Numbers to Words API

with front-end application

October 2017

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version # | Changes | Author | Date |
| 1.0 | Recorded the Numbers to Words API purpose, process, technical details and usage | Marvin Lacuna | 19 Oct 2017 |

Reviewers

| Reviewed by | Title | Approval Evidence (optional) | Date |
| --- | --- | --- | --- |
| Marvin Lacuna | Tech Architect |  | 19 Oct 2017 |

Table of Contents

[1. Introduction: About This Document 4](#_Toc496225202)

[1.1 Purpose 4](#_Toc496225203)

[1.2 Target Audience 4](#_Toc496225204)

[1.3 Glossary 4](#_Toc496225205)

[2. Prerequisites 5](#_Toc496225206)

[3. Functional Requirements 5](#_Toc496225207)

[4. Process Diagram and Output 8](#_Toc496225208)

[1. US-001 - Convert Numbers to Words via web page 8](#_Toc496225209)

[2. US-002 - Convert Numbers to Words via Postman 9](#_Toc496225210)

[5. Technical Implementation 12](#_Toc496225211)

[5.1 Approach 12](#_Toc496225212)

[5.1.1 Summary 12](#_Toc496225213)

[5.1.2 Process API 17](#_Toc496225214)

[5.1.3 DataModel 17](#_Toc496225215)

[5.2 General Considerations 18](#_Toc496225216)

[5.3 Source Control 18](#_Toc496225217)

[5.4 Enhancements 18](#_Toc496225218)

[5.5 Troubleshooting 18](#_Toc496225219)

[5.6 Additional Resources 18](#_Toc496225220)

# **Introduction: About This Document**

## **Purpose**

This document details the purpose, process, technical details and usage of **NUMBERS TO WORDS API CONVERTER.**

As a background overview, the tech architect **MARVIN LACUNA** has been tasked to create a simple application that converts numbers to words using Microsoft C# language as the backend programming to showcase his skills within 48 hours’ time.

The code is publicly available and can be downloaded in GitHub - <https://github.com/marvinglennlacuna/NumbersToWordsConverter.Api>.

Note that the **NUMBERS TO WORDS API CONVERTER** is only a proof of concept and should not be used for commercial purposes.

## **Target Audience**

1. Technical Architects
2. Developers
3. Students

## **Glossary**

1. **C# Language** - C# is an elegant and type-safe object-oriented language that enables developers to build a variety of secure and robust applications that run on the .NET Framework. You can use C# to create Windows client applications, XML Web services, distributed components, client-server applications, database applications, and much, much more. Visual C# provides an advanced code editor, convenient user interface designers, integrated debugger, and many other tools to make it easier to develop applications based on the C# language and the .NET Framework. (<https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>)
2. **GitHub** - GitHub is a web-based Git or version control repository and Internet hosting service. (<https://en.wikipedia.org/wiki/GitHub>)
3. **Microsoft Visual Studio** - An integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services. (<https://www.visualstudio.com/>)
4. **Postman -** A Google Chrome app for interacting with HTTP APIs. It presents you with a friendly GUI for constructing requests and reading responses.

# **Prerequisites**

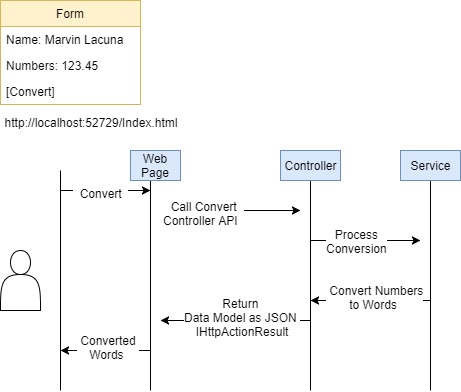
1. Understanding of Source Control and C# and MVC
2. Understanding of how Web API works
3. Understanding of HTML, JavaScript and JQuery
4. Understanding on how Postman works
5. Must have Visual Studio installed in your machine
6. Must have Postman installed in your machine
7. Must have Git and Git client installed in your machine

