

FLIGHT API Documentation

Document Revision History

Date	Version	Description
05-01-25	0.1	Document's first version

1. Purpose and Scope of the Document

This document serves as a comprehensive guide to the **Zoo Travel Technology Flight API**, which can be seamlessly integrated into various applications, such as web applications, mobile apps, and other platforms. It is specifically designed to assist developers and other stakeholders in understanding the functionalities and operational framework of the **Zoo Travel Technology Flight API Service**.

The primary objectives of this guide are:

- To provide detailed instructions for integrating the **Zoo Travel Technology Flight API** into your application.
- To explain the purpose and usage of each API method in a clear and concise manner.
- To outline the support resources available to facilitate efficient and effective API utilization.
- To describe how to access the resources provided by **Zoo Travel Technology Limited**.
- To detail the certification process required to achieve "Go Live" status for the API.

It is strongly recommended that you thoroughly review this document before commencing the integration process.

Prerequisites:

- Familiarity with web services and related technologies is assumed for using this guide effectively.

By following the instructions and guidelines outlined in this document, you will gain the necessary insights and tools to successfully integrate and leverage the **Zoo Travel Technology Flight API** for your applications.

2. Confidentiality of the Document

This document is a preliminary release and is subject to revisions prior to its final commercial publication. It is intended solely for informational purposes and is provided by **Zoo Travel Technology Limited**. **Zoo Travel Technology Limited** makes no warranties, express or implied, regarding the content of this document.

The information presented herein reflects the current perspective of **Zoo Travel Technology Limited** on the topics discussed as of the publication date. Any details, including URLs and other references to internet websites, are subject to modification without prior notice.

3. Journey Type

Journey Type Number	Purpose Name
1	One Way
2	Round Trip
3	Multicity

4. API Definition

An **Application Programming Interface (API)** serves as an interface or communication protocol that facilitates interaction between different components of a computer program. Its primary purpose is to streamline the development, implementation, and maintenance of software systems.

APIs can be designed for a variety of purposes, including web-based systems, operating systems, database systems, computer hardware, or software libraries. They define the methods and data formats that developers can use to interact with the system, promoting efficiency, reusability, and scalability in software development.

5. Authentication

The authentication method is responsible for generating and returning a unique Token ID, which serves as a secure identifier for the authenticated session. This Token ID must be included in the header or body of every subsequent request to ensure the server recognizes the authenticated user or system.

If the session expires or the user logs out, the authentication method must be invoked again to obtain a new Token ID. This updated Token ID must then be used in all following requests to maintain proper authentication. Ensuring the Token ID is consistently provided in each request is critical for maintaining a secure and seamless interaction between the client and server.

#	Parameter Name	Data Type	Constraints	Remarks
1	username	string	required	API User ID
2	password	string	required	API Password

API Name	Authentication
Details	Authenticate user for token generation
Method	POST
Access To	API Server
Request Object	<pre>{ "Email": "{{email}}", "Password": "{{password}}" }</pre>
Response Object	<pre>{ "data": { "Token": "your-token-here" } }</pre>
Method Name/URL	{{base_url}}/login
Header Parameter	Accept: application/json Content-Type: application/json
Request Format	"Username": String, "Password":String
Response Payload Format	<pre>{ "Request Time": String, "ResponseTime": String, "RequestURL": String, "Success": bool, "Message": String, "Payload": { "access token" :String, "token_type" :String, "userName": String, ".issued":String, ".expires": String }, "Payload Type": String }</pre>
Sample Request	<pre>{ "Email": "user@example.com", "Password": "yourpassword" }</pre>

6. Login API Documentation

The **Login API** is used to authenticate users by verifying their credentials (email and password). Upon successful login, the API returns a unique **SessionID** to be used for subsequent API requests.

