# SOFTWARE APPLICATION TEST DOCUMENT

# reliserv

**Prepared By:**
QA Team / Nyo Me Han

**Organization:**
ServiceBridge Team

**Repository:**
https://github.com/ServiceBridgeteam/reliserv

**Date:**
[02-28-2026]

**Version:**
1.0

## 1. Document Revision History

| Version | Date | Description | Author |
|---|---|---|---|
| 1.0 | 02-28-2026 | Initial draft | Nyo Me Han |
| | | | |
| | | | |

## 2. Introduction

### 2.1 Purpose

This document describes the test plan, test cases, and test results for the reliserv application maintained by ServiceBridge Team. It is intended to verify that all functional and non-functional requirements are correctly implemented and that the system behaves reliably under expected and edge-case conditions.

### 2.2 Scope

This test document covers the following areas of the reliserv application:

- Core service reliability and uptime functionality
- API endpoint behavior and response validation
- Service registration, discovery, and health-check mechanisms
- Authentication and authorization workflows
- Error handling and fault tolerance
- Performance under normal and peak load conditions
- Integration with dependent external services

### 2.3 Intended Audience

This document is intended for QA engineers, developers, project managers, and stakeholders involved in the reliserv project.

### 2.4 References

| Document / Resource | Location / URL |
|---|---|
| GitHub Repository | https://github.com/ServiceBridgeteam/reliserv |
| API Documentation | [Link to API docs] |
| Requirements Specification | [Link to requirements doc] |
| Architecture Document | [Link to architecture doc] |
| Project Plan | |

## 3. Test Objectives

The primary objectives of this test effort are:

1. Verify that all specified functional requirements are correctly implemented.
2. Validate that the application handles error states and edge cases gracefully.
3. Confirm that performance benchmarks are met under load.
4. Ensure security controls are enforced and cannot be bypassed.
5. Confirm compatibility with supported environments and configurations.
6. Identify and document defects for remediation before release.

## 4. Test Environment

### 4.1 Environment Summary

| Parameter | Value |
|---|---|
| Environment Type | Staging / Pre-Production |
| Operating System | [e.g., Ubuntu 22.04 LTS / Windows Server 2022] |
| Runtime / Language Version | [e.g., Node.js 20.x / Go 1.22 / Java 21] |
| Database | [e.g., PostgreSQL 15 / Redis 7] |
| Containerization | [e.g., Docker 24 / Kubernetes 1.29] |
| CI/CD Pipeline | [e.g., GitHub Actions] |
| Test Framework | [e.g., Jest / Pytest / JUnit] |
| Base URL | https://[staging-host]/api |

### 4.2 Test Tools

| Tool | Purpose | Version |
|---|---|---|
| Postman / REST Client | API endpoint testing | 1.0 |
| GitHub Actions | CI pipeline and automated test runs | N/A |

### 4.3 Test Data

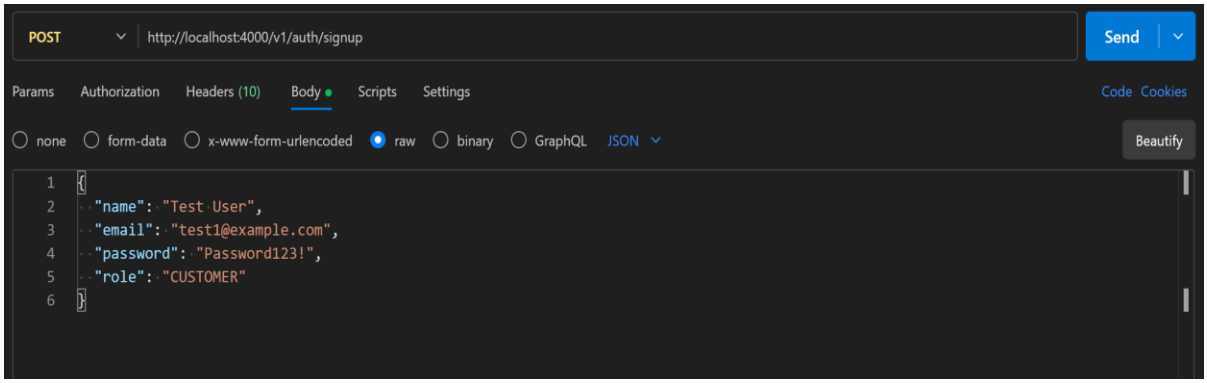
All test data used in this document is synthetic and does not contain real production data. Test data sets should be prepared in the test fixtures folder of the repository before execution.

