Back-end Interview Exercise

- Follow the directions below to build an Asp.Net Web API service
- Create a `README.md` that explains how to run the project
- Check all of the above into a new public repository in GitHub
- Email the repo link to your HR contact!

Problem

At PeopleNet we're all about the transportation industry. For this exercise we would like to help route a truck from point A to point B on a fictional "map".

Your web service should expose a single endpoint on port 8080. It should be a POST method that accepts our "map" as its body. The endpoint should be `/solveMaze`. It should return a JSON response that contains an integer property `steps` that represents the number of steps to get from A to B. Additionally, it should return a string `solution`, similar to the input, with an additional character, `@`, representing the solution path.
The POST body will be a string representing the maze, that contains the following types of characters:

- `.` represents an open road
- `#` represents a blocked road
- `A` represents the starting point
- `B` represents the destination point

Anything outside the bounds of the array should be considered a wall. In addition, you may only move in horizontal or vertical directions. Diagonal movements are not allowed.

Example input maze:

```
##########
#A...#...#
#.#.##.#.#
#.#.##.#.#
#.#....#B#
#.#.##.#.#
#....#...#
##########
```

Example visualization of the solution, using `@` to represent the correct path.

```
##########
#A@@.#...#
#.#@##.#.#
#.#@##.#.#
#.#@@@@#B#
#.#.##@#@#
#....#@@@#
##########
```

For the above sample, the web service should return:

```
{
  "steps": 14,
  "solution": "##########
               #A@@.#...#
               #.#@##.#.#
               #.#@##.#.#
               #.#@@@@#B#
               #.#.##@#@#
               #....#@@@#
               ##########"
}
```

Goal

Your solution should be able to solve the included mazes `maze1.txt` (pictured above), `maze2.txt`, and `maze3.txt` correctly. Each maze solution should run in under a minute on reasonable hardware.