# **Functional Requirements**

|  |  |
| --- | --- |
| ID: US-001 | As an anonymous user, I want to input my name and numbers in a web form page and convert the numbers into words then display to a web page. |
| US name | Convert Numbers to Words via web page. |
| Actors | **Anonymous User** |
| Priority | Must |
| Initial conditions | None |
| Main flow | 1. User goes to <http://localhost:52729/Index.html> 2. Fill in the name field with “Marvin Lacuna” 3. Fill in the numbers field with 123.45 4. Click the **Convert** button |
| Alternative flow(s) | N/A |
| Acceptance criteria | 1. Webpage must be able to display numbers in words   Output  Marvin Lacuna  ONE HUNDRED TWENTY-THREE DOLLARS AND FORTY-FIVE CENTS |
| External references | N/A |
| Additional details | 1. weReview and test the functionality. 2. Ensure the form validations are tested 3. Name field must only accept alphabet and space 4. Numbers field must only accept numbers and decimal points 5. Numbers field must only process from 1 up to 9,999,999 million |
|  | |
| ID: US-002 | As a technical user, I want to POST my name and numbers as JSON via Chrome Postman, the response body must return the ‘name’, ‘numbers’ and ‘numbers to words’ as a JSON. Status code should be 200 OK. |
|  | Convert Numbers to Words via Postman |
| Actors | **Technical User** |
| Priority | Must |
| Initial conditions | Ensure Postman is installed. If not, download [here](https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop?hl=en). |
| Main flow | 1. Technical opens the Chrome Postman 2. Enter this in the ‘Request URL’ field : <http://localhost:52729/api/Converter/Process> 3. In the ‘**Headers**’ tab, put ‘Content-Type’ in the ‘Key’ field and the ‘application/json’ in the ‘Value’ field 4. In the ‘Body’ tab, select ‘raw’ radio button and put this JSON string in the text field   {  "Name":"Marvin Glenn Lacuna",  "Numbers":123.45  }   1. Click **Send** button |
| Alternative flow(s) | N/A |
| Acceptance criteria | 1. The response body must display status 200 OK with JSON string as the result   {  "Name": "Marvin Glenn Lacuna",  "Numbers": 123.45,  "Words": "ONE HUNDRED TWENTY-THREE DOLLARS AND FORTY-FIVE CENTS"  } |
| External references | N/A |
| Additional details | 1. Review and test the functionality. 2. Test invalid JSON object. It should display an error handling message. 3. Name field must only accept alphabet and space 4. Numbers field must only accept numbers and decimal points 5. Test with GET and it should display method not allowed. |

# **Process Diagram and Output**

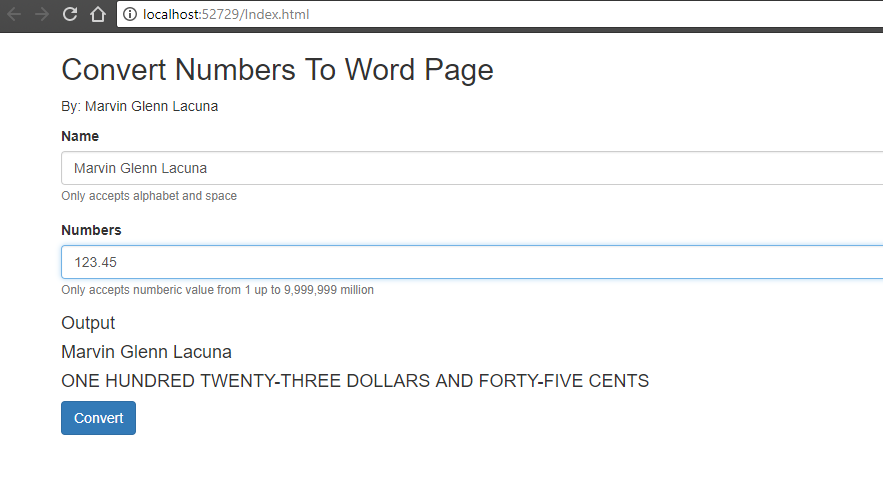
## **US-001 - Convert Numbers to Words via web page**



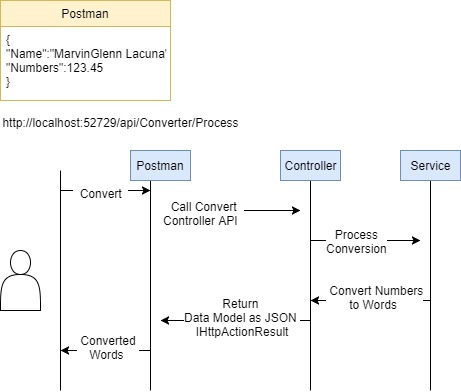
Note: To edit the diagram, use the attached file below and open in <https://www.draw.io/>