## 5. Test Cases

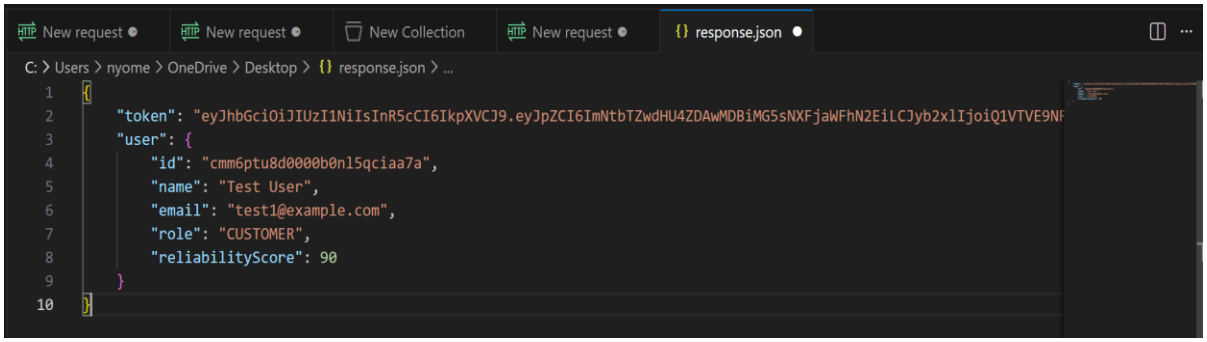
### 5.1 API Validation

#### 5.1.1 Authentication - Signup

### Request

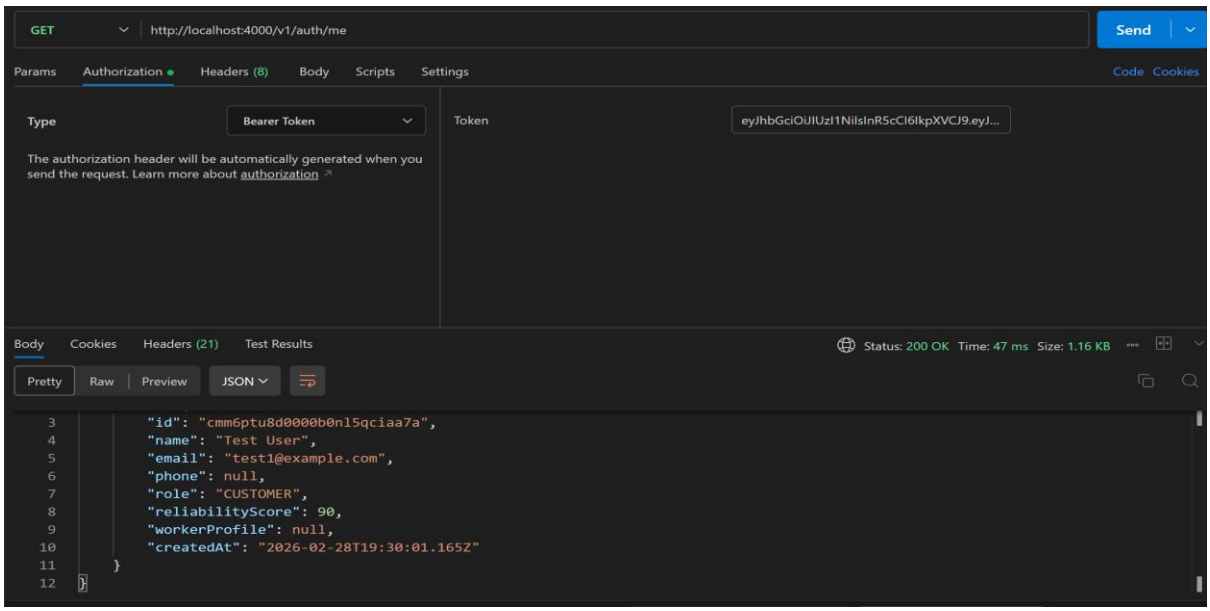



### Response

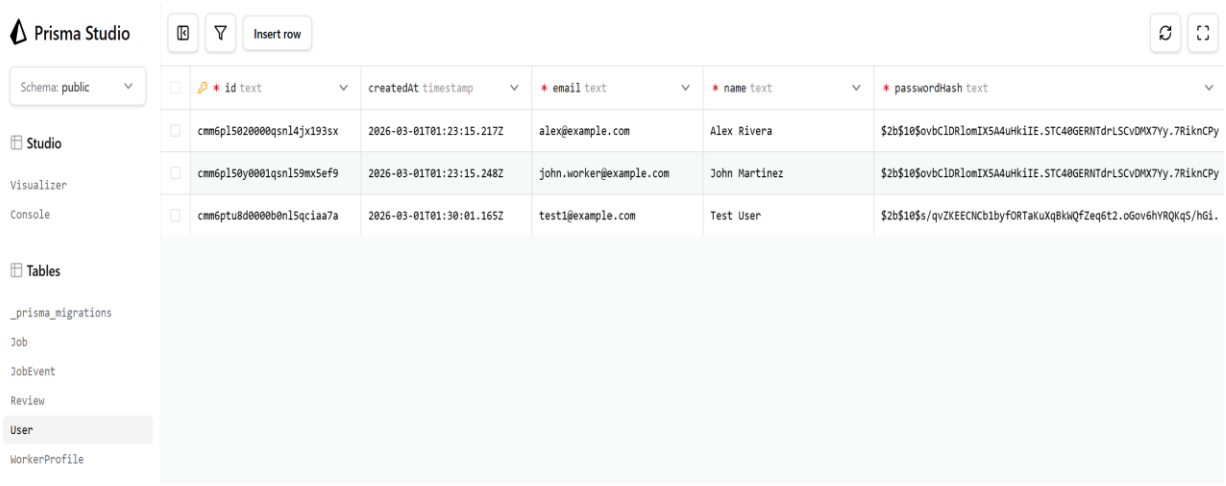