6.1 Request Format:

- **Method:** POST
- **Endpoint:** `{{base_url}}/user/login`

6.2 Request Body:

Key	Type	Description	Example
Email	String	The email address of the user.	"test@example.com"
Password	String	The password associated with the user account.	"Password"

6.3 Example Request:

```
{  
  "Email": "ota@example.com",  
  "Password": "Password"  
}
```

6.4 Response:

6.4.1 Success Response:

Key	Type	Description	Example
Status	String	Indicates if the login was successful.	"Success"
SessionID	String	A unique identifier for the session.	"a1b2c3d4e5f6g7h8i9j0"

6.4.2 Example Success Response:

```
{  
  "Status": "Success",  
  "SessionID": "a1b2c3d4e5f6g7h8i9j0"  
}
```

6.4.3 Error Response:

Key	Type	Description	Example
ErrorCode	String	Code identifying the specific error.	"ERR_INVALID_CREDENTIALS"
Message	String	Error message detailing the issue.	"Invalid email or password."

6.4.4 Example Error Response:

```
{
  "ErrorCode": "ERR_INVALID_CREDENTIALS",
  "Message": "Invalid email or password."
}
```

Notes:

1. Ensure the **Email** and **Password** provided are valid and correspond to an existing user account.
2. Use the returned **SessionID** for all subsequent authenticated requests.
3. Handle error responses gracefully to inform the user about necessary corrective actions.

7. Register API Documentation

The **Register API** is used to create a new user account by providing necessary details such as **name**, **email**, and **password**. This allows users to access the system and perform subsequent operations.

7.1 Request Format:

- **Method:** POST
- **Endpoint:** `{{base_url}}/user/register`

7.2 Request Body:

Key	Type	Description	Example
FirstName	String	The first name of the user.	"Airways"
LastName	String	The last name of the user.	"Office"
Email	String	The email address to register the account.	"ota@example.com"

Password	String	The password for the account.	"Password"
ConfirmPassword	String	Confirmation of the password provided.	"Password"

7.3 Example Request:

```
{
  "FirstName": "Airways",
  "LastName": "Office",
  "Email": "ota@example.com",
  "Password": "Password",
  "ConfirmPassword": "Password"
}
```

7.4 Response:

7.4.1 Success Response:

Key	Type	Description	Example
Status	String	Indicates if the registration succeeded.	"Success"
Message	String	Confirmation message about the registration.	"User registered successfully."

7.4.2 Example Success Response:

```
{
  "Status": "Success",
  "Message": "User registered successfully."
}
```

7.4.3 Error Response:

Key	Type	Description	Example
ErrorCode	String	Code identifying the specific error.	"ERR_EMAIL_ALREADY_EXISTS"
Message	String	Error message detailing the issue.	"Email is already registered."

7.4.4 Example Error Response:

```
{  
  "ErrorCode": "ERR_EMAIL_ALREADY_EXISTS",  
  "Message": "Email is already registered."  
}
```

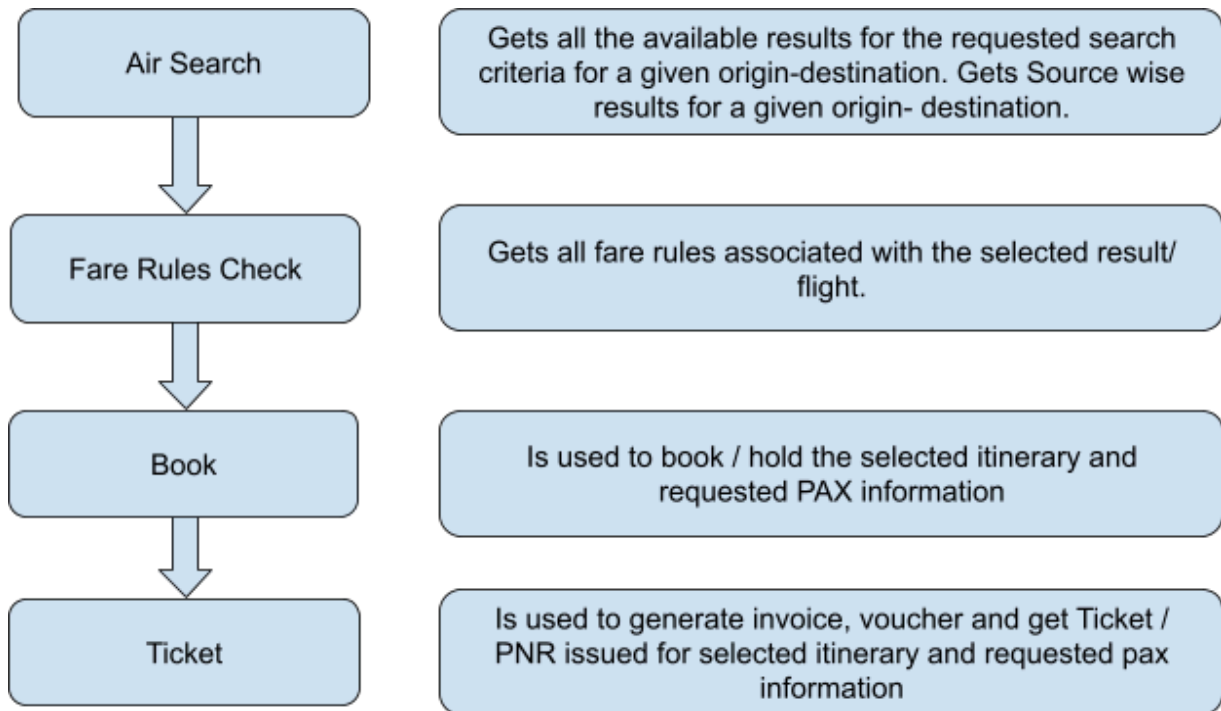
Notes:

1. Ensure that the **Email** is not already registered in the system to avoid errors.
2. The **Password** and **ConfirmPassword** fields must match.
3. Handle error responses to inform users of corrective actions, such as choosing a different email or ensuring passwords match.
4. Strong passwords are recommended to ensure account security.

8. Booking

8.1 Booking Path Way:

The Zoo Travel Technology Ltd Flight API offers the following pathway to create a booking as per the provided criteria.



8.2. Search: Flight Availability and Fare Options

The **Search** method provides information about flight availability along with the cheapest fare options for a given query. This functionality is designed to handle single queries and includes an optional field for the **Return Date**.

8.2.1 Key Features:

- Return Date Field:**
 - This field is optional but should be filled when requesting a round-trip search.
 - If provided, it specifies the date for the return journey, enabling the system to search for round-trip options.
- Search Process:**

- Upon receiving a properly formatted query, the booking engine begins parsing and validating the request.
- The engine then initiates a comprehensive search across all possible sources to find the best flight options matching the criteria.
- If successful, the system returns a **Result Object**, which contains the available flight options.

8.2.2. Flight Types Supported:

The Search method supports the following three types of flights:

1. One-Way Flight:

- A ticket that allows the passenger to travel to a single destination without a return trip.
- Ideal for travelers who do not need to come back to their starting point.

2. Round Trip:

- A ticket for a journey that includes traveling from the starting location to a destination and then returning to the original starting point.
- Requires the **Return Date** to be specified in the query.

3. Multi-City Flight:

- A ticket that allows a traveler to visit multiple destinations in one trip.
- Offers flexibility by adding extra legs with stopovers, enabling visits to more than one location.

This structured approach ensures that users can efficiently search for and book flights that meet their travel needs, whether they are planning a simple one-way trip, a round trip, or a multi-city itinerary.

Key	Type	Description	Example
Adults	Integer	Number of adult passengers	1
Children	Integer	Number of child passengers (2-11 years)	0
Kids	Integer	Number of kids (generally under 2 years)	0
Infants	Integer	Number of infant passengers (0-2 years)	0
SeatClass	String	Preferred seat class (Economy, Business, or First Class)	"Economy"
TripType	String	Type of trip (One Way, Round Trip, Multi City)	"One Way"

SessionID	String	Unique session identifier	"{{SessionID}}"
PreferredAPI	Array of Strings	Preferred flight search API(s)	["sabre", "amadeus", "travelport", "usba", "flyhub"]
Segments	Array of Objects	Flight segments details	[{ "Origin": "DAC", "Destination": "CGP", "DepartureDateTime": "2025-01-01" }]
PreferredAirline	Array of Strings (Optional)	Preferred airlines to be considered in search	["EK", "EY", "MH", "BS", "BG"] (Use later)
ExcludedAirlines	Array of Strings (Optional)	Airlines to exclude from the search	["EK", "EY", "MH", "...] (Use later)

Note: Some fields are marked for future use (e.g., **PreferredAirline** and **ExcludedAirlines**).

