

## API DOCUMENTATION

**BASE URL:** <https://hotel-booking-app-backend-30q1.onrender.com>

**NOTE:** Please ensure to set *[Content-Type: application/json]* in the request headers.

**NOTE:** User and Vendor profile images would follow this src pattern: BASE URL/uploads/user.profileImage (e.g 1738600581987.jpg)

**NOTE:** OTPs expire in 10mins

**NOTE:** You can use the token from login responses for authorization in protected routes if it is required.

### Register Vendors

*BASE URL/api/vendors/register*

#### POST REQUEST

```
{
  "name": string,
  "businessName": string,
  "businessType": string,
  "email": string,
  "phone": string,
  "address": string,
  "branch" : string,
  "password": string,
  "role": string,
  "profileImage": string,
  "services": [string],
  "otp": String,
  "otpExpires": Date,
  "isVerified": { Boolean}
}
```

#### RESPONSE

```
{
  "message": string,
  "vendor": {
    "name": string,
    "email": string,
    "phone": string,
```

```
    "address": string,
    "password": string,
    "role": string,
    "profileImage": string,
    "services": [string],
    "_id": string,
    "createdAt": string,
    "updatedAt": string,
    "__v": 0
  }
}
```

=====

### **verify OTP**

*BASE URL/api/vendors/verify-otp*

POST REQUEST

```
{
  "email": sting,
  "otp": sting
}
```

RESPONSE

```
{"message": string}
```

=====

### **Resend OTP**

*BASE URL/api/vendors/resend-otp*

POST REQUEST

```
{
  "email": sting,
}
```

RESPONSE

```
{"message": string}
```

=====

## Vendor Login

*BASE URL/api/vendors/login*

### POST REQUEST

```
{  
  "email": string,  
  "password": string  
}
```

### RESPONSE

```
{  
  "message": string,  
  "profile": {  
    "id": string,  
    "name": string,  
    "businessName": string,  
    "email": string,  
    "address": string,  
    "branch": string,  
    "profileImage": string,  
    "services": [string],  
    "token": string  
  }  
}
```

=====

## get vendors

### GET REQUEST

*BASE URL/api/vendors?businessName=businessName goes here*

**Note:** vendors can be fetched using either businessName, businessType or branch or both as query parameters

**Authorization:** JWT Bearer

**secret:** diys684iyu2hpre87u386

## RESPONSE

```
[
  {
    "_id": string,
    "name": string,
    "businessName": string,
    "businessType": string,
    "email": string,
    "phone": string,
    "address": string,
    "branch": string,
    "role": string,
    "profileImage": string,
    "services": [string],
    "isVerified": boolean,
    "createdAt": date,
    "updatedAt": date,
  },
];
```

=====

## USER ENDPOINTS

### register users

*BASE URL/api/users/register*

## POST REQUEST

```
{
  "firstName": string,
  "lastName": string,
  "email": string,
  "password": string,
  "phone": string,
  "profileImage": string,
  "otp": String,
  "otpExpires": Date,
  "isVerified": { Boolean}
}
```

## RESPONSE

```
{
  "message": string,
  "user": {
    "firstName": string,
    "lastName": string,
    "email": string,
    "password": string,
    "profileImage": string,
    "_id": string,
    "createdAt": string,
    "updatedAt": string,
    "__v": 0
  }
}
```

=====

## verify OTP

*BASE URL/api/users/verify-otp*

## POST REQUEST

```
{
  "email": sting,
  "otp": sting
}
```

## RESPONSE

```
{
  "message": "Email verified successfully."
}
```

=====

## Resend OTP

*BASE URL/api/users/resend-otp*

## POST REQUEST

```
{
  "email": sting,
}
```

## RESPONSE

```
{"message": string}
```

=====

## login user

*BASE URL/api/users/login*

POST REQUEST

```
{
  "email": string,
  "password": string
}
```

RESPONSE

```
{
  "message": string,
  "token": string
}
```

=====

### **get user profile**

GET REQUEST

*BASE URL/api/users/profile/:id*

**Authorization:** JWT Bearer

**secret:** diys684iyu2hpre87u386

RESPONSE

```
{
  "id": string,
  "firstName": string,
  "lastName": string,
  "email": string,
  "phone": string,
  "profileImage": string,
  "createdAt": string,
  "updatedAt": string,
}
```

