

Train Service API

An api for managing train services, stations, user wallets, and ticketing.

Auth

API for user register, login and authentication

POST Register

localhost:3000/api/auth/register

This endpoint allows you to register a new user.

- **Endpoint:** `/api/auth/register`
 - **Method:** POST
 - **Parameters:** None
 - **Authorization:** Anyone
 - **Request Body:**
 - `name: (string)` The name of the user.
 - `email: (string)` The email address of the user.
 - `password: (string)` The password for the user account.
 - **Expected Response:** Upon a successful login, the API returns a JSON object with the following fields:
 - `_id: (string)` The unique identifier of the registered user.
 - `name: (string)` The name of the registered user.
 - `email: (string)` The email address of the registered user.
 - `role: (string)` The role of the registered user.
-

POST Login

localhost:3000/api/auth/login

This API endpoint allows users to log in by providing their email and password.

password.

- **Endpoint:** `/api/auth/login`
 - **Method:** POST
 - **Parameters:** None
 - **Authorization:** Registered User
 - **Request Body:**
 - `name: (string)` The name of the user.
 - `email: (string)` The email address of the user.
 - `password: (string)` The password for the user account.
 - **Expected Response:** Upon a successful login, the API returns a JSON object with the following fields:
 - `_id: (string)` The unique identifier of the registered user.
 - `name: (string)` The name of the registered user.
 - `email: (string)` The email address of the registered user.
 - `role: (string)` The role of the registered user.
 - `token:` A token for authentication.
-

GET Profile

`localhost:3000/api/auth/Profile`

This endpoint retrieves the profile information for the authenticated user.

- **Endpoint:** `/api/auth/profile`
- **Method:** GET
- **Parameters:** None
- **Authorization:** Registered User
- **Expected Result:**
 - `_id (string)`: The unique identifier of the user.
 - `name (string)`: The name of the user.
 - `email (string)`: The email address of the user.
 - `wallet (object)`: Contains information about the user's wallet:
 - `balance (number)`: The current balance in the user's wallet.
 - `transactions (array)`: An array of objects representing the transactions made in the wallet, each containing:
 - `method (string)`: The method of the transaction.
 - `amount (number)`: The amount of the transaction.
 - `description (string)`: A description of the transaction.
 - `_id (string)`: The unique identifier of the transaction.
 - `date (string)`: The date of the transaction.
 - `ticket (array)`: An array of objects representing the tickets associated with the user, each containing:

associated with the user, each containing:

- `user (string)`: The unique identifier of the user who purchased the ticket.
- `train (string)`: The unique identifier of the train associated with the ticket.
- `startStation (string)`: The unique identifier of the starting station.
- `endStation (string)`: The unique identifier of the ending station.
- `departureTime (string)`: The scheduled departure time of the train.
- `_id (string)`: The unique identifier of the ticket.
- `createdAt (string)`: The timestamp when the ticket was created.
- `__v (number)`: The version key.

AUTHORIZATION Bearer Token

Token <token>

POST Create Admin

localhost:3000/api/auth/create-admin

This endpoint allows you to register a new user.

- **Endpoint:** `/api/auth/create-admin`
- **Method:** POST
- **Parameters:** None
- **Authorization:** Admin
- **Request Body:**
 - `name: (string)` The name of the user.
 - `email: (string)` The email address of the user.
 - `password: (string)` The password for the user account.
- **Expected Response:** Upon a successful login, the API returns a JSON object with the following fields:
 - `_id: (string)` The unique identifier of the registered user.
 - `name: (string)` The name of the registered user.
 - `email: (string)` The email address of the registered user.
 - `role: (string)` The role of the registered user.

AUTHORIZATION Bearer Token

Token <token>

Body raw (json)

```
json

{
  "name": "Mir Tanim Ahmed",
  "email": "mtanim.ahmed18@gmail.com",
  "password": "12345678"
}
```

Station

CRUD operations for Stations.

GET allStation

localhost:3000/api/station

This endpoint is designed to fetch the list of all stations.

- **Endpoint:** `/api/station`
 - **Method:** GET
 - **Parameters:** None
 - **Authorization:** Anyone
 - **Expected Response:** A JSON object containing an array of stations, each with relevant details like `_id`, `name` and the location of the station.
-

GET stationById

localhost:3000/api/station/:id

This endpoint retrieves the details of a specific station identified by the provided ID.