8.2.3 Example Request:

```
{
  "Adults": 1,
  "Children": 0,
  "Kids": 0,
  "Infants": 0,
  "SeatClass": "Economy",
  "TripType": "One Way",
  "SessionID": "{{SessionID}}",
  "PreferredAPI": ["sabre", "amadeus", "travelport", "usba",
"flyhub"],
  "Segments": [
    {
      "Origin": "DAC",
      "Destination": "CGP",
      "DepartureDateTime": "2025-01-01"
    }
  ]
}
```

8.2.4 Response:

The API will return a list of flight options that match the search criteria, including details like flight timings, seat availability, and pricing. The response may also include additional information, such as flight durations and layovers if applicable.

8.2.5 Notes:

- **Adults, Children, Kids, and Infants** must be specified to correctly calculate pricing and availability.
- **SeatClass** defines the class preference for the flight.
- **TripType** should reflect whether the user is looking for a **one-way, round-trip, or multi-city** journey.
- **PreferredAPI** indicates the preferred sources for flight data, though this is optional.
- **Segments** should include the necessary flight details for the requested trip (**origin, destination, and date**).

9. Search Itinerary Details API Documentation

The Search Itinerary Details API retrieves detailed information about a specific itinerary based on a given **SessionID** and **ItineraryID**. This API is essential for fetching details of a travel plan for review or further operations.

9.1 Request Format:

- **Method:** POST
- **Endpoint:** `{{base_url}}/itinerary/search`

Request Body:

Key	Type	Description	Example
SessionID	String	A unique identifier for the current session.	"ABC123XYZ456"
ItineraryID	String	The unique identifier of the itinerary to fetch.	"IT12345678"

9.2 Example Request:

```
{
  "SessionID": "{{SessionID}}",
  "ItineraryID": "{{ItineraryID}}"
}
```

9.3 Example with Sample Data:

```
{
  "SessionID": "ABC123XYZ456",
  "ItineraryID": "IT12345678"
}
```

9.4. Response:

9.4.1 Success Response:

Key	Type	Description	Example
Status	String	Indicates the success of the operation.	"Success"
ItineraryID	String	The requested itinerary's unique identifier.	"IT12345678"
Details	Object	Contains the detailed information of the itinerary.	Refer to example response below.

9.4.2 Example Success Response:

```
{
  "Status": "Success",
  "ItineraryID": "IT12345678",
  "Details": {
    "FlightDetails": {
      "Airline": "Airline Name",
      "FlightNumber": "AI123",
      "DepartureTime": "2025-01-02T10:00:00",
      "ArrivalTime": "2025-01-02T14:00:00",
      "Origin": "JFK",
      "Destination": "LAX"
    },
    "PassengerDetails": [
      {
        "Title": "Mr",
        "FirstName": "John",
        "LastName": "Doe",
        "Type": "ADT",
      }
    ]
  }
}
```

```

        "Gender": "M"
    }
],
"PriceDetails": {
    "Currency": "USD",
    "TotalAmount": "450.00"
}
}
}

```

9.5 Error Response:

Key	Type	Description	Example
ErrorCode	String	Code identifying the specific error encountered.	"ERR_INVALID_SESSION"
Message	String	Error message detailing why the request failed.	"The session is invalid or expired."

9.5.1 Example Error Response:

```

{
  "ErrorCode": "ERR_INVALID_SESSION",
  "Message": "The session is invalid or expired."
}

```

Notes:

1. Ensure that the **SessionID** is valid and corresponds to an active session.
2. The **ItineraryID** must exist and be associated with the provided **SessionID**.
3. Handle error responses gracefully to prompt users for corrective actions, such as starting a new session or re-checking the **ItineraryID**.
4. The Details object in the success response includes information about flights, passengers, and pricing.

