Nodebucket

TECHNICAL DESIGN DOCUMENT

|  |  |
| --- | --- |
| **Author** | Walter McCue |
| **Last Updated** | 4/16/2023 12:35 PM |
| **Version Number** | 3.0.1 |

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason for Change | Version |
| Krasso, Richard | 10/17/2019 | New document format | 3.0.0 |
| McCue, Walter | 03/19/2023 | Web-450 Week 1 Update | 3.0.1 |
| McCue, Walter | 03/26/2023 | Web-450 Week 2 Update | 3.0.2 |
| McCue, Walter | 04/08/2023 | Web-450 Week 3 Update | 3.0.3 |
| McCue, Walter | 04/10/2023 | Web-450 Test Cases Update | 3.0.4 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[SECTION 1: INTRODUCTION 3](#_Toc32226123)

[1.1. Purpose 3](#_Toc32226124)

[1.2. Terminology 4](#_Toc32226125)

[1.3. User Personas 5](#_Toc32226126)

[1.4. User Stories 7](#_Toc32226127)

[1.5. Time Estimations 8](#_Toc32226128)

[SECTION 2: PROCESS DESIGN 9](#_Toc32226129)

[2.1. Prototypes 9](#_Toc32226130)

[2.2. ORD (Object Relational Diagram(s) 14](#_Toc32226131)

[2.3. NoSQL Document Diagram 15](#_Toc32226132)

[2.4. NoSQL Data Structure 16](#_Toc32226133)

[SECTION 3: QA TESTING 17](#_Toc32226134)

[3.1. QA Test Plan 17](#_Toc32226135)

[3.2. Unit Tests 26](#_Toc32226136)

[SECTION 4: REFERENCES 34](#_Toc32226137)

# SECTION 1: INTRODUCTION

## Purpose

The purpose of the nodebucket project is to allow authorized users the ability to create a to-do list, modify the list, move items to a done category, or delete items from the lists altogether. The chosen theme will be represented by the Hufflepuff House from the Harry Potter series. The application will be built using multiple technologies such as MongoDB, Angular, Node, and a few others. This document outlines the specific technologies used in table 1.2. This document also outlines the types of users we expect to use this application and the development schedule over the coming weeks.

## Terminology

|  |  |
| --- | --- |
| **Name** | **Comments** |
| MEAN Stack | MongoDB, Express, Angular, and Node.js |
| VS Code | Development IDE |
| SoapUI | API testing tool |
| MongoDB Compass | Database development IDE |
| Angular Material | <https://material.angular.io/> |
| Flex-layout | <https://github.com/angular/flex-layout> |
| primeNG | https://www.primefaces.org/primeng/#/ |
| Courses GitHub repository | https://github.com/buwebdev |
| Slack | Collaboration |
| Zoom | Virtual meetings |
| Trello | Task management software |
| OneDrive | Document sharing |
| GitHub | Source control |
| Blackboard | LMS (Learning Management System) |
| Office 365 | Email client |
| Git training | https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow |
| Regular expression cheat sheet | http://web.mit.edu/hackl/www/lab/turkshop/slides/regex-cheatsheet.pdf |
| Regular express training | https://regexr.com/ |

## User Personas

# Persona 1: Rebecca Turner

## Age: 31

## Gender: F

## Job Title: Accountant

## Marital Status: S

## Short Bio: Rebecca is a full-time accountant and has very poor organizational skills. She currently relies on hundreds of sticky notes plastered all over her work space to remind her of tasks, deadlines, and client callbacks. While she does have a method to her madness it is complete an udder chaos to everyone else, including her secretary. As such, when she meets in person with a client, she typically schedules a meeting at a nearby coffee shop. Moving from her office to the coffee shop and back throughout the day is not an efficient workflow. The nodebucket application would provide her a secure way to organize tasks and help remove her dependency on sticky notes. Then she could begin holding meetings in her office without losing time traveling back and forth a few times a day.

