You will make an API that plays the game called tic-tac-toe in the US, and called naughts and crosses in some countries.

Steps to reproduce the first Challenge:

1. Your server will be provided the current board in a GET request, using the ‘board’ parameter in the query string.

About this point the Api will look like this.

```
method: 'GET',
  url: 'https://tic-tac-toe-api.com/?board=+xxo++o++'
```

2. If the board string doesn't represent a valid tic-tac-toe board, or it’s not plausibly o’s turn, your server should return an HTTP response code 400 (Bad Request).

on this point the valid tic-tac-toe string should be nine string of (o,x or space) and check if o is not added in the occupied place or if client did not do the same mistake if so server should return (throw new HttpException(‘The request is not valid’))

3. Your server always plays as o

about this I have to make sure that all o’s turns should come from the server after making a request(x or client make a request).

4. Either player can go first:

this means that we have two different cases either send first request with space only which means the server is the first to play and return the response with O added in the board.

Second case is when me(X or client) goes first and send request with one X when it’s valid should return with an O added in the board.

5. If the board is a valid tic-tac-toe board and it is plausibly o's turn, your server should return a string representation of the same board with one ‘o’ added.

6. If possible and time permits, your tic-tac-toe api should play optimally (i.e. never lose when it is possible to force a tie, or tie when it is possible to win).

a. The best strategy is probably to search the game tree for winning moves:

those are some steps:

IF YOU GO FIRST…

Avoid placing your first piece on an edge square, and keep it on the center or a corner square. Placing it on an edge square will leave you vulnerable and give your opponent the advantage.

1) Center

If you mark the center, your opponent will either place his/her first piece on an edge or corner piece.

if they mark an edge, it's incredibly easy to win - There's no chance of even tying. Simply place your next piece on one of the two corners furthest from the edge piece. They will most likely block that move, which in turn gives them an opportunity to win. Block their move, and suddenly, you have two ways to win, and your opponent is helpless.

In our API I should consider this case I I send the request with x in center
if board = ++++x++++
to let the server win or tie:
the return string : board = o+++x++++ or ++o+x++++ or ++++x+o++ or +++ +x+++o

If they mark a **corner,**as a smarter opponent would, it's a little bit more complicated. Place your next mark on the opposite corner, or the corner that would make a diagonal of two X's and one O. If they place their next piece on an edge, they've made a mistake, and you now have *two* ways of winning, depending on which edge they placed their O on. Otherwise, assuming you keep counter-attacking, the game will end in a tie.

2) Corner
If they mark a corner, mark the center, or you will almost certainly lose against a good opponent. Then remember that there is one outcome in which a tie is possible from above.