[Technical Assessment](https://technical-assessment.konicaminoltamarketplace.com/)

**CONNECT THE DOTS FOR TWO PLAYERS**

**Overview**

Based on the requirement, I have chosen the WebSocket API Protocol for implementing the server part.

**Setup**

1. Client WebSocket API Config
2. Server
   1. Use NodeJS Express for run WebSocket Server and Client at same time
   2. Requirements:
      1. NodeJS - <https://nodejs.org/en/>
      2. Download and install node.js
      3. Through cmd use npm install and npm start to start the server.
      4. As configured, client will run at localhost:3000
      5. Server WebSocket will run at ws://localhost:8081
3. Websocket:

To connect and to listen the event. We have two events

* 1. ws.on('message')  
     For getting message from client
  2. ws.send

For sending the data to client.

The flow data for getting and sending data.

1. Whenever Client is ready, it will send INITIALIZE message to server

{

    "id": 1,

    "msg": "INITIALIZE",

    "body": **null**

}

1. After get INITIALIZE, server will respond to game start with first turn.

{

    "id": 1,

    "msg": "INITIALIZE",

    "body": {

        "newLine": **null**,

        "heading": "Player 1",

        "message": "Awaiting Player 1's Move"

    }

}

1. From this time, the flow of game will start with

a. Client send which node is clicked

{

    "id": 3,

    "msg": "NODE\_CLICKED",

    "body": {

        "x": 0,

        "y": 2

    }

}

b. Server will with flow like

* + 1. The first pick - move1
    2. The second pick - move2
    3. Each “move” will have 2 state “valid” and “invalid”

VALID\_START\_NODE, INVALID\_START\_NODE >> move1

VALID\_END\_NODE, INVALID\_END\_NODE >> move2

if each “move” is “invalid”, it will go to previous “move” in this case,

move 1 >> invalid, user must pick the first node again

move 2 >> invalid, user must pick the first node again

If “move 2” >> valid, Player will change from 1 >> 2 or 2 >> 1, then continue.

c. Game core: All logic of game is processed with two function checkMove1Valid and checkMove2Valid.

i. checkMove1Valid

It’s used for checking the node that user selects at first pick. It must be end of this path.

ii.checkMove2Valid

We process things here “Check the pair of node for draw line”

* We have 3 type line (Column, Row, Cross)
* Column is checked by two nodes have same X
* Row is checked by two nodes have same Y
* Cross is checked by 2x2 matrix and the path.

After any line is drawn, we will store the position of node to server for checking future node.  
Ex :

Node 1 ( x : 0 , y : 0)

Node 2 ( x : 0 , y : 1)

After that, if next pick is Node 1 or Node 2, they will be invalid, because of Node 1 and Node 2 is drawn the line already.