```
/**
 * @file ctc_block.js
 * @author Joey Damico
 * @date September 25, 2019
 * @summary Class that is a "block" or track, that makes up the
railroad
 *
 * @description This class is a section of track, between two
interlockings, this classes make up the railroad
 */

// Color Constants For Drawing Routes
const Empty = '#999999';
const Route = '#75fa4c';
const Occupied = '#eb3323';

/**
 * Class that is a "block" or track, that makes up part of the
railroad. This class is a section of track,
 * between two interlockings, this classes make up the railroad. The
block class variables that are basically
 * characteristics of a real piece of track
 *
 * @member block_name -> The name of the piece of track, usually the
two location it bridges
 * @member block_size -> The size of the track, (i.e. how long it
takes for a train to travel it)
 * @member block_status -> Wheter the block is Empty, Routed (A train
is coming), or Occupied (A train is there)
 * @member train_symbol -> The symbol or the train that occupys that
block
 */
class CTC_Block {
    /**
     * constructor(),
     * @summary The Constructor of the CTC_Block Class
     *
     * @description Sets all the memeber variables to their initial
values, when the application starts
     *
     * @param p_name, The Name of the Block
     * @param p_size, The Size of the Block
     * @param p_status, Current Status. Only Used for debugging when
build the applications
     */
    constructor(p_name, p_size, p_status) {
        this.block_name = p_name;
        this.block_size = p_size;
        this.block_status = p_status;
        this.train_symbol = null;
```

```
    }
    // ---- END constructor() ----

    /**
     * get_block_status()
     * @summary Getter for the block_status member variable
     *
     * @returns The current status of the block
     */
    get_block_status() {
        return this.block_status;
    }
    // ---- END get_block_status() ----

    /**
     * get_size()
     * @summary Getter for the block_size member variable
     *
     * @return The size of the block
     */
    get_size() {
        return this.block_size;
    }
    // ---- END get_size() ----

    /**
     * get_symbol()
     * @summary Getter for the train_symbol memebr variable
     *
     * @returns The symbol of the trail that is currently in the block
     */
    get_symbol() {
        return this.train_symbol;
    }
    // ---- END get_symbol() ----

    /**
     * reset_block()
     * @summary Resets the Block status to Empty
     *
     * @description This is used to reset the block, when the CTC
controller refreshes the train and route locations
     */
    reset_block() {
        // Check if the Block Is Routed
        if (this.block_status === Route) {
            this.block_status = Empty;
        }
    }
    // ---- END reset_block() ----
```

```
    /**
     * set_symbol()
     * @summary Setter for the train_symbol member variable
     *
     * @param n_symbol, The new symbols to set the member variable too
     */
    set_symbol(n_symbol) {
        this.train_symbol = n_symbol;
    }
    // ---- END set_symbol() ----

    /**
     * set_block_status()
     * @summary Sets the block current status based off of what tag is
passed in
     *
     * @param p_status, A String which is the Kinda of status of what
to set the block too
     */
    set_block_status(p_status) {
        if (p_status === 'Empty') {
            this.block_status = Empty;
        }
        else if (p_status === 'Route') {
            this.block_status = Route;
        }
        else if (p_status === 'Occupied') {
            this.block_status = Occupied;
        }
        else {
            console.log("ERROR!! - CTC_Block " + this.block_name +
" [set_block_status()]");
        }
    }
}

// This is required when using ReactJS
export default CTC_Block;
```