10. Passenger Booking API

The **Passenger Booking API** allows the user to provide detailed passenger information for booking a flight. This includes essential personal details, passport information, and contact details for each passenger.

10.1 Request Body:

Key	Type	Description	Example
SessionID	String	Unique session identifier linking the booking request to a session	"{{SessionID}}"
ItineraryID	String	Unique identifier for the selected itinerary	"{{ItineraryID}}"
Passenger	Array of Objects	List of passengers with their respective details. Each passenger object contains the following fields:	-
Passenger.Title	String	Title of the passenger (e.g., Mr, Mrs, Miss, Mstr)	"Mr"
Passenger.FirstName	String	Passenger's first name	"Abir"
Passenger.LastName	String	Passenger's last name	"Hossain"
Passenger.Type	String	Passenger type (e.g., ADT for Adult, CHD for Child, INF for Infant)	"ADT"
Passenger.Gender	String	Gender of the passenger (M for Male, F for Female)	"M"

Passenger.DateOfBirth	String	Date of birth of the passenger in YYYY-MM-DD format	"1978-09-18"
Passenger.Email	String/Null	Email address of the passenger (can be null for infants)	"test@abc.com"
Passenger.CountryPhoneCode	String	Country code for the passenger's phone number	"+880"
Passenger.Phone	String/Null	Phone number of the passenger (can be null for infants)	"1729963311"
Passenger.PassportNumber	String	Passport number of the passenger	"ZT12342342234"
Passenger.PassportExpiryDate	String	Expiry date of the passport in YYYY-MM-DD format	"2030-09-18"
Passenger.CountryCode	String	Country code of the passenger	"BD"

10.2 Example Request:

```
{
  "SessionID": "{{SessionID}}",
  "ItineraryID": "{{ItineraryID}}",
  "Passenger": [
    {
      "Title": "Mr",
      "FirstName": "Abir",
      "LastName": "Hossain",
      "Type": "ADT",
      "Gender": "M",
      "DateOfBirth": "1978-09-18",
      "Email": "test@example.com",
      "CountryPhoneCode": "+880",
      "Phone": "1729963311",
      "PassportNumber": "ZT12342342234",
      "PassportExpiryDate": "2030-09-18",
      "CountryCode": "BD"
    }
  ]
}
```

Notes:

- The **Passenger** array allows multiple passengers, each represented by an individual object with complete details.
- Infant passengers can have `null` values for fields like **Email** and **Phone**.
- Ensure all required fields are provided to avoid booking errors.

11. Booking Confirmation API

The **Booking Confirmation API** is used to finalize and confirm a booking after all necessary passenger details and payment have been provided. This API ensures that the booking is validated and confirmed with the travel system.

Request Format:

Method: POST

Endpoint: `{{base_url}}/booking/confirm`

Request Body:

Key	Type	Description	Example
SessionID	String	A unique identifier for the user session, generated during the booking process.	" <code>{{SessionID}}</code> "
ItineraryID	String	A unique identifier for the selected itinerary, created during the search or selection process.	" <code>{{ItineraryID}}</code> "

Example Request:

```
{
  "SessionID": "{{SessionID}}",
  "ItineraryID": "{{ItineraryID}}"
}
```

}

Response:

Success Response:

Key	Type	Description	Example
BookingID	String	Unique identifier for the confirmed booking.	"BKG1234567890"
Status	String	Status of the booking confirmation (e.g., Confirmed, Pending).	"Confirmed"
Message	String	Confirmation message detailing the success of the booking process.	"Booking successfully confirmed."
ItineraryDetails	Object	Detailed information about the confirmed itinerary, including flight details, passenger details, etc.	{...}

Error Response:

Key	Type	Description	Example
ErrorCode	String	Code identifying the specific error encountered.	"ERR_SESSION_EXPIRED"
Message	String	Error message detailing why the booking confirmation failed.	"Session expired. Please initiate a new booking session."