- **Endpoint:** `/api/station/:id`
 - **Method:** GET
 - **Parameters:**
-

- `id (string, required)`: The unique identifier of the station to retrieve details for.
- **Authorization:** Anyone
- **Expected Response:** A JSON object containing the station details like `_id`, name and the location of the station.

PATH VARIABLES

`id`

POST createStation

`localhost:3000/apistation/create`

This endpoint is used to create a new station.

- **Endpoint:** `/api/station/create`
- **Method:** POST
- **Parameters:** None
- **Authorization:** Admin
- **Request Body:**
 - `name (string, required)`: The name of the station.
 - `location (string, required)`: The location of the station.
- **Expected Response:** A JSON object containing the details of the newly created station, including its unique identifier.

AUTHORIZATION Bearer Token

Token	<token>
-------	---------

PUT updateStation

`localhost:3000/api/station/update/:id`

This endpoint is used to update an existing station's details.

- **Endpoint:** `/api/station/update/:id`
- **Method:** PUT
- **Parameters:**
 - `id (string, required)`: The unique identifier of the station to be

updated.

- **Authorization:** Admin
- **Request Body:**
 - **name** (string, optional): The new name of the station.
 - **location** (string, optional): The new location of the station.
- **Expected Response:** A JSON object containing the updated details of the station.
- **Usage:** This endpoint is ideal for modifying the details of an existing station in the system.

AUTHORIZATION Bearer Token

Token <token>

PATH VARIABLES

id

DELETE deleteStation

localhost:3000/api/station/delete/:id

This endpoint is used to delete an existing station.

- **Endpoint:** `/api/station/delete/:id`
- **Method:** DELETE
- **Parameters:**
 - `id (string, required)`: The unique identifier of the station to be deleted
- **Authorization:** Admin
- **Expected Response:** A JSON object confirming the deletion of the station.
- **Usage:** This endpoint is ideal for removing a station from the system.

AUTHORIZATION Bearer Token

Token <token>

PATH VARIABLES

id

Train

CRUD operations for Trains.

GET allTrain

localhost:3000/api/train

This endpoint is designed to fetch the list of all trains.

- **Endpoint:** `/api/train`
 - **Method:** GET
 - **Parameters:** None
 - **Authorization:** Anyone
 - **Expected Response:** The response will include an array of train details, where each object contains the train ID, name, and an array of stops. Each stop includes the station details (ID, name, location), time, and base fee.
-

GET trainById

localhost:3000/api/train/:id

This endpoint retrieves the details of a specific train identified by the provided ID.

- **Endpoint:** `/api/train/:id`
- **Method:** GET
- **Parameters:**
 - `id (string, required)`: The unique identifier of the train to retrieve details for.
- **Authorization:** Anyone
- **Expected Response:** A JSON object containing the train details identified by the ID, which contains the train ID, name, and an array of stops. Each stop includes the station details (ID, name, location), time, and base fee.

PATH VARIABLES

id

POST createTrain

localhost:3000/api/train/create

This endpoint allows the creation of a new train with specified name and stops. This endpoint is used to create a new station.

- **Endpoint:** `/api/train/create`
- **Method:** POST
- **Parameters:** None
- **Authorization:** Admin
- **Request Body:**
 - `name (string)`: The updated name of the train.
 - `stops (array)`: An array of objects containing updated information about the stops along the train route.
 - `station (string)`: The ID of the station.
 - `time (string)`: The updated time of arrival at the station.
 - `baseFee (number)`: The updated base fee for the station.
- **Expected Response:** A JSON object containing the details of the newly created train, including its unique identifier.

AUTHORIZATION Bearer Token

Token	<token>
-------	---------

PUT updateTrain

localhost:3000/api/train/update/:id

This endpoint is used to update the details of a train with a specific ID.

- **Endpoint:** `/api/train/update/:id`
- **Method:** PUT
- **Parameters:**
 - `id (string, required)`: The unique identifier of the train to be updated.
- **Authorization:** Admin
- **Request Body:**

- `name (string)`: The updated name of the train.
- `stops (array)`: An array of objects containing updated information about the stops along the train route.
 - `station (string)`: The ID of the station.
 - `time (string)`: The updated time of arrival at the station.
 - `baseFee (number)`: The updated base fee for the station.
- **Expected Response:** A JSON object containing the updated details of the train.
- **Usage:** This endpoint is ideal for modifying the details of an existing train in the system.