## 5.2 Configuration & Environment

### 5.2.1 JWT Middleware (Request - Response)



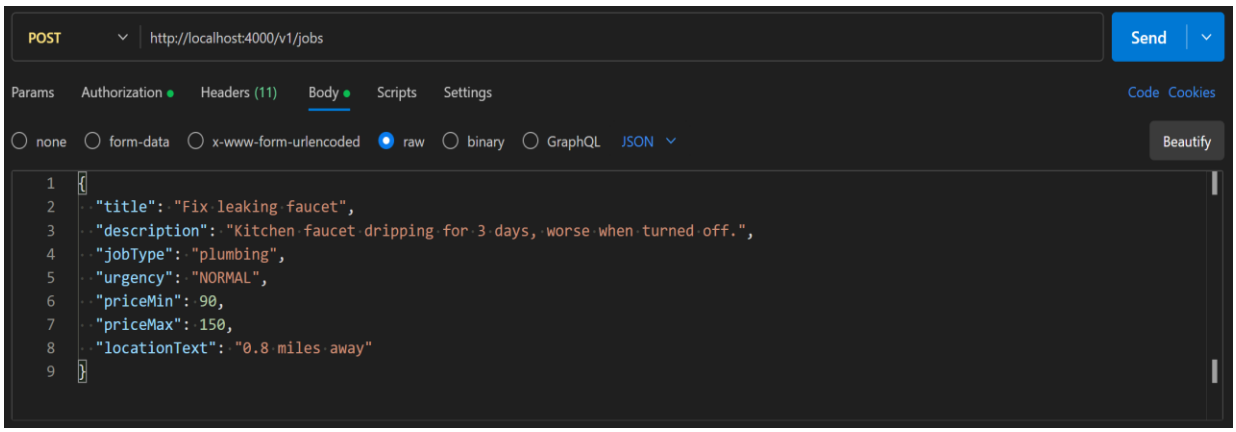
### 5.2.2 DB Proof — Prisma Studio (visual verification)