=====

### **SESSION ENDPOINT**

#### **create session**

*BASE URL/api/sessions/*

POST REQUEST

```
{
```

```
"userId": string,  
"token": string,  
"expiresAt": string  
}
```

## RESPONSE

```
{  
  "message": string,  
  "session": {  
    "userId": string,  
    "token": string,  
    "expiresAt": "2025-01-26T20:15:10.877Z",  
    "_id": string,  
    "createdAt": sting,  
    "updatedAt": string,  
    "__v": 0  
  }  
}
```

=====

## get session

### GET REQUEST

*BASE URL/api/sessions/:id*

## RESPONSE

```
{  
  "_id": sting,  
  "userId": string,  
  "token": string,  
  "expiresAt":string,  
  "createdAt": string,  
  "updatedAt": string,  
  "__v": 0  
}
```

=====

## delete session

### DELETE REQUEST

*BASE URL/api/sessions/delete/:id*

## RESPONSE

```
{ "message": string}
```

=====

## Create Menu

### POST REQUEST

*BASE URL/api/vendors/create-menu*

**Authorization:** Bearer Token

**secret:** {token}

**Note:** token is gotten when the vendor is logged in.

### REQUEST

```
{
  "addOns" : [string],
  "availabilityStatus" : boolean ,
  "category": string,
  "cuisineType" : string,
  "dietaryInfo" : [string],
  "discountPrice" : number,
  "dishName" : string,
  "description" : string,
  "itemImage" : string,
  "maxOrderPerCustomer" : number,
  "portionSize" : string,
  "preparationTime" : number,
  "price" : number,
  "spiceLevel" : string,
  "stockQuantity" : number
}
```

### RESPONSE

```
{
  "message": string,
  "menu": {
    "vendor": string,
    "addOns" : [string],
    "availabilityStatus" : boolean ,
    "category": string,
    "cuisineType" : string,
```



```
    "dietaryInfo" : [string],
    "discountPrice" : number,
    "dishName" : string,
    "description" : string,
    "itemImage" : string,
    "maxOrderPerCustomer" : number,
    "portionSize" : string,
    "preparationTime" : number,
    "price" : number,
    "spiceLevel" : string,
    "stockQuantity" : number
    "_id": string,
    "createdAt": date,
    "updatedAt":date,
    "__v": 0
  }
}
```

=====

### **get menus**

GET REQUEST

*BASE URL/api/vendors/menus?vendorId=vendorId goes here*

**Note:** menus can be fetched using either a vendorId, cuisineType,category or dishName as query parameters

**Authorization:** JWT Bearer

**secret:** diys684iyu2hpre87u386

## RESPONSE

```
{
  "success": boolean,
  "menus": [
    {
      "_id": string,
      "vendor": string,
      "addOns" : [string],
      "availabilityStatus" : boolean ,
      "category": string,
      "cuisineType" : string,
      "dietaryInfo" : [string],
      "discountPrice" : number,
      "dishName" : string,
      "description" : string,
      "itemImage" : string,
      "maxOrderPerCustomer" : number,
      "portionSize" : string,
      "preparationTime" : number,
      "price" : number,
      "spiceLevel" : string,
      "stockQuantity" : number
      "createdAt": date,
      "updatedAt":date,
      "__v": 0
    }
  ]
}
```

