# Connect Four - Server

## Goals

Create a simple server which accepts a single API call to evaluate the board state of a game of Connect Four.

**The code should be written as if you would be happy to put into a pull request for someone else to review.** (Obviously there is some amount of time boundary - favor a working system vs one that doesn't)

If you aren't familiar with Connect Four, here is a quick explanation and gif:

http://mathworld.wolfram.com/Connect-Four.html

as well as a video

the rules: [https://www.youtube.com/watch?v=H3FYRM9a0i4](

## Specification and Description

Please create an api call of the following form. We may run tests against your server.

Please use port `8080` as well.

**POST** `/evaluate_board_state`

Accepts a game state as `JSON` (specified below).

- `0` denotes an empty slot
- `1` denotes a Black piece (or player 1)
- `2` denotes a Red piece (or player 2)

Example:

```
{
    "board": [
        [0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 2, 0, 0],
        [0, 2, 2, 2, 1, 1, 0],
        [0, 1, 1, 1, 1, 2, 0],
    ]
}
```

Should produce a winning response for Black

Upon receiving the above, the route handler should:

- Validate the request body and provide errors accordingly
- Evaluate the board state around a variety of conditions. Included are a set of conditions to account for, you are also free to account for more as desired.
    - If the board state is valid, return who is the winner, stalemate, or whos turn it is (Black vs Red).
        - If there is a winner, return information around the location of the winning pieces if possible.
    - If the board state is invalid, return an error response, along with reasoning around the error, here are several to account for, accounting of at least one would be great! Feel free to come up with others as time permits:
        - Does one color have more pieces than they should on the board?
        - Is there a piece (or pieces?) disconnected from the rest? (stretch goal)
        - Is there more than one set of (at least) 4 connected pieces? (stretch goal)

More information around game state / game progression and algorithms.

- Black (player 1) *always* go first
- A simple algorithm for checking if there are winners is fine. Here is the simplest as a pure function of board state:
    - Check for (at least) 4 horizontal pieces of the same color in a row
    - Check for (at least) 4 vertical pieces of the same color in a row
    - Check for (at least) 4 top left to bottom right diagonal pieces of the same color in a row
    - Check for (at least) 4 bottom left to top right diagonal pieces of the same color in a row

We will be looking for *completeness*, but also *quality*. We also empathize with this is being developed on a pseudo-timer.

# Delivery

When complete, please add your project to your github and add `jaredpetker` as a collaborator. You can do this as a private repo if you like as well.