

QA Test Plan

Project: Todo App

Author: Kosta Terziski

Date: 2025-07-27

1. Overview

This test plan outlines the testing strategy for a simple Todo application with a React frontend and Node.js backend. The app includes user login, todo item creation, editing, deletion, and listing. API and UI test automation is implemented using Jest, Supertest, and Cypress.

2. Objectives

- Verify critical user flows: login, add/edit/delete todo items.
- Ensure backend API endpoints work correctly and securely.
- Confirm UI elements function properly and reflect backend changes.
- Validate error handling and edge cases.

3. Scope

In Scope:

- API testing: login, CRUD for todo items.
- UI testing: user interaction with the frontend.
- Functional testing of login and todo list.
- Automation of test cases using Jest + Supertest and Cypress.

Out of Scope:

- Performance and load testing.
- Extensive cross-browser testing.

4. Test Items

- React Frontend (client/)
- Node.js Backend (server/)
- API endpoints:
 - POST /login

- GET /items
- POST /items
- PUT /items/:id
- DELETE /items/:id

5. Test Strategy

5.1 Unit & API Testing (Jest + Supertest)

- Validate login with valid and invalid credentials.
- Create, retrieve, update, delete todo items.
- Test error responses (e.g., edit/delete non-existent items).

5.2 UI Testing (Cypress)

- Simulate user login.
- Add/edit/delete todo items.
- Validate UI updates and state.
- Use Page Object Model for maintainability.

6. Environment

- Frontend: React (Create React App)
- Backend: Node.js + Express
- Test Frameworks: Jest + Supertest (API), Cypress (UI)
- OS: Windows 10
- Node version: $\geq 18.x$

7. Test Data

Username: admin

Password: admin

Sample todo item:

```
{  
  "text": "Buy groceries"  
}
```

8. Test Cases Summary

API Test Cases:

TC1 - Login with correct credentials -> 200 OK + token
TC2 - Login with wrong credentials -> 401 Unauthorized
TC3 - Create a todo -> 201 Created
TC4 - Retrieve all todos -> 200 OK + list of items
TC5 - Update a todo -> 200 OK + updated item
TC6 - Delete a todo -> 204 No Content
TC7 - Update non-existent todo -> 404 Not Found

UI Test Cases:

TC8 - User login -> Todo list is shown
TC9 - Add a todo -> Item appears in the list
TC10 - Edit a todo -> Item is updated
TC11 - Delete a todo -> Item is removed from list
TC12 - Login with invalid credentials -> Error message shown

9. Risk Assessment

- Backend fails to return proper token -> Validate response structure in tests
- Token not used in requests -> Add tests for unauthorized access
- UI doesn't reflect API changes -> Cypress tests check state after each action

10. CI/CD & Coverage (Optional)

- GitHub Actions for automated testing on push.
- Use Istanbul (via `jest --coverage`) to measure backend test coverage.
- Cypress coverage via `cypress-coverage` plugin.

11. Deliverables

- API test suite (server/tests/api.test.js)
- Cypress UI suite (client/cypress/e2e/...)
- Test Plan document (this)