmed ticket

notification to the mobile number provided in **Point-5**. Use the response JSON object to appropriately map and validate your UI implementation.

```
https://api.serverpe.in/api/mockapis/serverpeuser/api/mocktrain/reserved/confirm-ticket
```

Headers

Header	Value
x-api-key	YOUR_API_KEY_HERE

Request Body

```
{ "booking_id":58, "can_send_mock_ticket_sms":true }
```

Point-7

POST Mock PNR status

This endpoint returns **booked ticket details** along with **passenger information** based on the provided **PNR number**. The PNR is generated during the booking process in **Point-6**. The response structure closely simulates **real-world railway ticket data**, making it ideal for UI development and testing.

```
https://api.serverpe.in/api/mockapis/serverpeuser/api/mocktrain/reserved/pnr-status
```

Headers

Header	Value
x-api-key	YOUR_API_KEY_HERE

Request Body

```
{ "pnr":"PNR001514" }
```

Endpoint-8

POST Train schedules

The Train Schedule API provides the complete route timetable for a given train number. It returns a station-wise breakdown including arrival time, departure time, halt duration, and day sequence. This mock API is designed to help UI developers simulate real-world railway schedule screens such as journey timelines, station lists, and progress indicators without relying on live railway systems. Key use cases: Display full train route with timings Build journey timeline UI Show intermediate stops and halts Test schedule-based calculations (arrival/departure) ⚠️ This is a simulated API for learning, testing, and UI development only. No real railway data or live connections are used.

`https://api.serverpe.in/api/mockapis/serverpeuser/api/mocktrain/reserved/train-schedule`

Headers

Header	Value
x-api-key	YOUR_API_KEY_HERE

Request Body

```
{ "train_number": "12614" }
```

Endpoint-9

POST Mock Ticket cancellation

This endpoint simulates ticket cancellation by accepting a valid **PNR number** along with an array of **passenger IDs** in the request body. It supports both partial and full ticket cancellation scenarios and updates the booking status accordingly. Use the response JSON object to reflect cancellation status changes in your UI.

`https://api.serverpe.in/api/mockapis/serverpeuser/api/mocktrain/reserved/cancel-ticket`

Headers

Header	Value
x-api-key	YOUR_API_KEY_HERE

Request Body

```
{ "pnr": "PNR001521", "passengerids": [ 1071674 ] }
```

Endpoint-10

POST Trains between two stations

This endpoint retrieves a list of trains running between the specified **source station** and **destination station**. It simulates real-world railway data and is useful for building train discovery, route selection, and availability listing UIs. Use the response JSON object to accurately map and validate your UI implementation.

```
https://api.serverpe.in/api/mockapis/serverpeuser/api/mocktrain/reserved/trains-between-two-stations
```

Headers

Header	Value
x-api-key	YOUR_API_KEY_HERE

Request Body

```
{ "source_code": "YPR", "destination_code": "UBL" }
```

POST Live station

This endpoint provides simulated live station data by returning a list of trains scheduled to arrive at or depart from the specified **station code**. The `next_hours` parameter defines the time window (e.g., 2, 4, or 8), and the response includes trains occurring from the **current time** up to the specified number of upcoming hours. This API helps UI developers build live station boards and arrival/departure tracking screens. Use the response JSON object to accurately map and validate your UI implementation.

```
https://api.serverpe.in/api/mockapis/serverpeuser/api/mocktrain/reserved/live-station
```

Headers

Header	Value
x-api-key	YOUR_API_KEY_HERE

Request Body

```
{ "station_code":"MYS", "next_hours":2 }
```

Endpoint-12

POST Mock Live train running status

This endpoint provides a simulated live running status of a train based on the provided **train number**. It returns real-time-like information such as current location, last passed station, upcoming stations, and delay or on-time status. This mock API helps UI developers build live train tracking and status screens without relying on real railway systems. Use the response JSON object to accurately map and validate your UI implementation.

```
https://api.serverpe.in/api/mockapis/serverpeuser/api/mocktrain/reserved/train-live-running-status
```

Headers

Header	Value
x-api-key	YOUR_API_KEY_HERE

Request Body

```
{ "train_number":"12080" }
```

Feedback & Improvements

I continuously strive to improve the quality and usability of our mock APIs. Your feedback, suggestions, and improvement ideas are highly appreciated.

For technical queries or support, please write to support@serverpe.in. For feedback, suggestions, or feature requests, reach out to feedback@serverpe.in.

Generated on 27/12/2025, 10:28:12 am
API Documentation - ServerPe App Solutions