# Persona 2: Marge Simpson

## Age: 42

## Gender: F

## Job Title: Housewife

## Marital Status: M

## Short Bio: Marge is a stay-at-home wife and mother to three beautiful children. She does all the cooking, cleaning, and caretaking around the house so her husband can provide financially for their needs. She has the overall schedule memorized for when her husband leaves for work and when the two elementary aged children leave for school. What she needs is a way to set up a list of chores and tasks to be done throughout the week, prioritizing bigger tasks when she has alone time with the baby. She would like some way to delegate tasks to her husband and children too, but the important features are the ability to create and update tasks as needed.

# Persona 3: Michael Dickerson

## Age: 41

## Gender: M

## Job Title: Crypto Trader

## Marital Status: Divorced

## Short Bio: Michael lives in a persistent state of high-level-stress and perceives almost everything as a threat against him in some way, including his mother as he lives in her basement. His only means of decompression is smoking marijuana and trading cryptocurrency. He has an entire room dedicated to mining cryptocurrencies as well as his main workstation that is used to make larger trades and move different parts of his operation around. He believes that his bank and the authorities are trying to shut down his operation and take his hard-earned virtual currency away from him. He would like an application such as nodebucket to help him securely manage a list of tasks that he has to complete on a daily and hourly basis. He wants to be able to log out so if he was to ever leave his workstation unlocked, his mother would not be able to walk in and see how he manages his operation.

## User Stories

**Rebecca Turner**

|  |  |
| --- | --- |
| Story | Points |
| As an accountant, I need to see a to-do list to prioritize my time | 3 |
| As an accountant, I need to be able to delete tasks if the client cancels | 1 |
| As an accountant, I need to make a list of financial products to buy, so that I can increase my client net worth | 3 |
| As an accountant I need the ability to log in to see my to-do list to keep them secure | 5 |
| As an accountant I need this to-do list for my team to stay on track | 3 |

**Marge Simpson**

|  |  |
| --- | --- |
| Story | Points |
| As a housewife, I need to access the application on a mobile device so I can manage my list on the go | 5 |
| As a housewife I need to be able to move my done to-do items to a done column so I can see what I've accomplished | 3 |
| As a housewife, I need a to do list to get chores done around the house, so they are all completed on time | 3 |
| As a housewife, I need my husband to see the updated list, so he doesn't forget to-do things | 3 |
| As a housewife, I need a to-do list so my children can do their chores. | 3 |

**Michael Dickerson**

|  |  |
| --- | --- |
| Story | Points |
| As a crypto trader, I need to be able to reorganize my to-do list to I can see what is more important to me. | 5 |
| As a crypto trader, I need to be able to log in, so my trading to-do list is secure | 5 |
| As a crypto trader I need to see what is done, so I can keep track of my clients | 3 |
| As a crypto trader, I need to bold important items, so I don't forget to trade crypto | 1 |
| As a crypto trader, I need to sign-out, so my information is secure. | 3 |

## Time Estimations

Development for the nodebucket project is expected to take 5 weeks.

**Week 1:** Planning, Prototyping, and initial project set-up.

**Week 2:** Sprint 1 [ Ord structure, MongoDB set-up, API set-up, log-in feature, log-out feature, UI set-up]

**Week 3:** Sprint 2 [ API refining, to-do feature set-up, UI refining ]

**Week 4:** Sprint 3 [ API finalization, to-do feature refining, UI finalization ]

**Week 5:** Testing, Deployment, Deliverable

**Backlog:** Additional features if time allows [ Mobile Friendly, !important items listed in bold ]

# SECTION 2: PROCESS DESIGN

## 2.1. Prototypes

<https://marvelapp.com/prototype/878c2j0>

1. Log In Screen
2. Log In Error
3. Home Screen (shows added tasks)
4. Tasks moved around
5. Delete Task
6. About
7. Contact
8. Sign Out
9. 404 Error