## 5.3 Core Service Functionality

### 5.3.1 Create Job

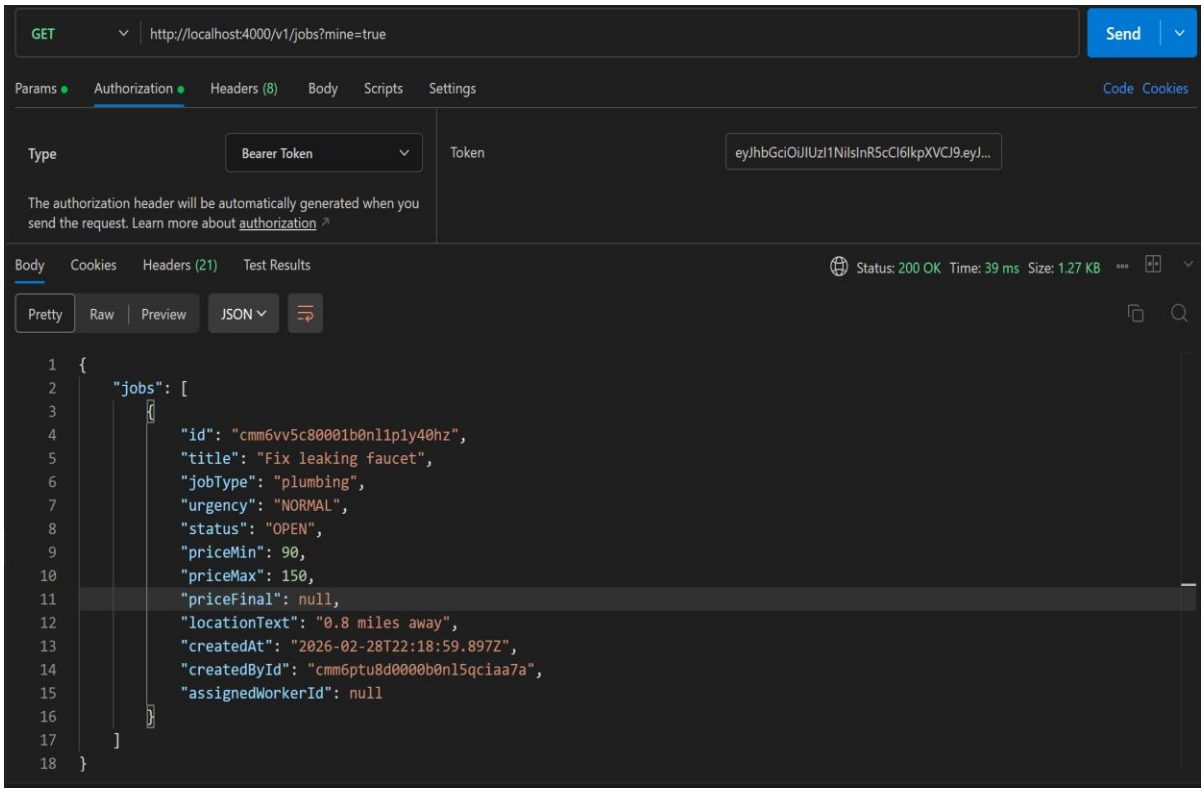
### Request



### Response


```
{
    "job": {
        "id": "cmm6vv5c80001b0nl1p1y40hz",
        "createdAt": "2026-02-28T22:18:59.897Z",
        "updatedAt": "2026-02-28T22:18:59.897Z",
        "title": "Fix leaking faucet",
        "description": "Kitchen faucet dripping for 3 days, worse when turned off.",
        "jobType": "plumbing",
        "urgency": "NORMAL",
        "status": "OPEN",
        "lat": null,
        "lng": null,
        "locationText": "0.8 miles away",
        "priceMin": 90,
        "priceMax": 150,
        "priceFinal": null,
        "lockedScope": null,
        "createdById": "cmm6ptu8d0000b0nl5qciaa7a",
        "assignedWorkerId": null,
        "events": [
            {
                "id": "cmm6vv5cb0002b0nl50i6dlsw",
                "createdAt": "2026-02-28T22:18:59.897Z",
                "jobId": "cmm6vv5c80001b0nl1p1y40hz",
                "actorId": "cmm6ptu8d0000b0nl5qciaa7a",
                "type": "CREATED",
                "note": "Job created"
            }
        ]
    }
}
```


## 5.3.2 List Jobs (Request – Response)

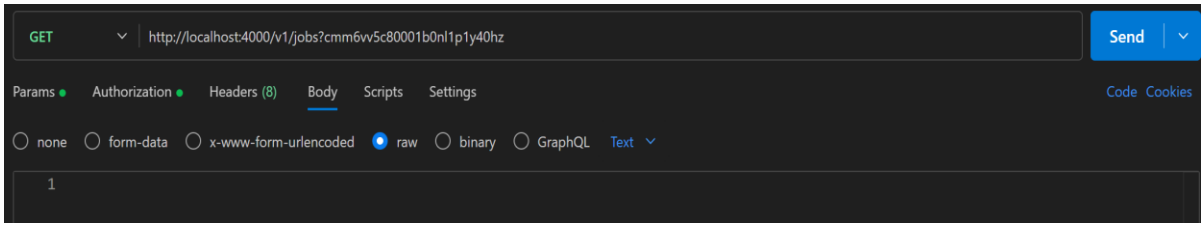




```
{
    "jobs": [
        {
            "id": "cmm6vv5c80001b0nl1p1y40hz",
            "title": "Fix leaking faucet",
            "jobType": "plumbing",
            "urgency": "NORMAL",
            "status": "OPEN",
            "priceMin": 90,
            "priceMax": 150,
            "priceFinal": null,
            "locationText": "0.8 miles away",
            "createdAt": "2026-02-28T22:18:59.897Z",
            "createdById": "cmm6ptu8d0000b0nl5qciaa7a",
            "assignedWorkerId": null
        }
    ]
}
```