AUTHORIZATION Bearer Token

Token <token>

PATH VARIABLES

id

DELETE deleteTrain

localhost:3000/api/train/delete/:id

This endpoint is used to delete an existing train.

- **Endpoint:** `/api/train/delete/:id`
- **Method:** DELETE
- **Parameters:**
 - `id (string, required)`: The unique identifier of the train to be deleted.
- **Authorization:** Admin
- **Expected Response:** A JSON object confirming the deletion of the train.
- **Usage:** This endpoint is ideal for removing a train from the system.

AUTHORIZATION Bearer Token

Token <token>

PATH VARIABLES

id

Wallet

POST addFund

localhost:3000/api/wallet/add-funds

This endpoint allows you to add funds to the user's wallet.

- **Endpoint:** `/api/wallet/add-funds`
- **Method:** POST
- **Parameters:** None
- **Authorization:** Registered User
- **Request Body:**
 - `method (string, required)`: The method used to add funds (e.g. bkaash).
 - `amount (number, required)`: The amount to be added to the wallet.
 - `description (string, optional)`: Additional description for the transaction.
- **Expected Response :** Upon successful execution, the server will respond with a status code of 200 and a JSON object containing the following fields:
 - `message (string)`: A message indicating the result of the operation.
 - `wallet (object)`:
 - `user (string)`: The user associated with the wallet.
 - `balance (number)`: The updated balance after adding funds.
 - `transactions (array)`: An array of transaction objects containing details of the transaction.
 - `method (string)`: The method used for the transaction.
 - `amount (number)`: The amount of the transaction.
 - `description (string)`: Additional description for the transaction.
 - `_id (string)`: Unique identifier for the transaction.
 - `date (string)`: Date of the transaction.
 - `_id (string)`: Unique identifier for the wallet.
 - `__v (number)`: Version of the wallet object.

AUTHORIZATION Bearer Token

Token

<token>

GET fundDetails

localhost:3000/api/wallet/details

This endpoint retrieves the details of the user's wallet.

- **Endpoint:** `/api/wallet/details`
- **Method:** GET
- **Parameters:** None
- **Authorization:** Registered User
- **Expected Response:** The response will be in JSON format with the following details:
 - `balance (number)`: The current balance in the user's wallet.
 - `transactions (array)`: An array of objects representing the transactions made in the wallet.
 - `method (string)`: The method of the transaction.
 - `amount (number)`: The amount of the transaction.
 - `description (string)`: A description of the transaction.
 - `_id (string)`: The unique identifier of the transaction.
 - `date (string)`: The date of the transaction.

AUTHORIZATION Bearer Token

Token <token>

Ticket

POST purchase

localhost:3000/api/ticket/purchase

This endpoint allows the user to purchase a train ticket.

- **Endpoint:** `/api/ticket/purchase`
- **Method:** POST
- **Parameters:** None
- **Authorization:** Registered User
- **Request Body:**
 - `trainId (string)` – The ID of the train for which the ticket is being purchased

- `purchased`.
 - `startStation (string)` – The ID of the starting station for the journey.
 - `endStation (string)` – The ID of the destination station for the journey.
 - `method (string)` – The payment method to be used for purchasing the ticket.
- **Expected Response:** The response will be in JSON format with the following details:
 - `message (string)`: A confirmation message regarding the ticket purchase.
 - `ticket (object)`: Details about the purchased ticket, including:
 - `user (string)`: The unique identifier of the user who purchased the ticket.
 - `train (string)`: The unique identifier of the train associated with the ticket.
 - `startStation (string)`: The unique identifier of the starting station.
 - `endStation (string)`: The unique identifier of the ending station.
 - `departureTime (string)`: The scheduled departure time of the train.
 - `_id (string)`: The unique identifier of the ticket.
 - `createdAt (string)`: The timestamp when the ticket was created.
 - `__v (number)`: The version key.
 - `balance (number)`: The current balance in the user's wallet.
 - `totalFare (number)`: The total fare for the purchased ticket.

AUTHORIZATION Bearer Token

Token <token>