Graphical user interface, website

Description automatically generated

Graphical user interface, website

Description automatically generated

Graphical user interface, website

Description automatically generated

Graphical user interface, website

Description automatically generated

Graphical user interface, website

Description automatically generated

Graphical user interface, website

Description automatically generated

Timeline

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

## 2.2. ORD (Object Relational Diagram(s)

Diagram

Description automatically generated

## 2.3. NoSQL Document Diagram

N/A

## 2.4. NoSQL Data Structure



# SECTION 3: QA TESTING

## 3.1. QA Test Plan

**Project Name:** Nodebucket < https://github.com/walter-mccue/nodebucket>

**Test Case 1 User Log In**

|  |  |
| --- | --- |
| **Test Executed by**: Walter McCue | **Test Execution date:** 4/10/23 |
| **Description:** Nodebucket Test Cases |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Expected** | **Actual Results** | **(Pass/Fail)** | **Notes** |
| 1 | Navigate to http://localhost:3000 | Log In Screen is displayed | Log In Screen is displayed | Pass |  |
| 2 | Enter a known valid empId: “1007” | User Enters “1007” | User Enters “1007” | Pass |  |
| 3 | Hit the “Log In” button | User is logged in | User is logged in | Pass |  |
| 4 | View the Nodebucket page | Home Page is displayed | Home Page is displayed | Pass |  |

|  |
| --- |
| Comments: This test case requires the user to run the “npm run dev” command in a terminal window to host the server. The user will navigate to “localhost:3000” and attempt to log in with a known valid empId “1007”. Once logged in, the user will have access to the home page. |

**Test Case 2 User Log Out**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Expected** | **Actual Results** | **(Pass/Fail)** | **Notes** |
| 1 | Click the profile icon in the top right | Current Log In status is displayed | Current Log In status is displayed | Pass |  |
| 2 | Click the “Log Out” button | User is logged out | User is logged out | Pass |  |
| 3 | View the Log In screen | Log In screen is displayed | Log In screen is displayed | Pass |  |

|  |
| --- |
| Comments: This test case should be attempted directly after the previous test case. While the user is viewing the Home Page, they click the profile icon in the top right corner. A menu will drop down that shows the current log in status as well as a button to log the user out. Upon successful log out, the user will be navigated back to the log-in screen, |

**Test Case 3 Log In Failure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Expected** | **Actual Results** | **(Pass/Fail)** | **Notes** |
| 1 | Navigate to http://localhost:3000 | Log In Screen is displayed | Log In Screen is displayed | Pass |  |
| 2 | Enter a known invalid empId: “9999” | User Enters “9999” | User Enters “9999” | Pass |  |
| 3 | Hit the “Log In” button | User is not logged in | User is not logged in | Fail |  |
| 4 | View the error message | Red error message is displayed | Red error message is displayed | Fail |  |

|  |
| --- |
| Comments: This test case can be attempted directly after the previous test case OR can be done independently by running the “npm run dev” command in a terminal window to host the server. The user will enter a known invalid empId “9999”. After failing to log in, the user will view the error message and the webpage will stay on the Log In page. |

**Test Case 4 ToDo Task Creation**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Expected** | **Actual Results** | **(Pass/Fail)** | **Notes** |
| 1 | Navigate to http://localhost:3000 | Log In Screen is displayed | Log In Screen is displayed | Pass |  |
| 2 | Enter a known valid empId: “1007” | User Enters “1007” | User Enters “1007” | Pass |  |
| 3 | Hit the “Log In” button | User is logged in | User is logged in | Pass |  |
| 4 | View the Nodebucket page | Home Page is displayed | Home Page is displayed | Pass |  |
| 5 | Click in the task input field and input any task item between 3 and 35 characters | Task Item Input | Task Item Input | Pass |  |
| 6 | Hit Enter or the submit button to the right | Success message is generated and task item is added to the ToDo column | Success message is generated and task item is added to the ToDo column | Pass |  |
| 7 | View ToDo column to see new task | Task is displayed in the ToDo column | Task is displayed in the ToDo column | Pass |  |
| 8 | Resize window | Components in the page automatically resize depending on screen size | Components in the page automatically resize depending on screen size | Pass |  |

|  |
| --- |
| Comments: This test case requires the user to run the “npm run dev” command in a terminal window to host the server. The user will navigate to “localhost:3000” and attempt to log in with a known valid empId “1007”. Once logged in, the user will have access to the home page. On the home page, the user can create a new task item for the ToDo column. |

