

## Introduction:

This report documents the testing activities performed on the **Vacation Management System API**. The system allows employees to submit vacation requests and managers to review, approve, or reject them. The API is built using a Java Spring Boot backend with an H2 in-memory database.

### Database Overview:

### Employees Table

The EMPLOYEES table stores user information including role and remaining vacation days.

jdbc:h2:mem:vacationdb
 

EMPLOYEES

VACATION\_REQUESTS

INFORMATION\_SCHEMA

Users

H2 2.3.232 (2024-08-11)

Run

Run Selected

Auto complete

Clear

SQL statement:

SELECT \* FROM EMPLOYEES

SELECT \* FROM EMPLOYEES;

ID	NAME	EMAIL	IS_MANAGER	REMAINING_VACATION_DAYS
1	John Employee	john@company.com	FALSE	25
2	Sarah Manager	sarah@company.com	TRUE	30
3	Mike Employee	mike@company.com	FALSE	15
4	Lisa Manager	lisa@company.com	TRUE	30

(4 rows, 3 ms)

Edit

### Vacation Requests Table

The VACATION\_REQUESTS table stores all vacation requests with their status and dates.

Sample Data:

SELECT \* FROM VACATION\_REQUESTS;

ID	AUTHOR_ID	STATUS	RESOLVED_BY_ID	REQUEST_CREATED_AT	VACATION_START_DATE	VACATION_END_DATE
1	1	approved	2	2025-08-15 22:45:47.244685	2025-12-01	2025-12-05
2	1	approved	2	2025-08-15 22:45:47.244685	2025-11-10	2025-11-15
3	3	rejected	4	2025-08-15 22:45:47.244685	2025-12-10	2025-12-20
4	1	pending	null	2025-08-15 22:55:29.025202	2025-12-10	2025-12-15

(4 rows, 2 ms)

## API Endpoint Testing

### Employee Endpoints:

#### 1. GET /api/employee/{id}/requests

- Retrieves vacation requests for a specific employee.
- Supports status filtering (approved, pending, rejected).

GET <http://localhost:8080/api/employee/1/requests> Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
-----	-------	-------------	-----------

Body Cookies Headers (8) Test Results 200 OK • 12 ms • 928 B Save Response

{ } JSON Preview Visualize

```
16   "durationInDays": 5
17 },
18 {
19   "id": 2,
20   "author": {
21     "id": 1,
22     "name": "John Employee",
23     "email": "john@company.com",
24     "remainingVacationDays": 30,
25     "manager": false
26   },
27   "status": "approved",
28   "resolvedBy": {
29     "id": 2,
30     "name": "Sarah Manager",
31     "email": "sarah@company.com",
32     "remainingVacationDays": 30,
33     "manager": true
```

#### 2. Submits a new vacation request:

Vacation API Tests New Import

Vacation Test / Employee Endpoints / vacation request Save Share

POST <http://localhost:8080/api/employee/3/requests> Send

Params Authorization Headers (9) Body Scripts Settings Cookies Beautify

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☐ JSON

```
1 {
2   "authorId": 1,
3   "vacationStartDate": "2026-06-01",
4   "vacationEndDate": "2026-06-15"
5 }
```

Body Cookies Headers (8) Test Results 200 OK • 793 ms • 542 B Save Response

{ } JSON Preview Visualize

```
2   "id": 5,
3   "author": {
4     "id": 3,
5     "name": "Mike Employee",
6     "email": "mike@company.com",
7     "remainingVacationDays": 15,
8     "manager": false
9   },
10  "status": "pending",
11  "resolvedBy": null,
12  "requestCreatedAt": "2025-08-24T19:19:20.6794283",
13  "vacationStartDate": "2026-06-01",
14  "vacationEndDate": "2026-06-15",
15  "durationInDays": 15
```

### 3. GET /api/employee/{id}/remaining-days

- Returns the remaining vacation days for an employee.

