```swift
//
//  ViewController.swift
//  MacBeacon
//


import Cocoa
import CoreBluetooth
import Alamofire


class ViewController: NSViewController {

    // MARK: IBOutlets

    @IBOutlet weak var roomNumberLabel: NSTextField!
    @IBOutlet weak var descriptionLabel: NSTextField!
    @IBOutlet weak var zoneLabel: NSTextField!
    @IBOutlet weak var tableView: NSTableView!
    @IBOutlet weak var beaconSearchField: NSSearchField!

    // MARK: Private properties

    private var beacons:[Beacon] = []
    private var filteredBeacons:[Beacon] = []
    private var selectedBeacon: Beacon?
    private var zones:[String] = []

    // MARK: NSViewController methods

    override func viewDidLoad() {
        super.viewDidLoad()

        tableView.dataSource = self
        tableView.delegate = self
        beaconSearchField.delegate = self

        getBeacons()
    }

    override func prepare(for segue: NSStoryboardSegue, sender: Any?) {
        // Prepare to segue to emitting view
        if segue.identifier!.rawValue == "toTransmitting" {
            let destination_VC = segue.destinationController as!
             EmittingViewController
            // Pass the selected beacon object to the new view controller
            destination_VC.beacon = selectedBeacon
        }
    }

    // MARK: Internal methods

    internal func getBeacons(){
        let parameters: Parameters = [
```

```
        // Use the special macos request type
        "type": "macos.admin.get_beacons",
        "args": [
            "query": ""
        ]
    ]

    Alamofire.request(HTTPHelper.url, method: .post, parameters: parameters,
     encoding: JSONEncoding.default).responseJSON {
        response in

        switch response.result {
        case .failure( _):

            return

        case .success(let data):
            // First make sure a dictionary is recieved: Data validation
            guard let json = data as? [String : AnyObject] else {
                // Print statement for debugging purposes, not seen by users.
                print("Failed to get expected dictionary from webserver.")
                return
            }

            // Then make sure that key/value pairs are correct: Data validation
            guard let success = json["successful"] as? Int, let beacons =
             json["beacons"] as? [[String: String]] else {
                // Print statement for debugging purposes, not seen by users.
                print("Failed to get expected data from webserver")
                return
            }

            if success == 1 {
                var beaconObjectList = [Beacon]()
                for beacon in beacons {
                    beaconObjectList.append(Beacon(beaconJSONObject: beacon))
                }
                self.beacons = beaconObjectList
                self.filteredBeacons = self.beacons

                // Make sure that a selected beacon exists to prevent errors
                self.selectedBeacon = self.beacons[0]
                self.updateFieldsWithSelection()
                self.tableView.reloadData()
            }
        }
    }
}

internal func updateFieldsWithSelection() {
    roomNumberLabel.stringValue = (selectedBeacon?.roomNumber)!
    descriptionLabel.stringValue = (selectedBeacon?.description)!
    zoneLabel.stringValue = (selectedBeacon?.zoneName)!
}
```

```
}

// MARK: Extentions

extension ViewController: NSSearchFieldDelegate {
    func searchFieldDidStartSearching(_ sender: NSSearchField) {
        // Filter beacons by search bar string
        filteredBeacons = beacons.filter({ (beacon: Beacon) -> Bool in
            if beacon.roomNumber.contains(beaconSearchField.stringValue) {
                return true
            } else {
                return false
            }
        })
        self.tableView.reloadData()
    }

    func searchFieldDidEndSearching(_ sender: NSSearchField) {
        filteredBeacons = beacons
        self.tableView.reloadData()
    }
}

extension ViewController: NSTableViewDataSource, NSTableViewDelegate {

    func numberOfRows(in tableView: NSTableView) -> Int {
        return filteredBeacons.count
    }

    func tableView(_ tableView: NSTableView, viewFor tableColumn: NSTableColumn?,
     row: Int) -> NSView? {
        // Label the rows of the table with the room numbers
        let task = tableView.makeView(withIdentifier: tableColumn!.identifier,
         owner: self) as! NSTableCellView
        task.textField?.stringValue = filteredBeacons[row].roomNumber as String
        return task
    }

    func tableViewSelectionIsChanging(_ notification: Notification) {
        // Ensure that row selection is not out of range of possible beacons (make
         sure that the row exists)
        if tableView.selectedRow != -1 {
            selectedBeacon = filteredBeacons[tableView.selectedRow]
            updateFieldsWithSelection()
        }
    }

}
```