**Test Case 5 About Navigation**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Expected** | **Actual Results** | **(Pass/Fail)** | **Notes** |
| 1 | Navigate to http://localhost:3000 | Log In Screen is displayed | Log In Screen is displayed | Pass |  |
| 2 | Enter a known valid empId: “1007” | User Enters “1007” | User Enters “1007” | Pass |  |
| 3 | Hit the “Log In” button | User is logged in | User is logged in | Pass |  |
| 4 | View the Nodebucket page | Home Page is displayed | Home Page is displayed | Pass |  |
| 5 | In the Navigation bar, click the About link | About Us page is displayed | About Us page is displayed | Pass |  |
| 6 | Resize window | Components in the page automatically resize depending upon screen size | Components in the page automatically resize depending upon screen size | Pass |  |

|  |
| --- |
| Comments: This test case requires the user to run the “npm run dev” command in a terminal window to host the server. The user will navigate to “localhost:3000” and attempt to log in with a known valid empId “1007”. Once logged in, the user will have access to the home page. On the home page, the user can navigate to the About Us page. |

**Test Case 6 Contact Navigation**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Expected** | **Actual Results** | **(Pass/Fail)** | **Notes** |
| 1 | Navigate to http://localhost:3000 | Log In Screen is displayed | Log In Screen is displayed | Pass |  |
| 2 | Enter a known valid empId: “1007” | User Enters “1007” | User Enters “1007” | Pass |  |
| 3 | Hit the “Log In” button | User is logged in | User is logged in | Pass |  |
| 4 | View the Nodebucket page | Home Page is displayed | Home Page is displayed | Pass |  |
| 5 | Click the Contact link | Contact Us page is displayed | Contact Us page is displayed | Pass |  |
| 6 | Resize the window | Content in the page is resized based on screen size | Content in the page is resized based on screen size | Pass |  |

|  |
| --- |
| Comments: This test case requires the user to run the “npm run dev” command in a terminal window to host the server. The user will navigate to “localhost:3000” and attempt to log in with a known valid empId “1007”. Once logged in, the user will have access to the home page. On the home page, the user can navigate to the Contact Us page. |

**Test Case 7 Task Reordering**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Expected** | **Actual Results** | **(Pass/Fail)** | **Notes** |
| 1 | Navigate to http://localhost:3000 | Log In Screen is displayed | Log In Screen is displayed | Pass |  |
| 2 | Enter a known valid empId: “1007” | User Enters “1007” | User Enters “1007” | Pass |  |
| 3 | Hit the “Log In” button | User is logged in | User is logged in | Pass |  |
| 4 | View the Nodebucket page | Home Page is displayed | Home Page is displayed | Pass |  |
| 5 | Click in the task input field and input any task item between 3 and 35 characters | Task Item Input | Task Item Input | Pass |  |
| 6 | Hit Enter or the submit button to the right | Success message is generated and task item is added to the ToDo column | Success message is generated and task item is added to the ToDo column | Pass |  |
| 7 | View ToDo column to see new task | Task is displayed in the ToDo column | Task is displayed in the ToDo column | Pass |  |
| 8 | Drag and Drop the task item between the todo, doing, and done columns | Task is moved between the three columns | Task is moved between the three columns | Pass |  |

|  |
| --- |
| Comments: This test case requires the user to run the “npm run dev” command in a terminal window to host the server. The user will navigate to “localhost:3000” and attempt to log in with a known valid empId “1007”. Once logged in, the user will have access to the home page. On the home page, the user can create a new task item for the ToDo column. Once the task is added, the user can drag and drop the task between todo, doing, and done columns. |