The screenshot shows the Postman interface for a collection named 'Vacation Test'. The selected endpoint is 'Get remaining vacation days' under 'Employee Endpoints'. The request is a GET to 'http://localhost:8080/api/employee/3/remaining-days'. The response is a 200 OK with a status of 200 OK, 12 ms, and 255 B. The response body is shown in JSON format, displaying the remaining vacation days for employee 3.

Key	Value	Description
Key	Value	Description

## Manager Endpoints

### 1. GET /api/manager/requests

- Retrieves all vacation requests (for managers).

The screenshot shows the Postman interface for a collection named 'Vacation Test'. The selected endpoint is 'Get all requests' under 'Manager Endpoints'. The request is a GET to 'http://localhost:8080/api/manager/requests'. The response is a 200 OK with a status of 200 OK, 26 ms, and 1.94 KB. The response body is shown in JSON format, displaying a list of vacation requests.

```
77     "status": "pending",
78     "resolvedBy": null,
79     "requestCreatedAt": "2025-08-16T22:55:29.025202",
80     "vacationStartDate": "2025-12-10",
81     "vacationEndDate": "2025-12-15",
82     "durationInDays": 6
83   },
84   {
85     "id": 5,
86     "author": {
87       "id": 3,
88       "name": "Mike Employee",
89       "email": "mike@company.com",
90       "remainingVacationDays": 15,
91       "manager": false
92     },
93     "status": "pending",
94     "resolvedBy": null,
95     "requestCreatedAt": "2025-08-24T19:19:20.678428",
96     "vacationStartDate": "2026-06-01",
97     "vacationEndDate": "2026-06-15",
98     "durationInDays": 15
99   }
100 ]
```

## 2. PUT /api/manager/request/{id}/status

- Updates the status of a vacation request (approve/reject).

The screenshot shows a REST client interface for a PUT request to `http://localhost:8080/api/manager/request/4/status`. The request body is a JSON object: `{ "managerId": 4, "status": "rejected" }`. The response is a 200 OK status with a response time of 617 ms and a body size of 636 B. The response body is a JSON object: `{ "id": 4, "author": { "id": 1, "name": "John Employee", "email": "john@company.com", "remainingVacationDays": 25, "manager": false }, "status": "rejected", "resolvedBy": { "id": 4, "name": "Lisa Manager", "email": "lisa@company.com", "remainingVacationDays": 30, "manager": true } }`

Also rejected for id = 5.

Table:

SELECT * FROM VACATION_REQUESTS;						
ID	AUTHOR_ID	STATUS	RESOLVED_BY_ID	REQUEST_CREATED_AT	VACATION_START_DATE	VACATION_END_DATE
1	1	approved	2	2025-08-15 22:45:47.244685	2025-12-01	2025-12-05
2	1	approved	2	2025-08-15 22:45:47.244685	2025-11-10	2025-11-15
3	3	rejected	4	2025-08-15 22:45:47.244685	2025-12-10	2025-12-20
4	1	rejected	4	2025-08-15 22:55:29.025202	2025-12-10	2025-12-15
5	3	rejected	4	2025-08-24 19:19:20.678428	2026-06-01	2026-06-15

(5 rows, 2 ms)

Edit

## Test Scenarios & Results

### Positive Tests

- All endpoints return expected status codes (200, 201).
- Vacation days are correctly deducted upon approval.
- Filtering by status works as expected.

### Negative Tests

- **Invalid employee ID** → 404 Not Found
- **Overlapping dates** → 400 Bad Request
- **End date before start date** → 400 Bad Request
- **Non-manager trying to approve** → 403 Forbidden
- **Invalid status transition** → 400 Bad Request

### Conclusion

The Vacation Management System API has been thoroughly tested using Postman. All core functionalities are working as expected. Error handling is robust, and business rules (e.g., vacation day deduction, overlapping requests) are correctly enforced. The API is ready for production use.