### 5.3.3 Get Job by ID

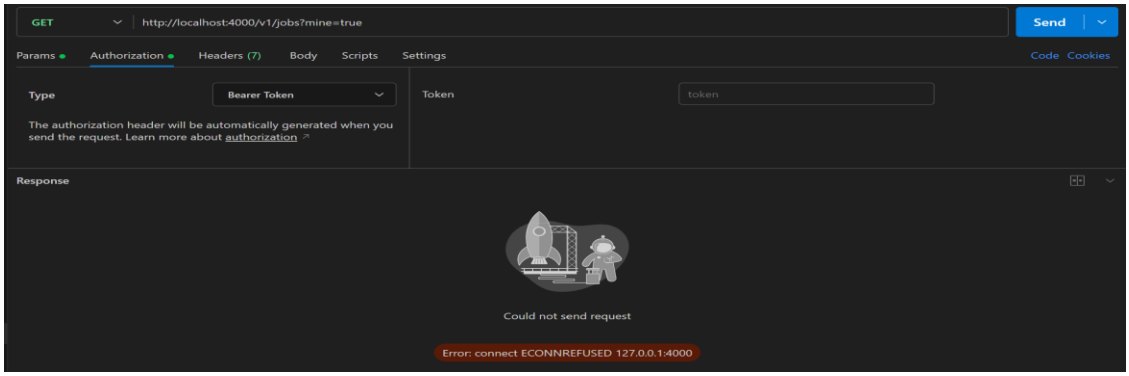
#### Request




#### Response


## 5.5 Security Proof

### 5.5.1 Validation Without Token




# 6. Defect Reporting

## 6.1 Defect Log

*Instructions: Record all defects discovered during testing below. Link to GitHub Issues where applicable.*

| Defect ID | Title | TC ID(s) | Severity | Steps to Reproduce | Expected | Actual | Status | GitHub Issue |
|---|---|---|---|---|---|---|---|---|
| DEF-001 | | | | | | | N.A | |
| DEF-002 | | | | | | | N.A | |
| DEF-003 | | | | | | | N.A | |
| DEF-004 | | | | | | | N.A | |

## 6.2 Severity Definitions

| Severity | Definition |
|---|---|
| Critical | System crash, data loss, security breach, or complete feature failure blocking core workflows. |
| High | Major feature broken with no workaround; significantly impacts usability. |
| Medium | Feature partially broken; workaround exists but is burdensome. |
| Low | Minor cosmetic or non-blocking issue; minimal user impact. |

# 7. Test Execution Summary

## 7.1 Execution Details

| Parameter | Value |
|---|---|
| Test Cases | 5.1 – 5.5 |
| Test Start Date | 02-28.2026 |

| Parameter | Value |
|---|---|
| Test End Date | 02-28.2026 |
| Executed By | Nyo Me Han |
| Environment | Development / Staging |

### 7.2 Test Results Summary

| Test Category | Total | Passed | Failed | Blocked | Not Run | Pass % |
|---|---|---|---|---|---|---|
| Authentication & Authorization | 1 | Yes | | | | 100 |
| Core Service Functionality | 3 | Yes | | | | 100 |
| API Validation & Error Handling | 0 | | | | | |
| Security Validation | 1 | Yes | | | | 100 |
| Performance Tests | 0 | | | | | |
| Configuration & Environment | 2 | Yes | | | | 100 |
| TOTAL | 7 | | | | | |

### 7.3 Outstanding Risks

| Risk | Likelihood | Impact | Mitigation |
|---|---|---|---|
| NA | NA | | |
| NA | NA | | |
| | | | |

## 8. Sign-Off

The following stakeholders have reviewed and approved this test document:

| Role | Name | Signature | Date |
|---|---|---|---|
| QA Lead | Nyo Me Han | NYO ME HAN | 02-28-2026 |
| Development Lead | | | |
| Product Owner / Manager | | | |
| Release Manager | | | |

## 9. Abbreviations

| Abbreviation | Definition |
|---|---|
| API | Application Programming Interface |
| CI/CD | Continuous Integration / Continuous Delivery |
| JWT | JSON Web Token |
| OOM | Out Of Memory |
| QA | Quality Assurance |
| RPS | Requests Per Second |
| SLA | Service Level Agreement |
| TC | Test Case |
| DEF | Defect |
| P95/P99 | 95th / 99th Percentile Latency |

## 10. Notes & Additional Observations

*Use this section to capture any additional observations, out-of-scope items, or information not covered elsewhere in this document.*

[Add notes here]