Rakhesh Rajan

mailsrakhesh@gmail.com

Abstract

This document provides an overview on the solution, the business logic and guides on how to consume the API’s.

BATTLESHIP GAME

API Specifications & Solution Overview

# Business Logic Overview

## Board Unit Coordinates.

The coordinates in the board is identified with the below coordinate’s numbers. The board can be changed to any size and the coordinate identification will follow the below pattern. While calling API’s the coordinates location should be passed considering the below pattern.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Y** | 1,1 | 2,1 | 3,1 | 4,1 | 5,1 | 6,1 | 7,1 | 8,1 | 9,1 | 10,1 |
| 1,2 | 2,2 | 3,2 | 4,2 | 5,2 | 6,2 | 7,2 | 8,2 | 9,2 | 10,2 |
| 1,3 | 2,3 | 3,3 | 4,3 | 5,3 | 6,3 | 7,3 | 8,3 | 9,3 | 10,3 |
| 1,4 | 2,4 | 3,4 | 4,4 | 5,4 | 6,4 | 7,4 | 8,4 | 9,4 | 10,4 |
| 1,5 | 2,5 | 3,5 | 4,5 | 5,5 | 6,5 | 7,5 | 8,5 | 9,5 | 10,5 |
| 1,6 | 2,6 | 3,6 | 4,6 | 5,6 | 6,6 | 7,6 | 8,6 | 9,6 | 10,6 |
| 1,7 | 2,7 | 3,7 | 4,7 | 5,7 | 6,7 | 7,7 | 8,7 | 9,7 | 10,7 |
| 1,8 | 2,8 | 3,8 | 4,8 | 5,8 | 6,8 | 7,8 | 8,8 | 9,8 | 10,8 |
| 1,9 | 2,9 | 3,9 | 4,9 | 5,9 | 6,9 | 7,9 | 8,9 | 9,9 | 10,9 |
| 1,10 | 2,10 | 3,10 | 4,10 | 5,10 | 6,10 | 7,10 | 8,10 | 9,10 | 10,10 |
| **X** | | | | | | | | | |

## How the ships coordinates are assigned?

Logic checks for the first unallocated unit, then checks for the direction. Based on the size required, it gets the next unit numbers. Code next checks if the coordinates are valid (is it inside 10 X 10?), or allocated (any in the bunch)?

For e.g. if the user requests to allocate 100 units with unit size. It will occupy all board and next request it will return message saying there is no space.

* <https://battleshipapi1.azurewebsites.net/api/BattleShip/CreateShip/2/h>

will allocate 1,1; 2,1 when called the first time.

* <https://battleshipapi1.azurewebsites.net/api/BattleShip/CreateShip/4/h>

will allocate 3,1 ;4,1; 5,1; 6,1.

* <https://battleshipapi1.azurewebsites.net/api/BattleShip/CreateShip/6/h>

will find the 7,1coordinate unused, but the next coordinates would be 7,1; 8,1; 9,1; 10,1; 11,1; 12,1. Since 11 and 12 are invalid it will look for next valid coordinates and the correct once will be – 1,2; 2,2; 3,2; 4,2; 5,2; 6,2;

Ships can be allocated in any direction in any order.

## Multi User Support

The game is multi user supporting. If user name is passed with the API’s it will allocate/fire for the specific user. If user name not specified it will hit the default “System” user. Calling with out the user creates the scenario, playing against “System”.

# API Specifications

# CreateShip

Following are the variations of the method

1. [https://battleshipapi1.azurewebsites.net/api/BattleShip/CreateShip/{size}/{direction}](https://battleshipapi1.azurewebsites.net/api/BattleShip/CreateShip/%7bsize%7d/%7bdirection%7d)
2. [https://battleshipapi1.azurewebsites.net/api/BattleShip/CreateShip/{size}/{direction}/{playerName}](https://battleshipapi1.azurewebsites.net/api/BattleShip/CreateShip/%7bsize%7d/%7bdirection%7d/%7bplayerName%7d)

Size – Integer (How many number of units are required for the ship?)

Direction – “h” or” v”, Default value if “h” (h for horizontal, v for vertical)

Calling this method, the first time will initialize the board for the system user. Users can send the size of the ship and the direction of the ship. This method will return the coordinates and the response code.

Syntax to be used. For e.g.

* <https://battleshipapi1.azurewebsites.net/api/BattleShip/CreateShip/2/h> -

Will create a board when the first time called and will place the ship in 1,1 and 2,1 coordinates.

* <https://battleshipapi1.azurewebsites.net/api/BattleShip/CreateShip/3/h> - will place the ship in 3 units available in the horizontal direction.

To place a ship in vertical direction, use the below style

* <https://battleshipapi1.azurewebsites.net/api/BattleShip/CreateShip/3/v>

Example of calling API with username

* <https://battleshipapi1.azurewebsites.net/api/BattleShip/CreateShip/1/h/Rakhesh>

# Fire

Following are the variations of the method

1. [https://battleshipapi1.azurewebsites.net/api/BattleShip/Fire/{x}/{y}](https://battleshipapi1.azurewebsites.net/api/BattleShip/Fire/%7Bx%7D/%7By%7D)
2. [https://battleshipapi1.azurewebsites.net/api/BattleShip/Fire/{x}/{y}](https://battleshipapi1.azurewebsites.net/api/BattleShip/Fire/%7Bx%7D/%7By%7D)/{playerName}

This method accepts the coordinates and returns the status. If ships all units are hit then or if all ships are down then that information also will be additionally provided to the caller.

Example call below

Without User -

* <https://battleshipapi1.azurewebsites.net/api/BattleShip/Fire/1/1>

With User -

* https://battleshipapi1.azurewebsites.net/api/BattleShip/Fire/1/1/Rakhesh

# RestartGame

Once all ships are sunk, any calls to Fire will be returned back with a message "Game is already completed. Please restart the game to continue playing."

Calling this "RestartGame" method will reset all settings including users and the play is restarted.

Syntax to be used - <https://battleshipapi1.azurewebsites.net/api/BattleShip/RestartGame>

# Design Capabilities

* The architecture allows single / multi user gaming.
* The board size is configured as 10 x 10. It is changeable to any size.
* The ship direction is currently Vertical and Horizontal. The logic is extendable to support any new allocation pattern.

Game does a few validations for example -

* If a user tries to fire after the game is completed, then he will receive the message. "message": "Game is already completed. Please restart the game to continue playing."
* If a user tries to before allocating ships, he will get the message "message": "No ships placed in the board."
* If all units are already allocated to the board when trying to allocate a new ship or passing a wrong size (for e.g. 12) while allocating ship will prompt the message

“message": "No units available in the board"

* If a coordinate is already hit, the user will be thrown the error - "message": "This unit is already hit."
* Business logic checks for invalid coordinates and already allocated coordinates by default.

Note:

The application is developed using the below IDE version 