BookOrGetBooked

API Documentation

# Booking

## Create a Booking (POST)

POST /api/Booking

Creates a new booking for a given user and service.

**Request Body**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Description** |
| userId | int | yes | The ID of the user making the booking. |
| serviceId | int | yes | The ID of the service being booked. |
| timeSlot | string (date-time) | yes | Date/Time for the booking (ISO 8601). |

**Example JSON**:

|  |
| --- |
| {  "userId": 1,  "serviceId": 1,  "timeSlot": "2024-12-31T03:32:45.553Z"  } |

**Response:**

**200 OK** – Returns a JSON object containing the newly created booking details inside a Result<T> wrapper.

|  |
| --- |
| {  "data": {  "id": 1,  "userId": 1,  "serviceId": 1,  "timeSlot": "2024-12-31T03:32:45.553Z",  "status": "Pending"  },  "isSuccess": true,  "error": null  } |

**cURL**:

|  |
| --- |
| curl -X POST "https://localhost:7246/api/Booking" \  -H "accept: \*/\*" \  -H "Content-Type: application/json" \  -d '{  "userId": 1,  "serviceId": 1,  "timeSlot": "2024-12-31T03:32:45.553Z"  }' |

## Get a Booking by ID (GET)

GET /api/Booking/{id}

Retrieves a booking by its ID.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Type** | **In** | **Required** | **Description** |
| id | int | path | yes | The booking’s unique ID. |

**Response:**

* 200 OK – A JSON object representing the booking.
* 404 Not Found – If no booking is found for the given ID.

**Example Response**:

|  |
| --- |
| {  "id": 1,  "userId": 1,  "serviceId": 1,  "timeSlot": "2024-12-31T03:32:45.553",  "status": "Pending"  } |

**cURL**:

|  |
| --- |
| curl -X GET "https://localhost:7246/api/Booking/1" \  -H "accept: \*/\*" |

# Currency

## Get All Currencies (GET)

GET /api/Currency

Retrieves all available currencies in the system.

**Response**

* **200 OK** – An array of currencies.

**Example Response**:

|  |
| --- |
| [  {  "id": 1,  "code": "USD",  "name": "United States Dollar",  "symbol": "$",  "isEnabled": true  },  {  "id": 2,  "code": "EUR",  "name": "Euro",  "symbol": "€",  "isEnabled": true  },  {  "id": 3,  "code": "GBP",  "name": "British Pound",  "symbol": "£",  "isEnabled": true  },  {  "id": 4,  "code": "DKK",  "name": "Danish Krone",  "symbol": "kr.",  "isEnabled": true  }  ] |

**cURL**:

|  |
| --- |
| curl -X GET "https://localhost:7246/api/Currency" \  -H "accept: \*/\*" |

# Service

## Get All Services by User ID (GET)

GET /api/Service/user/{userId}

Retrieves all services for a specific provider (user).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Type** | **In** | **Required** | **Description** |
| userId | int | path | yes | The provider’s (user’s) ID. |

**Response**

* **200 OK** – An array of services belonging to the specified user.

**Example Response**:

|  |
| --- |
| [  {  "id": 1,  "name": "Consultation",  "description": "A 30-minute consultation session",  "price": 50,  "currencyId": 4,  "providerId": 1  },  {  "id": 2,  "name": "Full Service",  "description": "Complete service package",  "price": 200,  "currencyId": 4,  "providerId": 1  }  ] |

**cURL**:

|  |
| --- |
| curl -X GET "https://localhost:7246/api/Service/user/1" \  -H "accept: \*/\*" |

# Schemas

## BookingRequestDTO

Represents the request payload for creating a new booking.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Description** |
| userId | int | yes | The user creating the booking |
| serviceId | int | yes | The ID of the service to be booked |
| timeSlot | string (date-time) | yes | The date/time for the requested booking (ISO 8601 format) |

**Example**:

|  |
| --- |
| {  "userId": 1,  "serviceId": 1,  "timeSlot": "2024-12-31T03:32:45.553Z"  } |

# Additional Information

* **Authentication**: (Not yet implemented or documented—if your API eventually requires authentication, mention the Bearer token or other security scheme here.)
* **Error Handling**: The API uses a Result<T> object in some cases. On failures, it returns isSuccess = false and a descriptive error message.
* **Date Format**: All date-time values are in [ISO 8601](https://en.wikipedia.org/wiki/ISO_8601) format.

**Last Updated**: December 31, 2024.

This documentation is automatically discoverable via [Swagger](https://swagger.io/) at:

|  |
| --- |
| GET <https://localhost:7246/swagger/v1/swagger.json> |

or by visiting the **Swagger UI** endpoint (often https://localhost:7246/swagger).