**Output**:



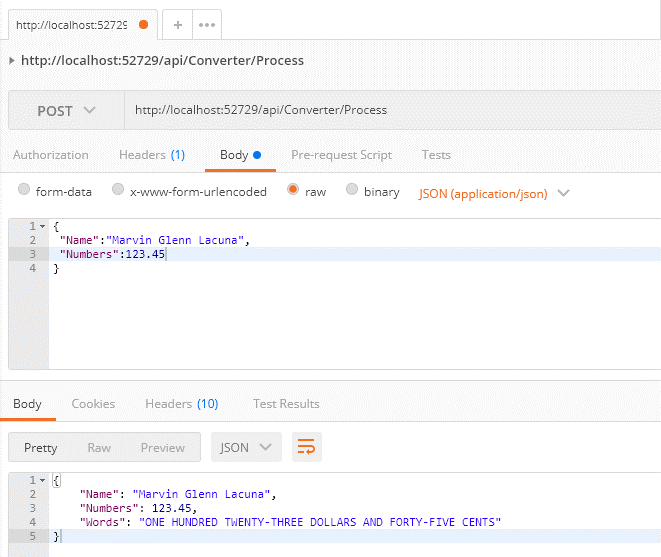
## **US-002 - Convert Numbers to Words via Postman**



Note: To edit the diagram, use the attached file below and open in <https://www.draw.io/>



**Output**:



# **Technical Implementation**

## **Approach**

### **Summary**

* Created a DataModel to store the user input data (name & numbers)
* Created a NumbersToWordsModel to store the place value
  1. Ones – ranges from 1-9
  2. Teens – ranges from 10 – 19
  3. Tens – increment by 10-90
  4. Hundreds, thousands, ten thousand, hundred thousand, million - ranges from 100-9999999
* Created an API Controller that will be exposed as an entry point/endpoint of conversion

1. Validate the model against regex filters and error handling. Use the ResponseMessageService to pass the error message and return as ResponseErrorMessageModel.
   * 1. Check if numbers model has string value or less than 1 and greater than 9,999,999, then throw bad request with error message

Status code: 400 Bad Request

“Numbers must be numeric value from 1 up to 9,999,999 million”

* + 1. Check if name model is null or empty, then throw bad request with error message

Status code: 400 Bad Request

“Name cannot be null or Model is invalid format.”

* + 1. Check if name model has integer, then throw bad request with error message

Status code: 400 Bad Request

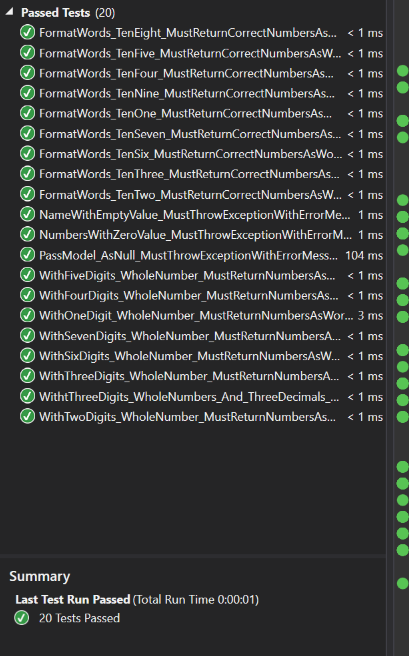
“Name must be alphabet and space only and cannot be null.”

