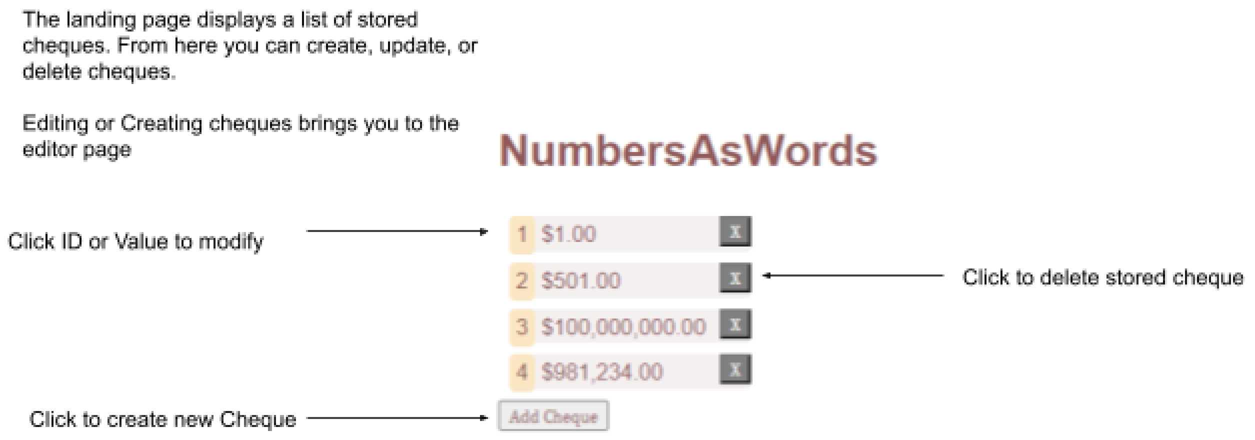
# NumbersAsWords

## Summary

The NumbersAsWords application allows users to manage amounts applied to stored cheques.
In order to ease the process of writing cheques, the application displays the cheque's amount as words.

## Application Overview



The landing page displays a list of stored cheques. From here you can create, update, or delete cheques.

Editing or Creating cheques brings you to the editor page

NumbersAsWords

Click ID or Value to modify

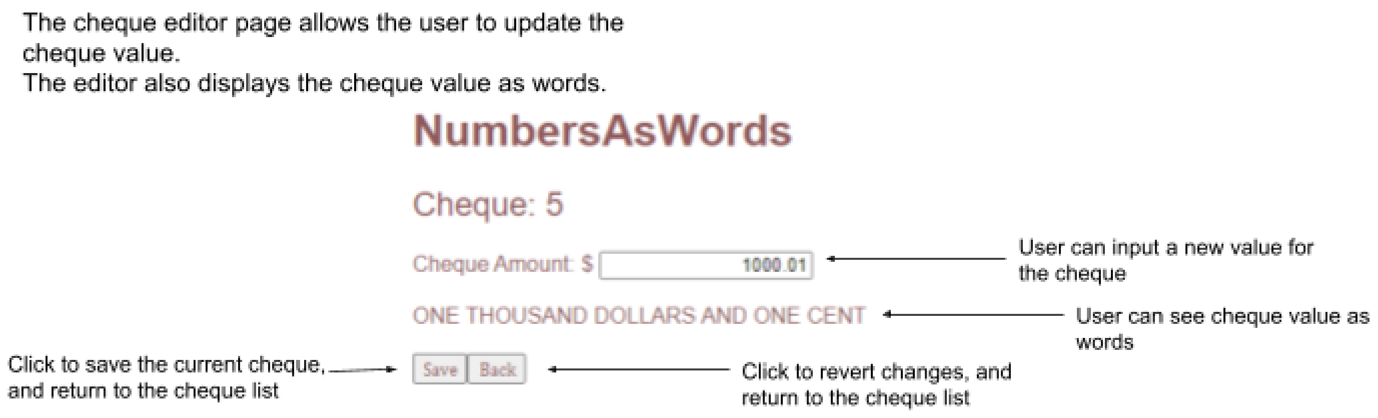
| | | |
|---|---|---|
| 1 | $1.00 | X |
| 2 | $501.00 | X |
| 3 | $100,000,000.00 | X |
| 4 | $981,234.00 | X |

Click to delete stored cheque

Click to create new Cheque

Add Cheque

---



The cheque editor page allows the user to update the cheque value.
The editor also displays the cheque value as words.

NumbersAsWords

Cheque: 5

Cheque Amount: $ 1000.01

User can input a new value for the cheque

ONE THOUSAND DOLLARS AND ONE CENT

User can see cheque value as words

Click to save the current cheque, and return to the cheque list

Save Back

Click to revert changes, and return to the cheque list

## Resources Used

- angular.io/guide/setup-local - For setup/configuration of NodeJS and Angular
- angular.io/tutorial - Tour Of Heroes tutorial for angular basics, routing, etc.
- typescriptlang.org/docs/ - For general Typescript documentation
- angular.io/guide/testing - For unit testing
- Stackoverflow.com - For clarification of documentation and errors

## Test Plan

Automatic:
The application contains unit tests to automatically check values. These tests can be run by running *ng test* from the command line interface.

Manual:
This test plan assumes the user has read README.md and has the application running.
The scope of this test plan is limited to checking the word representation of cheque values.

1. From the homescreen, the user should see a list of predefined cheques.
2. Select an existing Cheque, or click the Add Cheque button
3. The user should be presented with an numeric input field
4. Enter data from the following table and check the correctness of the word output

-

| Input | Output | Notes |
|---:|---|---|
| 1000000000000000 | ten trillion dollars | 100 trillion is over the 10 trillion max, so the field should revert to 10 trillion when a higher number is entered |
| -50000 | | 0 is the minimum value, so the field should revert to 0 when a lower number is entered |
| 0 | | $0 cheques are not supported, so no words are returned |
| 1 | one dollar | |
| 0.01 | one cent | |
| 47 | forty seven dollars | |
| 0.58 | fifty eight cents | |
| 1.36 | one dollar and thirty six cents | |
| 125 | one hundred and twenty five dollars | |
| 1000001 | one million and one dollars | |
| 89467215.36 | eighty nine million four hundred and sixty seven thousand two hundred and fifteen dollars and thirty six cents | |
| 1.306 | one dollar and thirty one cents | |
| z | | The user should not be able to enter non-numeric characters |

## Known Limitations

- The application does not handle $0 or <$0 cheque values.
- The application does not handle >$10 trillion cheque values. Higher values introduce precision issues in the first two decimal places, but there should be little to no call for cheques this large.

## Angular and Typescript functionality used

- Systems
    - NumberFormatter
        - Used to format numbers
        - Currently implemented:
            - IntAsArray - converts a number to an array of digits,
            - **formatNumArrayAsWords - converts an array of digits into words**
    - Cheque System
        - System used by Cheque List and Cheque Details to set and display cheque data
        - Uses mock API for CRUD functions.
        - **Uses NumberFormatter system to convert cheque values into a string**
- Components
    - ChequeList
        - Displays a list of cheques from the Cheque Service
    - ChequeInput
        - Allows cheque values to be modified
    - ChequeDetail
        - Nested in ChequeInput
        - **Displays cheque value as words**
- Interfaces
    - Cheque
        - Interface used for cheque objects
- Routing
    - Routing used to navigate between landing page/cheque list and cheque editor
    - Can navigate to individual cheque with /cheque/:ID
- Karma unit tests
    - Runs unit tests for multiple cheque values to check the correct string is generated