=====

## Create Booking

POST REQUEST

*BASE URL/api/users/bookings*

**Authorization:** Bearer Token

**secret:** {token}

**Note:** token is gotten when the user is logged in.

**Note:** This endpoint works for booking both restaurants and hotels

REQUEST (for Hotel booking. type = hotel)

```
{
  "type": string,
  "vendor": string,
  "roomNumber": number,
  "guests": number,
  "checkIn": date,
  "checkOut": date
}
```

REQUEST (for Restaurant booking. type = restaurant)

```
{
  "vendorId": string,
  "businessName": string,
  "location": string,
  "partySize": number,
  "menuId": string,
  "tableNumber": number,
  "tableType": string,
  "meal": string,
  "pricePerTable": number,
  "guests": number,
  "totalPrice": number,
  "specialRequest": string,
  "image": string,
  "date": string
}
```

## RESPONSE

```
{
  "success": boolean,
  "message": string,
  "booking": {
    "userId": string,
    "vendorId": string,
    "businessName": string,
    "location": string,
    "menuId": string,
    "partySize": number,
    "tableNumber": number,
    "tableType": string,
    "meal": string,
    "pricePerTable": number,
    "guests": number,
    "totalPrice": number,
    "specialRequest": string,
    "image": string,
    "date": string,
    "status": string,
    "_id": string,
    "bookingDate": string,
    "createdAt": string,
    "updatedAt": string,
    "__v": 0
  }
}
```

=====

## Get Bookings

### GET REQUEST

*BASE URL/api/users/booking*

**Note:** Bookings can be fetched using either vendor's id, user's id, the booking type or even a combination of any of the three as query parameters. Calling this endpoint without any query parameters would get all bookings from different vendors, which might be used in owners dashboard

examples:

*BASE URL/api/users/bookings?type=hotel*

*BASE URL/api/users/bookings?type=restaurant*

*BASE URL/api/users/bookings?vendorId={vendor's id here}*

**Authorization:** JWT Bearer

**secret:** diys684iyu2hpre87u386

## RESPONSE

```
[
  {
    "_id": string,
    "user": string,
    "type": string,
    "vendor": string,
    "roomNumber": number/null,
    "tableNumber": number/null,
    "checkIn": date,
    "checkOut": date,
    "status": string,
    "bookingDate": string
  }
]
```

=====

## Cancel Booking

### PATCH REQUEST

*BASE URL/api/users/bookings/cancel/{booking id}*

**Authorization:** Bearer Token

**secret:** {token}

**Note:** token is gotten when the user is logged in.

## RESPONSE

```
{
  "message": string,
  "booking": {
    "_id": string,
    "user": string,
    "type": string,
    "vendor": string,
    "menuId": string,
    "roomNumber": number/null,
    "tableNumber": number/null,
    "guests": number/null,
    "checkIn": date,
    "checkOut": date,
    "status": string,
    "bookingDate": date,
    "createdAt": date,
  }
}
```

```
    "updatedAt":date,
    "__v":0
  }
}
```

=====

## Update Booking

PATCH REQUEST

*BASE URL/api/users/bookings/update/{booking id}*

**Authorization:** Bearer Token

**secret:** {token}

**Note:** token is gotten when the user is logged in.

REQUEST (for Hotel booking. type = hotel)

```
{
  "type": string,
  "vendor": string,
  "roomNumber": number,
  "guests": number,
  "checkIn": date,
  "checkOut": date
}
```

REQUEST (for Restaurant booking. type = restaurant)

```
{
  "type": string,
  "vendor": string,
  "tableNumber": number,
  "guests": number,
}
```

RESPONSE

```
{
  "message":string,
  "booking":{
    "_id":string,
    "user":string,
    "type":string,
    "vendor":string,
    "menuId":string,
    "roomNumber":number/null,
  }
}
```

```
"tableNumber":number/null,
"guests":number/null,
"checkIn":date,
"checkOut":date,
"status":string,
"bookingDate":date,
"createdAt":date,
"updatedAt":date,
"__v":0
}
}
```