1. Process the DataModel to Converter Service

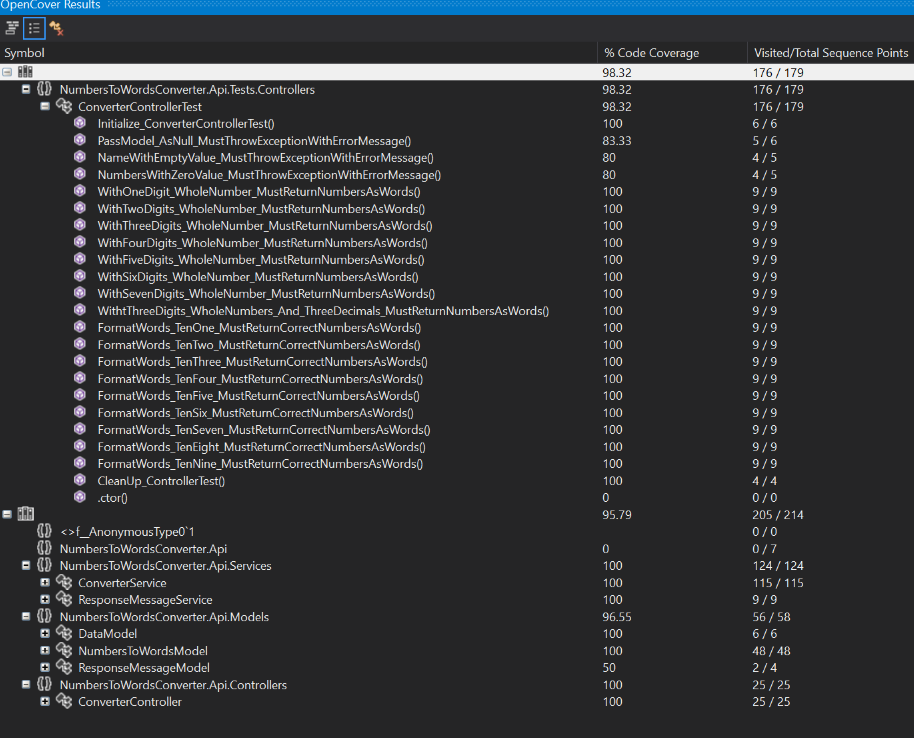
* Created Converter Service to do the logic of conversion

1. Add conditions to determine the digits. If matched, then convert the number range to words. Do recursive call until all numbers are converted to words. Use ‘/’ to get the whole number and % to get the remainder.
2. Format the words and clean TEN-ONE, TEN-TWO to TEN-NINE words to teens.
3. Format the words with ‘–‘

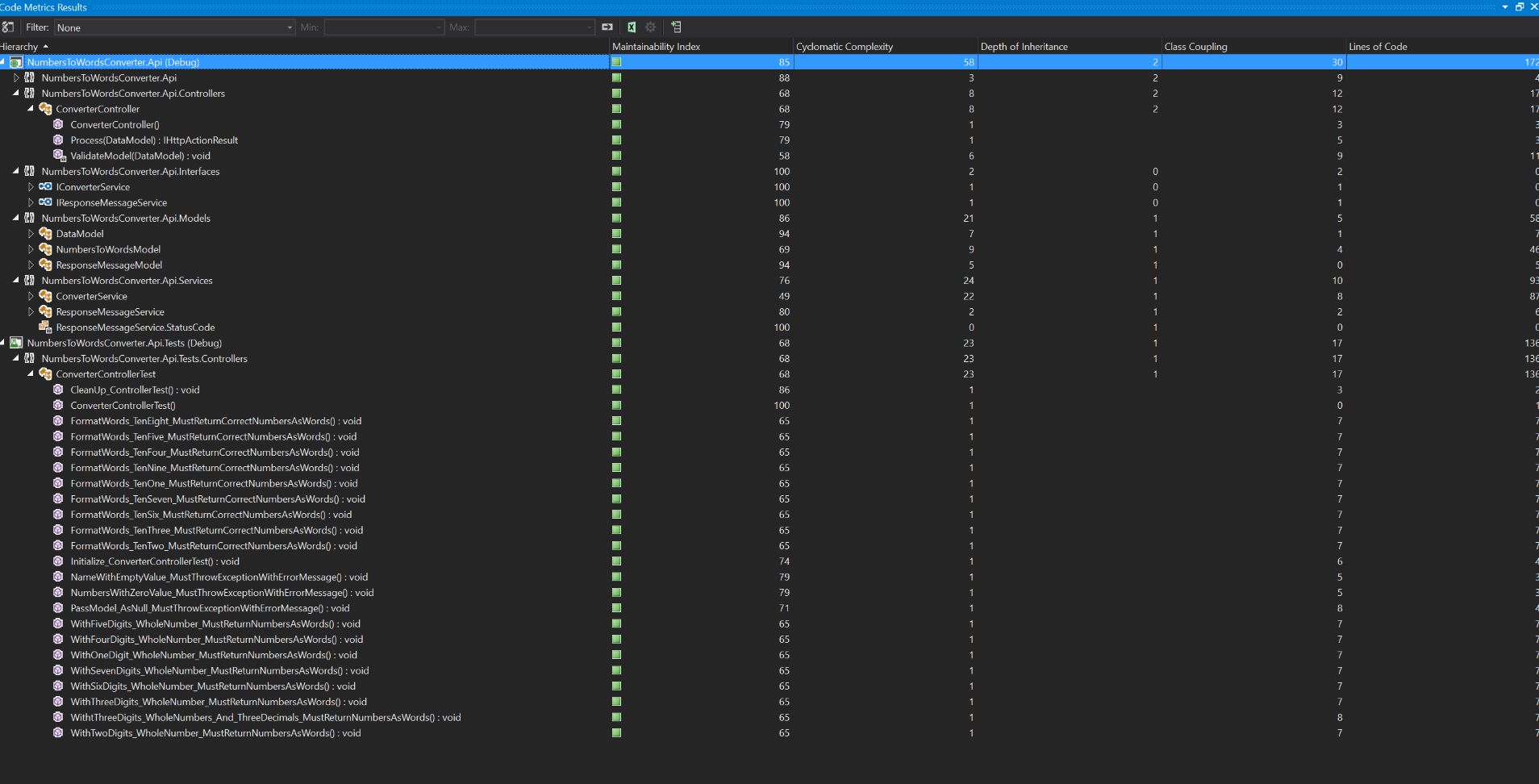
* Created a front-end application the displays the numbers to word in the web page
  1. Created a form with name and numbers field
  2. Created a button to help trigger the conversion
  3. Created an error handling in the html input type to prevent unexpected value
  4. Called the API via AJAX by passing the name and numbers as JSON object using Post.
  5. Displayed the converted numbers to words to the web page using Jquery
* Created unit testing to test all the scenarios using MSTest.



