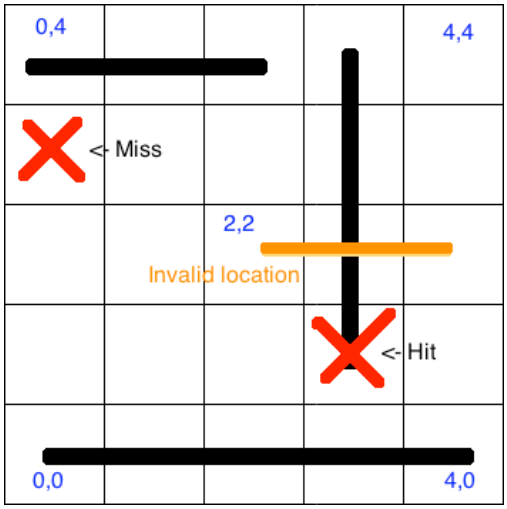
BattleShip Game

This is a take home interview question for potential candidates.

In the game of battleship, an enemy is attacking your city from the sea. They are attacking you with air carriers, battleships and submarines which are of different sizes. Two ships can not overlap on the map. Your job is to shoot down all of the enemy ships. The enemy has radar deflection technology on their ships, so you do not know where they are. Luckily, the ships can not move without giving away their position, so throughout the game, they will not move. You can fire a rocket anywhere on the map. If you hit the ship, you will see a big explosion so you know you've destroyed the ship. Once the ship is hit, it will sink to the ocean floor, so attempting to shoot the same ship will result in a miss.

The objective is to implement this game as a web-service. It will have 2 endpoints - to place a ship (**/ship**) and to shoot at a position on the map (**/shoot**). The map will be a 20 x 20 grid. Ship sizes will be 5 for an air carrier, 4 for a battleship and 3 for a submarine. Each ship type can appear zero or more times on the grid. An implementation of the game, without the /shoot endpoint, has been provided in python. Edit the provided program (battleship.py) so that the game works as described in this spec.

This small map will help illustrate the game.



In this example, you've sent a ship to (0,0) of length 5 horizontally. A ship of length 4 to (3,4) vertically. The ship sent to (2,2) horizontally is invalid, since there is another ship at (3,2). Shooting at (0, 3) resulted in a miss and shooting at (3,1) resulted in a hit.

Send an enemy battleship to a location on the map

POST $BASE_URL/ship

Parameters

You will need to pass a json map containing the location, direction and length of the ship in the body. See the example below

Parameters

```
{
  "location": {"x": 5, "y": 3},    // the location of the ship. This is the top-most/
left-most point on the map for a vertical/ horizontal ship respectively.
  "direction": "horizontal",       // either horizontal or vertical
  "length": 3                      // The number of spaces the ship will take on the
map
}
```

Responses

200: The ship was successfully placed on the map

400: An error occurred in sending the ship to war

```
{
  "code": "ERR_OTHER",                          // The required error code. See list
of error codes below
  "message": "An un-handled error occurred"     // An optional description of the
error
}
```

Shoot an enemy battleship

POST $BASE_URL/shoot

Parameters:

Pass in the location on the map where you would attempt to shoot.

```
{
  "location": {"x": 5, "y": 3"} // The location on the map where you will attempt to
shoot
}
```

Responses:

200:

If the missile was fired successfully. There are 3 possible outcomes.

```
// The missile has successfully taken down a ship.
{ "result": "Hit" }

// The missile did not hit a ship.
{ "result": "Miss" }

// All the ships on the map have been sunk.
{ "result": "Victory" }
```

400:

```
{
  "code": "ERR_OTHER" // The required error code. See list of error codes below
  "message": "A generic error occurred" // An optional description of the error.
}
```

Error codes:

ERR_OTHER - An un-handled error occurred

ERR_SHIP_COLLIDE - A ship already exists on the map at this location

ERR_INVALID_LOC - An invalid position on the map