Mars Rover C# Web API Test

Project Prompt

For your test, we'd like for you to develop a RESTful Web API that NASA will use to move a Mars rover from the space station.

In order to control a rover, NASA sends a simple string of letters. The possible letters are "L", "R", and "M". "L" and "R" makes the rover spin 90 degrees left or right respectively, without moving from its current spot. "M" moves the rover forward one grid point, and maintains the same heading.

A rover's position and location is represented with X and Y coordinates and a letter representing one of the four cardinal directions (N, E, S, W). The plateau is divided up into a grid to simplify navigation. The default starting position of the Rover is 0, 0, N (X, Y, Direction).

Goals

The web services must:

- Save data between calls to the Web API after initial start-up (but a database is not required)
- Support JSON
- Properly validate input parameters
- Follow RESTful best practices for naming, routing, HTTP verbs (actions), and HTTP status codes

For the solution, you are expected to use:

- C# .NET, Web API, and Visual Studio
- Swagger for API documentation
- Also feel free to use any free 3rd party libraries via NuGet

We will the grade the test based on the quality of your work so please don't rush through it.

The goal of the test is to understand how you:

1. Develop in C# and Web API
2. Implement RESTful Web API services
3. Structure a project
4. Use OOP and design best practices
5. Use .NET libraries

Your deliverable is a working VS solution back to us. You are free to deliver your completed test via zip file, Google Drive, GitHub, etc.

Please follow the instructions below as closely as possible. The test will likely take you 2-3 hours, but feel free to take more or less time.

Requirements

- Create an endpoint to create a new instance of a Rover
    - Request Parameters:
        - RoverId, Integer, Required
        - RoverName, String, Required
- Create an endpoint to rename an existing instance of a Rover
    - Request Parameters:
        - RoverId, Integer, Required
        - RoverName, String, Required
- Create an endpoint to move an instance of a Rover
    - Request Parameters:
        - RoverId, Integer, Required
        - MovementInstruction, String, Required
            - Valid instructions letters include “L”, “R”, “M”. No other characters (or spaces) should be allowed
    - Response:
        - RoverId, Integer
        - RoverName, String
        - CurrentX, Integer
        - CurrentY, Integer
        - CurrentDirection, String
- Create an endpoint to retrieve the current position/direction of an instance of a Rover
    - Request Parameters:
        - RoverId, Integer, Required
    - Response:
        - RoverId, Integer
        - RoverName, String
        - CurrentX, Integer
        - CurrentY, Integer
        - CurrentDirection, String

See next page for example scenarios

Example scenarios

Rover starts at 0, 0, N (X = 0, Y = 0, Direction = N)

	5											
	4											
	3											
↑	2											
N	1											
	0						🟩 ↑					
S	-1											
↓	-2											
	-3											
	-4											
	-5											
		-5	-4	-3	-2	-1	0	1	2	3	4	5
					←	W		E	→			

1. Instruction: **“M”**. Move.
   b. Result: 0, 1, N.

	5											
	4											
	3											
↑	2											
N	1						🟥 ↑					
	0						🟩 ↑					
S	-1											
↓	-2											
	-3											
	-4											
	-5											
		-5	-4	-3	-2	-1	0	1	2	3	4	5
					←	W		E	→			

2. Instruction: **“RMM”**. Rotate right, Move, Move.
   a. Result: 2, 1, E.

	5											
	4											
	3											
↑	2											
N	1						🟩 ↑	🟩	🟥 →			
	0											
S	-1											
↓	-2											
	-3											
	-4											
	-5											
		-5	-4	-3	-2	-1	0	1	2	3	4	5
					←	W		E	→			

3. Instruction: **“LMMM”**. Rotate left. Move, Move, Move.
   c. Result: 2, 4, N.

	5											
	4								🟥 ↑			
	3								🟩			
↑	2								🟩			
N	1								🟩 →			
	0											
S	-1											
↓	-2											
	-3											
	-4											
	-5											
		-5	-4	-3	-2	-1	0	1	2	3	4	5
					←	W		E	→			