* Checked the code coverage to ensure that the testing has covered each condition using **OpenCover** VS Tools Extension.



* Checked the **Code Metrics** to determine the Cyclomatic Complexity



### **Process API**

|  |  |
| --- | --- |
| **API Path** | /api/Converter/Process |
| **Method** | Post |
| **Input** | DataModel |
| **Output** | IHttActionResult |
| **Remarks** | Convert Numbers to Words |

### **DataModel**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Mandatory** | **Remarks** |
| Name | String | Yes | Person’s name |
| Numbers | Integer | Yes | Numbers to convert to words |
| Words | String | No | Converted string |

**Sample Input JSON**

|  |
| --- |
| {  "Name":"Marvin Glenn Lacuna",  "Numbers":123.45  } |

**Sample Output JSON**

|  |
| --- |
| {  "Name": "Marvin Glenn Lacuna",  "Numbers": 123.45,  "Words": "ONE HUNDRED TWENTY-THREE DOLLARS AND FORTY-FIVE CENTS"  } |

## **General Considerations**

* Do not change the model or the implementation without running unit testing
* Ensure that the cove coverage is more than 90%
* Avoid excessive cyclometric complexity
* When forking the repository, do not check-in assemblies or remove the specific user files from the *.gitignore* exclusion list

## **Source Control**

Use Git client like [TortoiseGit](https://tortoisegit.org/) or [SourceTree](https://www.sourcetreeapp.com/) to download the code repository in the Github –

Download: <https://github.com/marvinglennlacuna/NumbersToWordsConverter.Api>

## **Enhancements**

* C# Log4Net: <https://stackify.com/log4net-guide-dotnet-logging/>
* Inversion of Control: <https://msdn.microsoft.com/en-us/library/ff921087.aspx?f=255&MSPPError=-2147217396>
* Controller Dispose method - Releases unmanaged resources and optionally releases managed resources. <https://msdn.microsoft.com/en-us/library/dd492699(v=vs.118).aspx>
* Improve Unit Testing

## **Troubleshooting**

* Ensure that you have the latest .Net Framework installed
* Ensure your user windows permission
* Ensure that the JSON string passed is in the correct format
* Ensure that you are calling the correct API endpoint: <http://localhost:52729/api/Converter/Process>
* Ensure that you have downloaded all the files from Github with no errors
* Ensure that your local anti-virus found the application safe (not harmful/no threats).

## **Additional Resources**

* <http://www.c-sharpcorner.com/article/create-simple-web-api-in-asp-net-mvc/>
* <https://docs.microsoft.com/en-us/aspnet/web-api/overview/testing-and-debugging/unit-testing-controllers-in-web-api>
* <https://github.com/OpenCover/opencover>