**Test Case 8 Task Delete**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Expected** | **Actual Results** | **(Pass/Fail)** | **Notes** |
| 1 | Navigate to http://localhost:3000 | Log In Screen is displayed | Log In Screen is displayed | Pass |  |
| 2 | Enter a known valid empId: “1007” | User Enters “1007” | User Enters “1007” | Pass |  |
| 3 | Hit the “Log In” button | User is logged in | User is logged in | Pass |  |
| 4 | View the Nodebucket page | Home Page is displayed | Home Page is displayed | Pass |  |
| 5 | Click in the task input field and input any task item between 3 and 35 characters | Task Item Input | Task Item Input | Pass |  |
| 6 | Hit Enter or the submit button to the right | Success message is generated and task item is added to the ToDo column | Success message is generated and task item is added to the ToDo column | Pass |  |
| 7 | View ToDo column to see new task | Task is displayed in the ToDo column | Task is displayed in the ToDo column | Pass |  |
| 8 | Click “No Thanks” | Task is not deleted and user is notified that delete was canceled | Task is not deleted and user is notified that delete was canceled | Fail |  |
| 9 | Click the delete icon for that task | Confirmation dialog box appears | Confirmation dialog box appears | Pass |  |
| 10 | Click “Confirm” | Task is deleted and user is notified that the task is deleted | Task is deleted and user is notified that the task is deleted | Pass |  |

|  |
| --- |
| Comments: This test case requires the user to run the “npm run dev” command in a terminal window to host the server. The user will navigate to “localhost:3000” and attempt to log in with a known valid empId “1007”. Once logged in, the user will have access to the home page. On the home page, the user can create a new task item for the ToDo column. The user can then open a dialog box to confirm or cancel the deletion of that created task. |

**Test Case 9 Not-Found Error**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Expected** | **Actual Results** | **(Pass/Fail)** | **Notes** |
| 1 | Navigate to http://localhost:3000 | Log In Screen is displayed | Log In Screen is displayed | Pass |  |
| 2 | Enter a known valid empId: “1007” | User Enters “1007” | User Enters “1007” | Pass |  |
| 3 | Hit the “Log In” button | User is logged in | User is logged in | Pass |  |
| 4 | View the Nodebucket page | Home Page is displayed | Home Page is displayed | Pass |  |
| 5 | Edit the end of the URL to include random characters | 404 Error page is displayed | 404 Error page is displayed | Fail |  |
| 6 | Resize the window | Content in the page is resized based on screen size | Content in the page is resized based on screen size | Pass |  |
| 7 | Click the return to “Platform” link | Home Page is displayed | Home Page is displayed | Pass |  |
| 8 | Click the profile icon | You are signed in as \*\*\* menu appears | You are signed in as \*\*\* menu appears | Pass |  |
| 9 | Click “Log Out” | User is logged out and navigated to the login page | User is logged out and navigated to the login page | Pass |  |
| 10 | Close the Log out confirmation alert | Alert box closes | Alert box closes | Pass |  |
| 11 | Edit the end of the URL to include random characters | 404 Error page is displayed | 404 Error page is displayed | Fail |  |
| 12 | Click the return to “Platform” link | Login page is displayed | Login page is displayed | Pass |  |

|  |
| --- |
| Comments: This test case requires the user to run the “npm run dev” command in a terminal window to host the server. The user will navigate to “localhost:3000” and attempt to log in with a known valid empId “1007”. Once logged in, the user will have access to the home page. On the home page, the user can alter the URL to view the 404 error page. The return to “platform” link will take the logged in user back to the home page. After logging out, the user can alter the URL to view the 404 error page again. Clicking the link this time will return the user back to the login screen. |

## 3.2. Unit Tests

findEmployeeById (bad empId: foo)

**Graphical user interface, text, application

Description automatically generated**

findEmployeeById

Graphical user interface, text, application

Description automatically generated

findEmployeeById

Graphical user interface, text, application

Description automatically generated

findAllTasks

Graphical user interface, text, application

Description automatically generated

findAllTasks

Graphical user interface, text, application

Description automatically generated

findAllTasks (bad empId: foo)

Graphical user interface, text, application

Description automatically generated

findAllTasks (bad empId: 1019)

Graphical user interface, text, application

Description automatically generated

createTask

Graphical user interface, text, application

Description automatically generated

createTask (validation failure)