Example Success Response:

```
{  
  "BookingID": "BKG1234567890",  
  "Status": "Confirmed",  
  "Message": "Booking successfully confirmed.",  
  "ItineraryDetails": {  
    "Flight": {  
      "Airline": "ABC Airlines",  
      "FlightNumber": "ABC123",  
      "Departure": "2025-01-01T10:00:00",  
      "Arrival": "2025-01-01T12:00:00",
```

```
    "Duration": "2h 0m"
  },
  "PassengerDetails": [
    {
      "Name": "Mr Abir Hossain",
      "Type": "ADT",
      "Seat": "12A",
      "MealPreference": "Vegetarian"
    }
  ]
}
```

Notes:

- **SessionID** must remain valid during the confirmation process; if it has expired, a new session must be initiated.
- The **ItineraryID** ensures the specific flight or travel package is confirmed.
- Successful confirmation returns a **BookingID**, which serves as the reference for any further modifications or cancellations.

11. Booking

The **Booking API** for Zoo Travel Technology allows users to book travel services such as flights, accommodations, and other travel-related services. This API provides a streamlined process for users to search, reserve, and finalize their travel bookings with the Zoo Travel platform.

11.1. Booking Details API

The **Booking Details API** retrieves comprehensive details of a specific booking identified by its **PNR**. This API is essential for fetching the booking information, including passenger details, itinerary, and ticketing status.

11.1.1 Request Format:

Method: POST

Endpoint: `{{base_url}}/booking/details`

11.1.2 Request Body:

Key	Type	Description	Example
SessionID	String	A unique identifier for the session in which the booking was made.	" <code>{{SessionID}}</code> "
ApiKey	String	The API key for authentication and access control.	"1s"
PNR	String	The Passenger Name Record for the booking whose details need to be retrieved.	"OJBZYB"

Example Request:

```
{
  "SessionID": "{{SessionID}}",
  "ApiKey": "1s",
  "PNR": "OJBZYB"
}
```

11.1.3 Response:

Success Response:

Key	Type	Description	Example
Status	String	Status of the booking (e.g., Confirmed , Pending , Cancelled).	"Confirmed"
PNR	String	Passenger Name Record of the booking.	"OJBZYB"
PassengerDetails	Array	List of passenger details, including name, type, and other personal information.	[...]
Itinerary	Object	Details of the travel itinerary, including origin, destination, and travel dates.	{...}
FareDetails	Object	Breakdown of the fare, including base fare, taxes, and total cost.	{...}

TicketStatus	String	Ticketing status of the booking (e.g., Ticketed, Not Ticketed).	"Ticketed"
---------------------	--------	---	------------

Error Response:

Key	Type	Description	Example
ErrorCode	String	Code identifying the specific error encountered.	"ERR_PNR_NOT_FOUND"
Message	String	Error message detailing why the request failed.	"PNR not found."

Example Success Response:

```
{
  "Status": "Confirmed",
  "PNR": "OJBZYZB",
  "PassengerDetails": [
    {
      "Title": "Mr",
      "FirstName": "John",
      "LastName": "Doe",
      "Type": "ADT",
      "Gender": "M",
      "DateOfBirth": "1980-01-01"
    }
  ],
  "Itinerary": {
    "Origin": "DAC",
    "Destination": "CGP",
    "DepartureDate": "2025-01-01",
    "ReturnDate": null
  },
  "FareDetails": {
    "BaseFare": 200.00,
    "Taxes": 50.00,
    "TotalCost": 250.00,
    "Currency": "USD"
  },
  "TicketStatus": "Ticketed"
}
```

Example Error Response:

```
{
  "ErrorCode": "ERR_SESSION_EXPIRED",
  "Message": "The session has expired. Please initiate a new session."
}
```

Notes:

- **SessionID** must be valid and active during the request.
- Ensure the **PNR** corresponds to a valid booking within the API's system.
- API responses include passenger details, itinerary, and fare breakdown, making it easier to display or manage booking information.
- Handle error responses gracefully for a seamless user experience.

11.2. Cancel Booking API

The **Cancel Booking API** allows users to cancel an existing booking identified by its **PNR** (Passenger Name Record) within the context of a valid session. It ensures that the booking is voided and any applicable cancellation rules are applied.