=====

## Add Bank Details

PATCH REQUEST

*BASE URL/api/vendors/save-payment*

**Authorization:** Bearer Token

**secret:** {token}

**Note:** token is gotten when the vendor is logged in.

REQUEST

```
{
  "businessName":string,
  "bankCode":string,
  "accountNumber"string,
  "percentageCharge":number
}
```

RESPONSE

```
{
  "message": string,
  "data": {
    "bankAccountName":string,
    "bankName": string,
    "bankCode": string,
    "accountNumber":string,
  }
}
```

```
    "paystackSubAccount": string",
    "recipientCode": string,
    "percentageCharge": number,,
    "splitGroupId": number,
    "split_code": string,
    "splitGroupName": string
  }
}
```

=====

### **Initialize payment**

POST REQUEST

*BASE URL/api/users/make-payment*

**Authorization:** Bearer Token

**secret:** {token}

**Note:** token is gotten when the user is logged in.

REQUEST

```
{
  "email": string
  "amount": string,
  "vendorId": string
}
```

RESPONSE

```
{
  "message": string,
  "data": {
    "authorization_url": string, (Note: used to redirect users to paystack payment page)
    "ref": string                (Note: this is the reference to be used in payment verification)
  }
}
```

=====

### **Verify payment**

POST REQUEST

*BASE URL/api/users/verify-payment*

**Authorization:** Bearer Token

**secret:** {token}

**Note:** token is gotten when the user is logged in.



## REQUEST

```
{
  "reference":string
}
```

## RESPONSE

```
{
  "message":string,
  "status":string,
  "transactionId":number,
  "amount": number,
  "currency": string
  "paid_at": string",
  "cerated_at": string,
  "channel":string ,
  "customer": {
    "id": number,
    "email":string",
    "customer_code":string
  },
  "subaccount": {
    "id": number,
    "subaccount_code":string,
    "business_name": string,
    "percentage_charge": number,
    "settlement_bank": string,
    "bank_id": number,
    "account_number": string,
    "active": boolean,
    "is_verified": boolean
  },
  "split": {
    "vendorPercentage": number,
    "platformPercentage": number,
    "subaccountCode": string
  }
}
```

=====

## Vendor Withdrawal

### POST REQUEST

*BASE URL/api/vendors/withdraw*

**Authorization:** Bearer Token

**secret:** {token}

**Note:** token is gotten when the vendor is logged in.

#### REQUEST

```
{
  "subaccount":string,
  "amount":number
}
```

#### RESPONSE

```
{
  "message": "Failed to initiate transfer.",
  "error": "You cannot initiate third party payouts as a starter business"
}
```

=====

#### Search Resturants

##### GET REQUEST

*BASE URL/api/users/restaurant-search?query=Sausage Bun*

**Authorization:** JWT Bearer

**secret:** diys684iyu2hpre87u386

##### RESPONSE

```
{
  "message": "Search Results",
  "data": [
    {
      "_id": "67be193d464e5689c3d3be00",
      "businessName": "Pause",
      "email": "testingsalem71@gmail.com",
      "phone": "08120839946",
      "address": "2 Ebenebe",
      "profileImage": null,
      "services": [
        "Restaurant"
      ],
      "createdAt": "2025-02-25T19:25:49.605Z",
      "menus": [

```

```
{
  "_id": "67be22af80a4341e6a6ebec7",
  "vendor": "67be193d464e5689c3d3be00",
  "description": "Two Mini Bun, scrambled eggs, tomato jam, sausage of choice,
Bacon jam gruyere cheese, chives,",
  "price": 19000,
  "category": "appetizer",
  "itemImage": null,
  "createdAt": "2025-02-25T20:06:07.652Z",
  "updatedAt": "2025-02-25T20:06:07.652Z",
  "__v": 0,
  "dishName": "Sausage Bun"
}
]
}
}
```