Graphical user interface, text, application

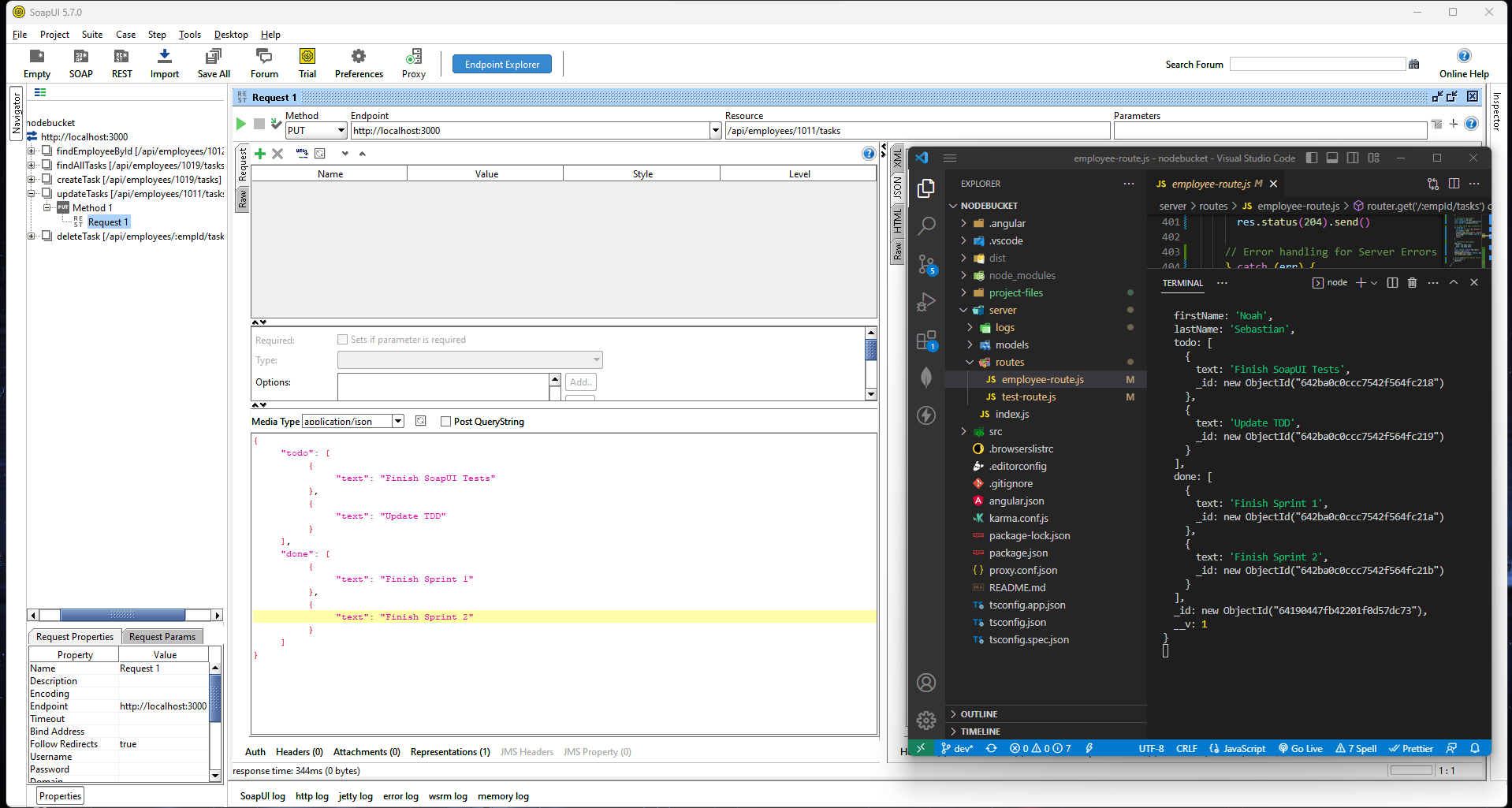
Description automatically generated

createTask (bad empId: 1019)

Graphical user interface, text, application

Description automatically generated

updateTasks



updateTaskes (bad empId)

Graphical user interface, text, application

Description automatically generated

updateTaskes (validation failure)

Graphical user interface, text

Description automatically generated

DeleteTask

Graphical user interface, text

Description automatically generated

DeleteTask (bad empId: 9999)

Graphical user interface, text, application

Description automatically generated

DeleteTask (bad taskId)

Graphical user interface, text, application

Description automatically generated

# SECTION 4: REFERENCES

[APA Reference List]