11.2.1 Request Format:

Method: POST

Endpoint: `{{base_url}}/booking/cancel`

11.2.2. Request Body:

Key	Type	Description	Example
SessionID	String	A unique identifier for the user session, generated during the booking process.	" <code>{{SessionID}}</code> "
PNR	String	The Passenger Name Record identifying the specific booking to be canceled.	" <code>{{PNR}}</code> "
GDS	String	The Global Distribution System used for the booking (e.g., 1s for Sabre, amadeus, travelport).	"1s"

Example Request:

```

{
  "SessionID": "{{SessionID}}",
  "PNR": "{{PNR}}",
  "GDS": "1s"
}

```

11.2.3 Response:

Success Response:

Key	Type	Description	Example
Status	String	Status of the cancellation process (e.g., Cancelled, Pending).	"Cancelled"
Message	String	Confirmation message detailing the success of the cancellation process.	"Booking successfully canceled."
CancellationFee	Number	The fee applied for the cancellation, if applicable.	50.00
RefundAmount	Number	The amount refunded to the user after deducting any fees.	150.00
Currency	String	The currency of the refund amount.	"USD"

Error Response:

Key	Type	Description	Example
ErrorCode	String	Code identifying the specific error encountered.	"ERR_PNR_NOT_FOUND"

Message	String	Error message detailing why the cancellation failed.	"PNR not found or invalid."
----------------	--------	--	-----------------------------

Example Success Response:

```
{
  "Status": "Cancelled",
  "Message": "Booking successfully canceled.",
  "CancellationFee": 50.00,
  "RefundAmount": 150.00,
  "Currency": "USD"
}
```

Example Error Response:

```
{
  "ErrorCode": "ERR_SESSION_INVALID",
  "Message": "The session is invalid or has expired. Please start a new session."
}
```

Notes:

- **SessionID** must remain valid during the cancellation process; if it has expired, a new session must be initiated.
- **PNR** must correspond to a booking within the given **GDS**.
- Cancellation fees and refund policies depend on the airline's or travel service provider's terms and conditions.
- Ensure proper handling of error codes for better user experience and error diagnostics.

12. Share PNR API

The **Share PNR API** is used to share a Passenger Name Record (PNR) with another party or system, typically for collaboration or external access purposes. This API provides functionality to transmit the PNR details securely to a specified **Pseudo City Code (PCC)**.

12.1 Request Format:

Method: POST

Endpoint: {{base_url}}//pnr/share

12.1.1 Request Body:

Key	Type	Description	Example
SessionID	String	A unique identifier for the session in which the API is invoked.	"{{SessionID}}"
PNR	String	The Passenger Name Record that needs to be shared.	"OIPMRI"
PCC	String	The Pseudo City Code to which the PNR will be shared.	"MZ4K"

Example Request:

```
{  
  "SessionID": "{{SessionID}}",  
  "PNR": "OIPMRI",  
  "PCC": "MZ4K"  
}
```

12.1.2 Response:

Success Response:

Key	Type	Description	Example
Status	String	Indicates the success of the PNR sharing operation.	"Success"
Message	String	Additional details about the operation.	"PNR shared successfully."
PNR	String	The Passenger Name Record that was shared.	"OIPMRI"
SharedWith	String	The Pseudo City Code (PCC) to which the PNR was shared.	"MZ4K"

Error Response:

Key	Type	Description	Example
ErrorCode	String	Code identifying the specific error encountered.	"ERR_INVALID_PNR"
Message	String	Error message detailing why the request failed.	"Invalid PNR provided."

Example Success Response:

```
{
  "Status": "Success",
  "Message": "PNR shared successfully.",
  "PNR": "OIPMRI",
  "SharedWith": "MZ4K"
}
```

Example Error Response:

```
{
  "ErrorCode": "ERR_INVALID_SESSION",
  "Message": "The session is invalid or expired. Please start a new session."
}
```

Notes:

- Ensure the **SessionID** is active and corresponds to an authenticated session.
- The **PNR** must be valid and already exist in the system.
- The **PCC** should be authorized to receive the shared PNR; otherwise, the request may fail.
- API responses confirm the status of the sharing operation, including the PNR and recipient PCC for auditing purposes.
- Handle error responses gracefully to inform the user of necessary corrective actions.