

Scope

ParaBank is a web-based application that allows users to perform common banking activities in a safe, demo environment. The system supports three types of roles: **Guest, Customer, and Manager**.

- **Guest:** Can access public pages, recover login information, and register to become a Customer.
- **Customer:** Can log in and perform banking functions such as account overview, fund transfers, bill payments, requesting loans, updating profile, and viewing transactions.
- **Manager:** A conceptual role (admin) used for requirement modeling and testing scope. The Manager can register new customers, manage accounts, approve/reject loans, and perform system-level actions like deposit, withdraw, and account deletion.

The application is designed primarily for **software testing practice** and does not handle real transactions. Its goal is to help QA learners practice writing test cases, identifying requirements, performing exploratory testing, and validating application behavior.

In-Scope Features/Pages to be Tested:

- Authentication & Access: Login, Logout, Forgot Login Info, Registration.
- Account Management: Account Overview, Transaction History, Open New Account.
- Transactions: Fund Transfer, Bill Payment, Deposit, Withdrawal.
- Loan Module: Loan Request & Approval.
- Customer Profile: Update Info, Delete Account, Delete Customer.
- Public Pages: Home, About Us, Services, Contact Us.
- Find Transactions: Transaction search by date, range, amount, transaction ID

Scope

Restful-Booker is a publicly available RESTful API designed for practicing API testing. It provides endpoints that simulate a basic hotel booking system, allowing users to create, retrieve, update, and delete bookings. The system is lightweight, stateless, and intended primarily for manual and automated API testing practice.

The API supports two levels of access:

- **Unauthenticated User:** Can create new bookings and retrieve existing booking information.
- **Authenticated User (Token-based):** Can update or delete existing bookings using a valid authentication token.

The application is designed for testing API behavior, status codes, payload validation, authorization handling, and CRUD operations. Its main purpose is to help QA engineers practice creating test cases, validating REST endpoints, performing exploratory API testing, and implementing automation frameworks.

In-Scope Endpoints / Features to Be Tested:

1. Authentication

- **Create Token (POST /auth)**
Used to generate a token for authorized operations such as updating or deleting bookings.

2. Booking Operations

- **Create Booking (POST /booking):**
Validate booking creation with required fields (firstname, lastname, total price, deposit paid, booking dates).
- **Get Booking IDs (GET /booking):**
Retrieve all booking IDs and verify response structure.
- **Get Booking by ID (GET /booking/{id}):**
Validate booking details including name, price, and date fields.
- **Update Booking (PUT /booking/{id}):**
Full update of all booking fields, requiring valid token.

- **Partial Update (PATCH /booking/{id}):**
Partial modification of booking fields such as firstname, lastname, or price.
- **Delete Booking (DELETE /booking/{id}):**
Requires authentication. Validate successful deletion and correct status codes.

3. Booking Filters

- **Filter by Firstname (GET /booking?firstname=)**
- **Filter by Lastname (GET /booking?lastname=)**
- **Filter by Check-in / Check-out dates**
- **Combined filtering fields**
Validate that returned booking IDs match the filter criteria.

4. API Health Check

- **GET /ping:**
Verify service availability and server responsiveness.
-

Testing Focus Areas

- API request/response validation
- HTTP status codes (200, 201, 400, 403, 404, 405)
- Authentication & token handling
- JSON schema validation
- Positive, negative, and boundary test scenarios
- Error handling for invalid IDs, missing fields, or incorrect HTTP methods
- Data consistency after updates or deletions
- Response time and basic performance checks